quantum reports

Tracked for Impact Factor

CITESCORE 3.3

Article

An Extended Analysis of the Correlation Extraction Algorithm in the Context of Linear Cryptanalysis

Christoph Graebnitz, Valentin Pickel, Holger Eble, Frank Morgner, Hannes Hattenbach and Marian Margraf

Special Issue

Quantum Computing: A Taxonomy, Systematic Review, and Future Directions

Edited by

Dr. Yousef Fazea

MDPI

https://doi.org/10.3390/quantum6040043

quantum reports

MDPI

Article

An Extended Analysis of the Correlation Extraction Algorithm in the Context of Linear Cryptanalysis

Christoph Graebnitz [1,*], Valentin Pickel [1], Holger Eble [2], Frank Morgner [2], Hannes Hattenbach [1] and Marian Margraf [1]

[1] Secure Systems Engineering, Fraunhofer AISEC, Lichtenbergstraße 11, 85748 Garching, Germany; valentin.pickel@aisec.fraunhofer.de (V.P.); hanneshattenbach.dev@gmail.com (H.H.); marian.margraf@aisec.fraunhofer.de (M.M.)

[2] Bundesdruckerei GmbH, Kommandantenstraße 18, 10969 Berlin, Germany; holger.eble@bdr.de (H.E.); frank.morgner@bdr.de (F.M.)

* Correspondence: christoph.graebnitz@aisec.fraunhofer.de

Abstract: In cryptography, techniques and tools developed in the subfield of linear cryptanalysis have previously successfully been used to allow attackers to break many sophisticated cryptographic ciphers. Since these linear cryptanalytic techniques require exploitable linear approximations to relate the input and output of vectorial Boolean functions, e.g., the plaintext, ciphertext, and key of the cryptographic function, finding these approximations is essential. For this purpose, the Correlation Extraction Algorithm (CEA), which leverages the emerging field of quantum computing, appears promising. However, there has been no comprehensive analysis of the CEA regarding finding an exploitable linear approximation for linear cryptanalysis. In this paper, we conduct a thorough theoretical analysis of the CEA. We aim to investigate its potential in finding a linear approximation with prescribed statistical characteristics. To support our theoretical work, we also present the results of a small empirical study based on a computer simulation. The analysis in this paper shows that an approach that uses the CEA to find exploitable linear approximations has an asymptotic advantage, reducing a linear factor to a logarithmic one in terms of time complexity, and an exponential advantage in terms of space complexity compared to a classical approach that uses the fast Walsh transform. Furthermore, we show that in specific scenarios, CEA can exponentially reduce the search space for exploitable linear approximations in terms of the number of input bits of the cipher. Neglecting the unresolved issue of efficiently checking the property of linear approximations measured by the CEA, our results indicate that the CEA can support the linear cryptanalysis of vectorial Boolean functions with relatively few (e.g., $n \leq 32$) output bits.

Keywords: quantum computing in cryptanalysis; linear cryptanalysis; linear approximation; correlation in linear cryptanalysis; vectorial boolean functions

MSC: 68Q12; 68W40; 94A60

Citation: Graebnitz, C.; Pickel, V.; Eble, H.; Morgner, F.; Hattenbach, H.; Margraf, M. An Extended Analysis of the Correlation Extraction Algorithm in the Context of Linear Cryptanalysis. *Quantum Rep.* **2024**, *6*, 714–734. https://doi.org/10.3390/quantum6040043

Academic Editor: Yousef Fazea

Received: 7 November 2024
Revised: 20 December 2024
Accepted: 21 December 2024
Published: 22 December 2024

1. Introduction

Quantum computing has the potential to enhance computations in different domains, particularly in cryptanalysis [1]. One of the most significant approaches in cryptanalysis is Shor's algorithm, which can be used to break asymmetric cryptography based on the prime factorization problem and the discrete logarithm problem [2–5]. Another approach is to use Grover's algorithm to determine the key of cryptographic primitives such as AES [6]. The progress in cryptanalysis using quantum computing impacts the research and usage of cryptographic primitives. For example, research into post-quantum cryptography has been massively accelerated mainly due to Shor's algorithm [7].

As quantum computing in cryptanalysis advances, a growing area of research is exploring whether it can enhance traditional cryptanalytic methods, such as algebraic [8–11],

differential [11–15], and linear [11,16–18] cryptanalysis. With that in mind, this paper focuses on analyzing the potential of a quantum algorithm to improve a crucial aspect of linear cryptanalysis. Before discussing the potential of that quantum algorithm, we first briefly introduce the essentials of linear cryptanalysis.

In 1993, Matsui and Yamagishi introduced the research field of linear cryptanalysis with a known plaintext attack on FEAL [19] to the public. Subsequently, Matsui showed that linear cryptanalysis could also be used to break the Data Encryption Standard (DES), requiring 2^{47} plaintext–ciphertext pairs to recover the encryption key [20]. Moreover, there are several extensions to linear cryptanalysis [21–23], but for the sake of simplicity, this paper focuses on Matsui's prototypical concept [20], which is introduced in the following paragraph.

Due to its statistical approach, applying linear cryptanalysis to a cipher or a vectorial Boolean function $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$ with m input and n output bits involves using a uniformly distributed random variable X with sample space $\mathbb{F}_2^m$. In detail, linear cryptanalysis requires at least one *mask* $(\alpha, \beta) \in \mathbb{F}_2^m \times \mathbb{F}_2^n$ such that the *linear approximation* $\langle \alpha, X \rangle = \langle \beta, f(X) \rangle$, expressed in terms of canonical inner products in $\mathbb{F}_2^m$, holds with a probability not equal to one half. In addition, the mask (α, β) with only zeros $\alpha = \beta = \mathbf{0}$ is called the trivial mask. Furthermore, the deviation from one half, specified as $\varepsilon_f : \mathbb{F}_2^m \times \mathbb{F}_2^n \to [-1/2, 1/2]$, is called the *bias* of a mask, and a mask with $\varepsilon_f(\alpha, \beta) \neq 0$ is called *biased*. The success of linear cryptanalysis depends on the size of the absolute bias of the mask used. Larger absolute biases are the most effective (see Section 2.1 for details), except for the trivial mask with $\varepsilon_f(\mathbf{0}, \mathbf{0}) = 1/2$, which provides no information. Therefore, finding nontrivial masks with a prescribed absolute bias is crucial for linear cryptanalysis.

Since there are overall 2^{m+n} masks, finding specific ones is assumed to be computationally hard in general. A standard approach to finding suitable masks is to calculate and check the bias of all masks using the fast Walsh transform (FWT) by Brown [24]. The exhaustive search utilizing the FWT has a time complexity of $\mathcal{O}((n \cdot T_f + m) \cdot 2^{m+n})$, where T_f denotes the cost of executing f (see Section 2.1 for details). In some cases, the time complexity of mask-seeking algorithms, e.g., the FWT, is acceptable for small vectorial Boolean functions $(m = n = 8)$ like S-boxes of substitution–permutation networks. However, it is assumed that there is no *classical* algorithm that finds suitable masks in plain generality within feasible time and space constraints.

An efficient *quantum* algorithm that may assist in finding appropriate masks for arbitrary vectorial Boolean functions is the Correlation Extraction Algorithm (CEA) [16,17]. Specifically, executing the CEA returns a biased mask, while the absolute bias of a mask monotonously relates to the probability of measuring it. Remarkably, simplified, the CEA computes the correlation, defined as $2 \cdot \varepsilon_f(\alpha, \beta)$, of all masks and encodes them into the amplitude of the quantum state using only one call to a quantum oracle that realizes f. Despite the fact that we cannot directly access the amplitudes of a quantum state, the CEA has an exponential speed-up in calculating the correlation of all masks compared to the *classical* standard approach with FWT.

Although we cannot directly access the correlations calculated by the CEA, the algorithm demonstrates the potential to enhance linear cryptanalysis. In the current literature, the CEA is used as a (sub-)routine in specific quantum versions of linear cryptanalysis [16–18] or property testing of vectorial Boolean functions [25,26]. However, none of the aforementioned works quantify the success rate of the CEA for measuring masks with a prescribed absolute bias, leaving this as an open problem that this paper addresses. Overall, this paper aims to analyze the capabilities and limitations of the CEA to understand its potential to find masks with a prescribed absolute bias for linear cryptanalysis.

1.1. Contribution

Our contribution investigates the capabilities of the CEA in finding a mask with a prescribed absolute bias within r executions. The following outlines the contributions of this paper.

- We formalize the probability of the CEA to output a mask with a prescribed absolute bias within r executions.
- We prove lower and upper bounds on r for a given fixed success probability to obtain at least one mask with a prescribed absolute bias.
- Based on the lower and upper bounds on r, we prove the asymptotic behavior of CEA regarding time, query, and memory complexity.
- Using CEA, we introduce an algorithm to find a mask with a prescribed absolute bias and prove its asymptotic behavior.
- We verify the behavior of CEA to measure a mask with a prescribed absolute bias by means of an empirical study on two vectorial Boolean functions with $m = 8$ input and $n = 8$ output bits each.
- We compare two approaches that use the FWT or CEA to find a mask with a prescribed absolute bias while discussing the potential of the CEA in linear cryptanalysis.

Overall, our contribution provides comprehensive insights about the CEA, enabling sound statements about its suitability in linear cryptanalysis.

1.2. Related Work

Malviya and Tiwari [16] introduced the algorithmic idea of the CEA for finding biased masks. Besides finding biased masks, CEA has applications in linear cryptanalysis using quantum computers: Malviya and Tiwari [18] presented a full quantum version of a linear cryptanalytic attack on an 8-bit SPN toy cipher using the CEA to find biased masks. Moreover, Hosoyamada [17] describes how to speed up multidimensional linear and integral distinguishers using the CEA. In addition to applying the CEA to attack a cipher, the CEA can also be used in the context of property testing of vectorial Boolean functions [25,26].

1.3. Organization

This paper begins by presenting the preliminaries for this work in Section 2, with some essential notations and definitions regarding linear cryptanalysis and the CEA. Section 3 then formalizes the success probability of the CEA to output a mask with a prescribed absolute bias. Section 4 analyzes the asymptotic behavior of the CEA, and Section 5 empirically verifies the statements from Sections 3 and 4. Afterward, Section 6 discusses the potential of the CEA to find masks with a prescribed absolute bias by comparing the CEA with a standard classical approach that uses the FWT. Ultimately, Section 7 shows the conclusions of this paper.

2. Preliminaries

2.1. Linear Cryptanalysis

Matsui [20] demonstrated that the data complexity, i.e., the number of plaintext–ciphertext pairs required to achieve a fixed success rate with Algorithm 1 depended on the bias of a mask. Table 1 highlights this relationship by displaying Table 2 of Matsui [20]. In particular, Table 1 illustrates the correlation between success rates, data complexity, and bias ε_f for Matsui's Algorithm 1. More precisely, Table 1 shows the reciprocal relation between absolute bias and data complexity.

Table 1. Data complexity values for specific success rates of Matsui's Algorithm 1 [20].

Success rate	84.1%	92.1%	97.7%	99.8%
Data complexity	$1/4 \cdot \varepsilon_f(\alpha,\beta)^{-2}$	$1/2 \cdot \varepsilon_f(\alpha,\beta)^{-2}$	$\varepsilon_f(\alpha,\beta)^{-2}$	$2 \cdot \varepsilon_f(\alpha,\beta)^{-2}$

Remark: The data complexity is the number of plaintext–ciphertext pairs needed to achieve a given success rate.

As a result of the relationship between bias and data complexity, performing linear cryptanalysis requires masks with a sufficiently large absolute bias. Problem 1 formalizes the search for such masks.

Problem 1. *For a function $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$, a uniformly distributed random variable X with sample space $\mathbb{F}_2^m$, a threshold $\tau \in (0, 1/2]$, and a bias $\varepsilon_f : \mathbb{F}_2^m \times \mathbb{F}_2^n \to [-1/2, 1/2]$, determine a nontrivial mask $(\alpha, \beta) \in \mathbb{F}_2^m \times \mathbb{F}_2^n$ that satisfies*

$$p_{\alpha,\beta} \coloneqq \Pr[\langle \alpha, X \rangle = \langle \beta, f(X) \rangle] = \frac{1}{2} + \varepsilon_f(\alpha, \beta) \tag{1}$$

with $\left|\varepsilon_f(\alpha, \beta)\right| \geq \tau$.

All algorithms that solve Problem 1 share the need to determine the bias of masks. This can be achieved by evaluating the Walsh transform of f. The following introduces the relation of Problem 1 to the Walsh transform, starting with its definition.

Definition 1 (Walsh transform (see [27]))**.** *The Walsh transform of $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$ is given by*

$$\mathrm{W}_f : \mathbb{F}_2^m \times \mathbb{F}_2^n \to \mathbb{Z}; (\alpha, \beta) \mapsto \sum_{x \in \mathbb{F}_2^m} (-1)^{\langle \alpha, x \rangle \oplus \langle \beta, f(x) \rangle}. \tag{2}$$

Next, using Definition 1, Theorem 1 establishes a relation between the bias and the Walsh transform of f at a given mask.

Theorem 1 (Calculation of $p_{\alpha,\beta}$)**.** *Let $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$ be a function and $(\alpha, \beta) \in \mathbb{F}_2^m \times \mathbb{F}_2^n$ a mask. Then, for $p_{\alpha,\beta}$, the following holds*

$$p_{\alpha,\beta} = \frac{1}{2} + \frac{\mathrm{W}_f(\alpha, \beta)}{2^{m+1}}$$

and therefore, $\varepsilon_f(\alpha, \beta) = \frac{\mathrm{W}_f(\alpha,\beta)}{2^{m+1}}$.

Proof of Theorem 1. Let X be a uniformly distributed random variable with sample space $\mathbb{F}_2^m$, $\mathcal{C}_f^{=}(\alpha, \beta) \coloneqq \left|\{x \in \mathbb{F}_2^m \mid \langle \alpha, x \rangle = \langle \beta, f(x) \rangle\}\right|$, and $\mathcal{C}_f^{\neq}(\alpha, \beta) \coloneqq \left|\{x \in \mathbb{F}_2^m \mid \langle \alpha, x \rangle \neq \langle \beta, f(x) \rangle\}\right|$. It holds that

$$\begin{aligned} p_{\alpha,\beta} &= \Pr[\langle \alpha, X \rangle = \langle \beta, f(X) \rangle] \\ &= \frac{\mathcal{C}_f^{=}(\alpha, \beta)}{2^m} \\ &= \frac{\mathcal{C}_f^{=}(\alpha, \beta) + \mathcal{C}_f^{=}(\alpha, \beta) + \mathcal{C}_f^{\neq}(\alpha, \beta) - \mathcal{C}_f^{\neq}(\alpha, \beta)}{2^{m+1}} \\ &= \frac{1}{2} + \frac{\mathrm{W}_f(\alpha, \beta)}{2^{m+1}}. \end{aligned}$$

Further, from the definition of $p_{\alpha,\beta}$, it follows

$$\varepsilon_f(\alpha, \beta) = \frac{\mathrm{W}_f(\alpha, \beta)}{2^{m+1}}.$$

□

Now, we focus on a quantity related to the bias, the *correlation* of f.

Definition 2. *The correlation of $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$ for $(\alpha, \beta) \in \mathbb{F}_2^m \times \mathbb{F}_2^n$ is given by*

$$\mathrm{C}_f(\alpha, \beta) \coloneqq \frac{\mathrm{W}_f(\alpha, \beta)}{2^m}. \tag{3}$$

Gathering implicit information about the bias, the correlations appear in the amplitudes of the quantum state of the CEA before measurement, as described in Section 2.2.

After showing the connection between the Walsh transform, correlation, and bias, we now give the time complexity of calculating the Walsh transform of f. To clarify, according to Theorem 1 and Definition 2, the correlation and bias depend on the Walsh transform. Since proving the time complexity for calculating Definition 1 is straightforward, Corollary 1 states its time complexity.

Corollary 1. *For $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$ with a time complexity of T_f and an arbitrary but fixed mask $(\alpha, \beta) \in \mathbb{F}_2^m \times \mathbb{F}_2^n$, the time complexity to calculate $\mathrm{W}_f(\alpha, \beta)$ according to Definition 1 is $\mathcal{O}((m + n + T_f) \cdot 2^m)$.*

Using Corollary 1 allows us to find the time complexity of the brute-force approach for solving Problem 1. Specifically, the brute-force approach for solving Problem 1 involves checking masks until a solution is found. In the worst case, the brute-force approach has a time complexity of $\mathcal{O}((m + n + T_f) \cdot 2^{2m+n})$. Further note that, in this article, the complexity analyses of classical algorithms are based on one-processor Real Random Access Machines (RRAMs). To clarify, in this paper, complexity analyses involving RRAMs that use basic arithmetic operations on reals ($\mathbb{R}$), such as addition and multiplication, have a complexity of $O(1)$.

In addition to the brute-force approach, we discuss Brown's FWT algorithm [24], a classical algorithm that can be used to solve Problem 1. Before moving on to further details, we introduce the following convention: let $f(0, \ldots, 0), f(0, \ldots, 1), \ldots, f(1, \ldots, 1)$ denote the evaluations of f for all $x \in \mathbb{F}_2^m$, ordered with respect to the binary input vectors in ascending order. For a fixed $\beta \in \mathbb{F}_2^n$, using the vector

$$v := \left((-1)^{\langle f(0,\ldots,0),\beta\rangle}, (-1)^{\langle f(0,\ldots,1),\beta\rangle}, \ldots, (-1)^{\langle f(1,\ldots,1),\beta\rangle}\right)^T \tag{4}$$

as input, the FWT computes the vector

$$w := \left(\mathrm{W}_f((0\ldots0),\beta), \mathrm{W}_f((0\ldots1),\beta), \ldots, \mathrm{W}_f((1\ldots1),\beta)\right)^T \tag{5}$$

as output. As displayed in Equation (5), the FWT calculates the Walsh transforms for all $\alpha \in \mathbb{F}_2^m$ and a fixed β. Thus, to solve Problem 1, executing the FWT for all $\beta \in \mathbb{F}_2^n \setminus \{\mathbf{0}\}$ is required in the worst case.

Based on this idea, Algorithm 1 specifies an approach to solve Problem 1 utilizing the FWT. For better understanding, we briefly introduce two essential notational conventions. By interpreting $\alpha \in \mathbb{F}_2^m$ as an integer, w_α denotes the entry α of the 2^m dimensional vector w. In addition, $\mathrm{FWT}(v)$ denotes the execution of the FWT for the 2^m dimensional vector as input. Subsequently, Theorem 2 gives the time complexity of Algorithm 1.

Theorem 2. *For $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$ with a time complexity of T_f, solving Problem 1 using Algorithm 1 has a worst-case time complexity of $\mathcal{O}((n \cdot T_f + m) \cdot 2^{m+n})$, an oracle complexity of $\mathcal{O}(2^{m+n})$, and a space complexity of $\mathcal{O}(2^m)$.*

Proof of Theorem 2. Calculating the time complexity of Algorithm 1 is straightforward. However, for completeness, we list the complexity of each line and then combine them. The following presents the time complexities of lines 1–11 for only one arbitrary but fixed $\beta \in \mathbb{F}_2^n \setminus \{\mathbf{0}\}$:

Line 1: $\mathcal{O}(m + n)$;
Line 2: $\mathcal{O}(m + n)$;
Line 3: $\mathcal{O}(n \cdot T_f \cdot 2^m)$;
Line 4: $\mathcal{O}(m \cdot 2^m)$ (according to Brown [24]);

Lines 5–11: The iteration considers 2^m values for α. For each value, evaluating the condition of line 5 has a time complexity of $\mathcal{O}(m)$. Thus, checking all entries of w gives the worst-case complexity of $O(m \cdot 2^m)$.

In the worst case lines 2–11 must be executed for all $\beta \in \mathbb{F}_2^n \setminus \{\mathbf{0}\}$, leading to a time complexity of $\mathcal{O}((n \cdot T_f + m) \cdot 2^{m+n})$.

Algorithm 1: Pseudocode description of solving Problem 1 using the FWT

Input: A vectorial Boolean function $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$ and a threshold $\tau \in (0, 1/2]$.

1 $res \leftarrow (\mathbf{0}, \mathbf{0})$;
2 **for** *all* $\beta \in \mathbb{F}_2^n \setminus \{\mathbf{0}\}$ **do**
3 $\quad v \leftarrow \left((-1)^{\langle f(0,\ldots,0),\beta\rangle}, (-1)^{\langle f(0,\ldots,1),\beta\rangle}, \ldots, (-1)^{\langle f(1,\ldots,1),\beta\rangle}\right)^T$;
4 $\quad w = \left(\mathrm{W}_f((0\ldots0),\beta), \mathrm{W}_f((0\ldots1),\beta), \ldots, \mathrm{W}_f((1\ldots1),\beta)\right)^T \leftarrow \mathrm{FWT}(v)$;
5 $\quad$ **for** *all* $\alpha \in \mathbb{F}_2^m$ **do**
6 $\quad\quad$ **if** $\left|w_\alpha / 2^{m+1}\right| \geq \tau$ **then**
7 $\quad\quad\quad res \leftarrow (\alpha, \beta)$;
8 $\quad\quad\quad$ break;
9 $\quad\quad$ **end**
10 $\quad$ **end**
11 **end**

Output: *res*

For the oracle complexity, we require 2^m oracle calls to initialize v in line 1, which is repeated for all $\beta \in \mathbb{F}_2^n \setminus \{\mathbf{0}\}$, giving the oracle complexity of $\mathcal{O}(2^{m+n})$.

The space complexity is again determined solely by the initialization of v in line 1. Brown's FWT algorithm requires $\mathcal{O}(1)$ additional space, as its calculations involving v are in-place [24]. Thus, the space complexity is $\mathcal{O}(2^m)$.

Finally, we show how Algorithm 1 solves Problem 1. According to Theorem 1, line 5 finds a mask that solves Problem 1. Thus, if one exists, Algorithm 1 returns a mask that solves Problem 1. □

2.2. The Correlation Extraction Algorithm (CEA)

The algorithm presented in this section appears in several publications [17,18,25,26] with different naming conventions. This paper uses the name Correlation Extraction Algorithm from Hosoyamada [17]. However, before we move on to the quantum circuit of the CEA, we introduce some notational conventions related to quantum computing.

A ket $|\phi\rangle \in \mathbb{C}^{2^m}$ denotes the quantum state of an m-qubit quantum register. In particular, $|0\rangle^{\otimes m}$ specifies the zero state on such a register according to the standard basis. Moreover, this paper utilizes the convention to label the vectors of quantum registers with binary vectors, i.e., $|\alpha\rangle = \bigotimes_i |\alpha_i\rangle$ with $\alpha_i \in \{0, 1\}$ for all i. As usual in quantum computing, the operators $H^{\otimes m}$ and $I^{\otimes m}$, for example, describe the Hadamard and identity gate for the whole m-qubit register, respectively. Finally, using $\oplus$ as the bitwise XOR, $U_f : \mathbb{C}^{2^{m+n}} \to \mathbb{C}^{2^{m+n}}$, $|x\rangle |y\rangle \mapsto |x\rangle |y \oplus f(x)\rangle$, defines the quantum oracle version of f.

Using the notational conventions related to quantum computing, Algorithm 2 specifies the CEA in pseudocode, and Figure 1 illustrates the quantum circuit of the CEA. In particular, the primary purpose of the CEA is to measure biased masks for vectorial Boolean functions. Specifically, the CEA exclusively outputs masks $(\alpha, \beta) \in \mathbb{F}_2^m \times \mathbb{F}_2^n$ with $\varepsilon_f(\alpha, \beta) \neq 0$, as the probability of measuring a mask relates to its absolute bias.

Algorithm 2: Pseudocode description of the CEA

Input: A quantum oracle $U_f : \mathbb{C}^{2^{m+n}} \to \mathbb{C}^{2^{m+n}}, |x\rangle |y\rangle \mapsto |x\rangle |y \oplus f(x)\rangle$, of $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$.

1 Initialize an m- and n-qubit register with $|\psi_1\rangle = |0\rangle^{\otimes m} |0\rangle^{\otimes n}$;
2 $|\psi_2\rangle = (H^{\otimes m} \otimes I^{\otimes n}) |\psi_1\rangle$;
3 $|\psi_3\rangle = U_f |\psi_2\rangle$;
4 $|\psi_4\rangle = (H^{\otimes m} \otimes H^{\otimes n}) |\psi_3\rangle$;
5 Measure $|\psi_4\rangle$ in the standard basis and obtain a mask $(\alpha, \beta) \in \mathbb{F}_2^m \times \mathbb{F}_2^n$;

Output: (α, β)

Based on the CEA specification, we highlight the quantum state of the CEA before measurement, explicitly $|\psi_4\rangle$, as displayed in Figure 1. In essence, Section 3 utilizes the state $|\psi_4\rangle$ to describe the probability of measuring a mask that solves Problem 1. However, recalling the relationship between the bias and correlation of f, expressed by $2 \cdot \varepsilon_f(\alpha, \beta) = \mathrm{C}_f(\alpha, \beta)$, Theorem 3 specifies $|\psi_4\rangle$ by underlining its relation to $\mathrm{C}_f(\alpha, \beta)$.

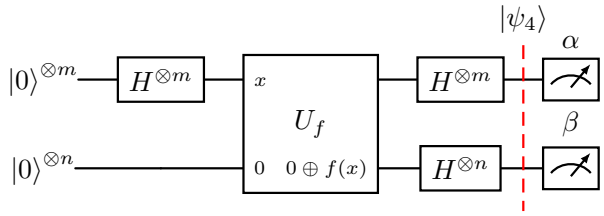

Figure 1. The quantum circuit of the CEA.

Theorem 3 ([17] Proposition 3)**.** *The quantum state of the CEA before the measurement is*

$$|\psi_4\rangle = \frac{1}{\sqrt{2^n}} \sum_{\alpha,\beta \in \mathbb{F}_2^m \times \mathbb{F}_2^n} \mathrm{C}_f(\alpha, \beta) |\alpha\rangle |\beta\rangle . \tag{6}$$

The proof of Theorem 3 has already been provided by Hosoyamada [17]. Nonetheless, we restate it here as it serves as a step-by-step analysis of Algorithm 2.

Proof of Theorem 3. The following proof is a detailed step-by-step analysis of Algorithm 2. Note that the line numbers from Algorithm 2 correspond to those in the following enumeration. Further, let $a \in \{0,1\}^m$ and $b \in \{0,1\}^n$ be bitstrings and $a|b$ denote their concatenation.

1. Initialize an m- and n-qubit register with $|\psi_1\rangle = |0\rangle^{\otimes m} |0\rangle^{\otimes n}$
2. $|\psi_2\rangle = (H^{\otimes m} \otimes I^{\otimes n}) |\psi_1\rangle = \frac{1}{\sqrt{2^m}} \sum_{x \in \mathbb{F}_2^m} |x\rangle |0\rangle^{\otimes n}$
3. $|\psi_3\rangle = U_f |\psi_2\rangle = \frac{1}{\sqrt{2^m}} \sum_{x \in \mathbb{F}_2^m} |x\rangle |f(x)\rangle$
4. After applying $(H^{\otimes m} \otimes H^{\otimes n})$ to $|\psi_3\rangle$, the register $|\psi_4\rangle$ is in the state

$$|\psi_4\rangle = \left(H^{\otimes m} \otimes H^{\otimes n}\right) \frac{1}{\sqrt{2^m}} \sum_{x \in \mathbb{F}_2^m} |x\rangle |f(x)\rangle) \tag{7}$$

$$= \frac{1}{\sqrt{2^{2m+n}}} \sum_{x \in \mathbb{F}_2^m} \sum_{\mu \in \mathbb{F}_2^{m+n}} (-1)^{\langle \mu, x|f(x)\rangle} |\mu\rangle . \tag{8}$$

For $\alpha, \beta \in \mathbb{F}_2^m \times \mathbb{F}_2^n$, let $\mu = \alpha|\beta$. Thus, Equation (8) can be rearranged as

$$|\psi_4\rangle = \frac{1}{\sqrt{2^{2m+n}}} \sum_{x \in \mathbb{F}_2^m} \sum_{(\alpha,\beta) \in \mathbb{F}_2^m \times \mathbb{F}_2^n} (-1)^{\langle \alpha|\beta, x|f(x)\rangle} |\alpha\rangle |\beta\rangle \tag{9}$$

$$= \frac{1}{\sqrt{2^{2m+n}}} \sum_{(\alpha,\beta) \in \mathbb{F}_2^m \times \mathbb{F}_2^n} \sum_{x \in \mathbb{F}_2^m} (-1)^{\langle \alpha, x\rangle \oplus \langle \beta, f(x)\rangle} |\alpha\rangle |\beta\rangle . \tag{10}$$

According to Definition 1, $\mathrm{W}_f(\alpha, \beta) = \sum_{x \in \mathbb{F}_2^m} (-1)^{\langle \alpha, x\rangle \oplus \langle \beta, f(x)\rangle}$ holds and thus

$$|\psi_4\rangle = \frac{1}{\sqrt{2^n}} \sum_{(\alpha,\beta) \in \mathbb{F}_2^m \times \mathbb{F}_2^n} \frac{\mathrm{W}_f(\alpha, \beta)}{2^m} |\alpha\rangle |\beta\rangle . \tag{11}$$

Finally, Definition 2 leads to

$$|\psi_4\rangle = \frac{1}{\sqrt{2^n}} \sum_{(\alpha,\beta) \in \mathbb{F}_2^m \times \mathbb{F}_2^n} \mathrm{C}_f(\alpha, \beta) |\alpha\rangle |\beta\rangle . \tag{12}$$

□

Using the proof of Theorem 3, we highlight the techniques used in the CEA and its time complexity. In particular, step 3 shows that applying U_f to the superposition of all $x \in \mathbb{F}_2^m$ evaluates f for all inputs with one single oracle call (see Section 4 for a discussion of the time complexity of U_f). Further, utilizing $(H^{\otimes m} \otimes H^{\otimes n})$, step 4 encodes the correlation of f for all masks in the amplitudes of the quantum state. As shown by Malviya and Tiwari [16], encoding the correlations of f for all masks with the CEA has an asymptotic time complexity that depends only on the depth of the circuit of U_f.

As a result of Theorem 3, measuring $|\psi_4\rangle$ using the standard basis, Algorithm 2 outputs a mask $(\alpha, \beta) \in \mathbb{F}_2^m \times \mathbb{F}_2^n$, whereby the probability of measuring it depends on its absolute correlation. To clarify, according to $2 \cdot \varepsilon_f(\alpha, \beta) = \mathrm{C}_f(\alpha, \beta)$, the greater the absolute bias of a mask, the more likely it is to be measured. Therefore, the CEA theoretically does not output masks with $\varepsilon_f(\alpha, \beta) = 0$. Sections 3 and 4 examine the relationship between the absolute bias of a mask and the probability of measuring it with the CEA, demonstrating that the CEA can help address Problem 1.

3. Formalizing the Success Probability of the CEA

This section elaborates on the CEA, focusing on its probability of measuring a mask that solves Problem 1. More precisely, it investigates the success probability of measuring at least one mask that solves Problem 1 using $r \geq 1$ CEA samples. Describing the success probability of the CEA relies on the set S of all masks that solve Problem 1, formally introduced in Definition 3.

Definition 3. *The set of masks solving Problem 1, for $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$ and $\tau \in (0, 1/2]$, is given by*

$$S \coloneqq \left\{ (\alpha', \beta') \in \mathbb{F}_2^m \times \mathbb{F}_2^n \middle| \left| \varepsilon_f(\alpha', \beta') \right| \geq \tau \wedge (\alpha', \beta') \neq (\mathbf{0}, \mathbf{0}) \right\}. \tag{13}$$

Incorporating the quantum state of the CEA before measuring a mask, as stated in Theorem 3, and the set S of masks that solve Problem 1, Theorem 4 specifies the probability of measuring at least one mask that solves Problem 1.

Theorem 4. *For $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$, $\tau \in (0, 1/2]$ and r samples $(\alpha, \beta)^{(1)}, \ldots, (\alpha, \beta)^{(r)} \in \mathbb{F}_2^m \times \mathbb{F}_2^n$ from the CEA, the probability to solve Problem 1 with at least one sample is*

$$\Pr\left[\bigvee_{i=1}^{r} (\alpha, \beta)^{(i)} \in S \right] = 1 - \left(1 - \frac{1}{2^n} \sum_{(\alpha,\beta) \in S} \mathrm{C}_f(\alpha, \beta)^2 \right)^r . \tag{14}$$

Proof of Theorem 4. In order to solve Problem 1, only masks with $|\varepsilon_f(\alpha,\beta)| \geq \tau$ are relevant. Thus, from Theorem 3, it follows

$$\Pr\Big[\Big|\varepsilon_f(\alpha,\beta)\Big| \geq \tau\Big] = \frac{1}{2^n} \sum_{(\alpha,\beta)\in S} \mathrm{C}_f(\alpha,\beta)^2, \tag{15}$$

which is the probability that a single execution of the CEA solves Problem 1. Further, utilizing Equation (15), the probability of solving Problem 1, with r repeated independent executions of the CEA, is given by

$$\Pr\left[\bigvee_{i=1}^{r} (\alpha,\beta)^{(i)} \in S\right] = 1 - \left(1 - \frac{1}{2^n} \sum_{(\alpha,\beta)\in S} \mathrm{C}_f(\alpha,\beta)^2\right)^r. \tag{16}$$

□

4. Complexity Analysis of the CEA

The CEA is a probabilistic algorithm that outputs biased masks. In detail, Theorem 4 shows that the probability of measuring a mask with bias threshold τ depends on the number of nontrivial masks of f that satisfy τ and the magnitude of τ. Thus, analyzing the complexity of the CEA requires estimating the number of CEA samples needed to measure a mask, which solves Problem 1 with a fixed probability.

Estimating the lower and upper bounds on the number of CEA samples uses the following definitions of the worst and best cases to solve Problem 1. According to Theorem 4 and Definition 3, the best case occurs when

$$\frac{1}{2^n} \sum_{(\alpha,\beta)\in S} \mathrm{C}_f(\alpha,\beta)^2 = \frac{2^n - 1}{2^n}. \tag{17}$$

To clarify, Equation (17) follows by excluding the trivial mask. In contrast, we define the worst case by conservatively limiting the number of elements in the set S to $|S| = 1$. As a remark, the conservative definition of the worst case ensures that estimating the upper bound of CEA samples includes the actual worst case. Using the best and worst cases, Theorem 5 describes the lower and the upper bounds of the number of CEA samples to measure a mask, which solves Problem 1 with a fixed probability.

Theorem 5. *Let $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$, $\tau \in (0, 1/2]$ and $(1-\delta) \in (0,1)$. Then, there is a number r of CEA samples, satisfying*

$$\frac{\log(\delta^{-1})}{\log(2^n)} \leq r \leq \frac{\log(\delta)}{\log(1 - 4\tau^2/2^n)}, \tag{18}$$

such that at least one sample solves Problem 1 with probability $(1-\delta)$.

Proof of Theorem 5. From Theorem 4, it follows that

$$1 - \left(1 - \frac{1}{2^n} \sum_{(\alpha,\beta)\in S} \mathrm{C}_f(\alpha,\beta)^2\right)^r = 1 - \delta. \tag{19}$$

To give an upper bound of samples, it is sufficient to assume the worst case $|S| = 1$, in which only one nontrivial mask with $|\varepsilon_f(\alpha,\beta)| \geq \tau$ exists. For $|S| = 1$, the following holds:

$$1 - \left(1 - \frac{4\tau^2}{2^n}\right)^r = 1 - \delta \iff r = \frac{\log(\delta)}{\log(1 - 4\tau^2/2^n)}. \tag{20}$$

The lower bound of samples is derived by the best case for the CEA. Specifically, the best case follows from Theorem 4, and excluding the trivial mask yields

$$\frac{1}{2^n}\sum_{(\alpha,\beta)\in S}\mathrm{C}_f(\alpha,\beta)^2=\frac{2^n-1}{2^n}. \tag{21}$$

Hence, in the best case, the CEA needs

$$1-\left(1-\left(\frac{2^n-1}{2^n}\right)\right)^r=1-\delta \iff r=\frac{\log(\delta^{-1})}{\log(2^n)} \tag{22}$$

samples. From Equations (20) and (22), it follows that

$$\frac{\log(\delta^{-1})}{\log(2^n)}\leq r\leq\frac{\log(\delta)}{\log(1-4\tau^2/2^n)}. \tag{23}$$

□

Next, using the bounds of Theorem 5, this section provides the asymptotic number of samples required by the CEA to measure at least one mask that solves Problem 1, with Corollary 2 and Theorem 6 describing the best- and worst-case scenarios, respectively.

Corollary 2. *In the best case, with a probability of* $(1-\delta)\in(0,1)$*, at least one of* $\mathcal{O}\left(\log\left(\delta^{-1}\right)/n\right)$ *CEA samples solves Problem 1.*

Theorem 6. *In the worst case, with a probability of* $(1-\delta)\in(0,1)$ *and* $\tau\in(0,1/2]$*, at least one of* $\mathcal{O}\left(\log\left(\delta^{-1}\right)\cdot 2^n/\tau^2\right)$ *CEA samples solves Problem 1.*

Proof of Theorem 6. This direct proof consists of two stages. The first stage presents an upper bound for Theorem 5 suitable for asymptotic estimation. The second stage then demonstrates the asymptotic upper bound using the bound from the first stage.

For the upper bound of Theorem 5, this paper chooses an additional upper bound. Specifically, for $x\in(0,1)$ and $\delta\in(0,1)$, the following holds:

$$\log(1-x)<-x \iff \frac{\log(\delta)}{\log(1-x)}<\frac{\log(\delta^{-1})}{x}. \tag{24}$$

Next, substituting x in Equation (24) with $4\tau^2/2^n$ leads to

$$\frac{\log(\delta)}{\log(1-4\tau^2/2^n)}<\frac{\log(\delta^{-1})\cdot 2^n}{4\tau^2}. \tag{25}$$

From the upper bound of Theorem 5 given in Equation (25), the asymptotic bound is given by

$$\frac{\log(\delta^{-1})\cdot 2^n}{4\tau^2}=\mathcal{O}\left(\frac{\log(\delta^{-1})\cdot 2^n}{\tau^2}\right). \tag{26}$$

In conclusion, the asymptotic upper bound of samples needed to measure at least one mask that solves Problem 1 is $\mathcal{O}(\log(\delta^{-1})\cdot 2^n/\tau^2)$. □

With the asymptotic bounds for the best- and worst-case scenarios, this paper now analyzes the query, memory, and time complexity of the CEA to measure at least one mask that solves Problem 1. Therefore, this paper discusses three quantum measures associated with the corresponding complexity in the gate model. We start our analysis with the query complexity of the CEA.

When estimating the query complexity of the CEA, it is necessary to consider the number of executions of its corresponding quantum oracle U_f. The asymptotic number of CEA samples that contain, with a probability of $(1-\delta)$, at least one mask that solves Problem 1 follows from Corollary 2 and Theorem 6. Consequently, the CEA needs $\mathcal{O}\left(\log\left(\delta^{-1}\right)/n\right)$ quantum oracle calls in the best case and $\mathcal{O}\left(\log\left(\delta^{-1}\right)\cdot 2^n/\tau^2\right)$ quantum oracle calls in the worst case.

Extending the analysis of complexities, this paper now examines the number of qubits of the CEA. In particular, the number of qubits the CEA requires depends on the number of input and output bits of f. As specified in Algorithm 2, the CEA needs $\mathcal{O}(m+n)$ qubits.

Next, this paper discusses the time complexity of the CEA. Since the vectorial Boolean function f and its quantum oracle are unknown, T_f and T_{U_f} are placeholders of their circuit depth, respectively. According to Kitaev et al. [28] (Thm. 7.3), every function f is realizable as a quantum oracle U_f with $T_{U_f} = \mathcal{O}(T_f)$. Using the placeholder T_f and incorporating the asymptotic query complexity of the CEA, the time complexity of the CEA is $\mathcal{O}(\log(\delta^{-1})\cdot T_f/n)$ in the best case and $\mathcal{O}(\log(\delta^{-1})\cdot 2^n\cdot T_f/\tau^2)$ in the worst case.

Table 2 summarizes the asymptotic time, query, and memory complexities of the CEA for measuring masks with a prescribed absolute bias τ, illustrating that query and time complexity depend mainly on the choice of f and τ. In detail, the success probability of the CEA according to Theorem 4, the prescribed absolute bias τ, the time complexity T_f, and the number of output bits of f affect its asymptotic behavior. In addition, the success probability of the CEA has only a logarithmic impact, allowing the success probabilities to be exponentially close to one. Consequently, in the worst case, the capabilities of the CEA to measure masks for solving Problem 1 in practice are limited to functions with polynomial time ($T_f = \mathrm{poly}(m+n)$), a relatively small output size, e.g., $n \leq 32$, and at least one nontrivial mask (α,β) with a constant absolute bias, e.g., $|\varepsilon_f(\alpha,\beta)| = \tau = 1/2$.

Table 2. Asymptotic complexities of the CEA for measuring at least one mask that solves Problem 1.

Complexity	Best Case	Worst Case
Time	$\mathcal{O}\left(\frac{\log(\delta^{-1})\cdot T_f}{n}\right)$	$\mathcal{O}\left(\frac{\log(\delta^{-1})\cdot T_f\cdot 2^n}{\tau^2}\right)$
Quantum queries	$\mathcal{O}\left(\frac{\log(\delta^{-1})}{n}\right)$	$\mathcal{O}\left(\frac{\log(\delta^{-1})\cdot 2^n}{\tau^2}\right)$
Qubits	$\mathcal{O}(m+n)$	

Remark: Let $(1-\delta)\in(0,1)$ be the success probability of the CEA, $\tau\in(0,1/2]$ the prescribed absolute bias, and T_f the time complexity of f.

Having established the complexity of the CEA for measuring a mask that addresses Problem 1, Algorithm 3 presents an approach that resolves Problem 1 using the CEA. The approach is straightforward, as it generates masks using the CEA and checks them to solve Problem 1. Hence, the aforementioned limitations of the CEA in measuring a mask that solves Problem 1 also affect Algorithm 3. Subsequently, Theorem 7 states the time and oracle complexities, as well as the classical space complexity and the quantum space complexity of Algorithm 3. By convention, we treat a classical computation step and the application of a quantum gate as equivalent. The reason for merging the oracle complexities is due to our treatment of the given classical and quantum oracles as equivalent in terms of complexity. However, we differentiate between the space and the qubit complexities to highlight the advantage the CEA provides in terms of bits and qubits used, which is exponential in comparison with, e.g., the classical solution using Brown's FWT algorithm Algorithm 1 (see Section 6).

Algorithm 3: Pseudocode description of solving Problem 1 using the CEA

Input: A vectorial Boolean function $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$, its quantum oracle version $U_f : \mathbb{C}^{2^{m+n}} \to \mathbb{C}^{2^{m+n}}$, $|x\rangle |y\rangle \mapsto |x\rangle |y \oplus f(x)\rangle$, a threshold $\tau \in (0, 1/2]$, and a success probability $\delta \in (0, 1)$.

1 $res \leftarrow (\mathbf{0}, \mathbf{0})$;
2 $r \leftarrow \frac{\log(\delta^{-1}) \cdot 2^{n-2}}{\tau^2}$;
3 **for** *i=1;* $i \leftarrow i + 1$ **to** r **do**
4 $(\alpha, \beta) \leftarrow$ execute CEA with U_f;
5 **if** $\left|\mathrm{W}_f(\alpha, \beta)/2^{m+1}\right| \geq \tau$ *and* $(\alpha, \beta) \neq (\mathbf{0}, \mathbf{0})$ **then**
6 res $\leftarrow (\alpha, \beta)$;
7 break;
8 **end**
9 **end**

Output: *res*

Theorem 7. *For $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$ and its quantum oracle $U_f : \mathbb{C}^{2^{m+n}} \to \mathbb{C}^{2^{m+n}}$ with time complexities $\mathcal{O}(T_f) = \mathrm{poly}(n + m)$, a threshold $\tau \in (0, 1/2]$, and $\delta \in (0, 1)$, Algorithm 3 solves Problem 1 with a success probability of $(1 - \delta)$, a time complexity of $\mathcal{O}(\log(\delta^{-1}) \cdot T_f \cdot 2^{m+n}/\tau^2)$, an oracle complexity of $\mathcal{O}(\log(\delta^{-1}) \cdot 2^{m+n}/\tau^2)$, a classical space complexity of $\mathcal{O}(m + n)$, and $\mathcal{O}(m + n)$ qubits.*

The reason behind setting $\mathcal{O}(T_f) = \mathrm{poly}(n + m)$ is that it simplifies the proof of Theorem 7 and makes the resulting asymptotic time complexity more compact.

Proof of Theorem 7. Calculating the time complexity of Algorithm 3 is straightforward. Due to $T_{U_f} = \mathcal{O}(T_f) = \mathrm{poly}(n + m)$, the time complexity of lines 1–9 is clearly dominated by the direct calculation of $\mathrm{W}_f(\alpha, \beta)$ in line 5 and the for loop in line 3. According to Corollary 1 and $\mathcal{O}(T_f) = \mathrm{poly}(n + m)$, line 5 has a time complexity of $\mathcal{O}(T_f \cdot 2^m)$. Further, in the worst case, line 5 is repeated r times. Thus, in the worst case, Algorithm 3 has a time complexity of $\mathcal{O}(\log(\delta^{-1}) \cdot T_f \cdot 2^{m+n}/\tau^2)$.

Next, we investigate the oracle complexity of Algorithm 3. In detail, Algorithm 3 calls f in line 5 and U_f in line 4. Due to Corollary 1, line 5 dominates with $\mathcal{O}(2^m)$ calls of f, which is repeated r times in the worst case. As r is given directly in line 2, we conclude the oracle complexity of $\mathcal{O}(\log(\delta^{-1}) \cdot 2^{m+n}/\tau^2)$.

Now, we estimate the space complexity of Algorithm 3. We highlight that the computation of the bias in line 5 would be performed in place. Therefore, the dominating factor for the classical space complexity is the space needed for the mask pair *res*, giving the space complexity as $\mathcal{O}(m + n)$.

Subsequently, we prove the number of qubits used by Algorithm 3. With line 4, Algorithm 3 executes the CEA, determining its required qubits. As shown in Table 2, the CEA uses $\mathcal{O}(m + n)$ qubits, as does Algorithm 3.

Last but not least, we discuss the probability of success of Algorithm 3. Specifically, Algorithm 3 succeeds if the expression in line 5 is evaluated as true. In particular, according to Theorem 1, the inequation in line 5 is true if and only if, in line 4, the CEA measures a mask that solves Problem 1. Thus, the success probability of Algorithm 3 directly depends on the probability of the CEA to measure at least one mask that solves Problem 1. Hence, using the upper bound of Theorem 5, the number of samples r in line 2 reflects the success probability of Algorithm 3. □

5. Empirical Study of the CEA

This paper complements the theoretical work of Sections 3 and 4 by presenting an empirical study of specific vectorial Boolean functions to verify the capabilities of the

CEA in measuring a mask that solves Problem 1. Since the probability of success of the CEA depends on the distribution of the Walsh transform of the vectorial Boolean function, this paper conducted a detailed study of two functions with different Walsh transform distributions. Figure 2 displays the distribution of the Walsh transform of the examined functions, the AES S-box [29] (Section 5.1.1) and the 8-bit XOR cipher with a fixed key $k \in \mathbb{F}_2^8$, defined as $\mathrm{XOR} : \mathbb{F}_2^8 \times \mathbb{F}_2^8; x \mapsto x \oplus k$.

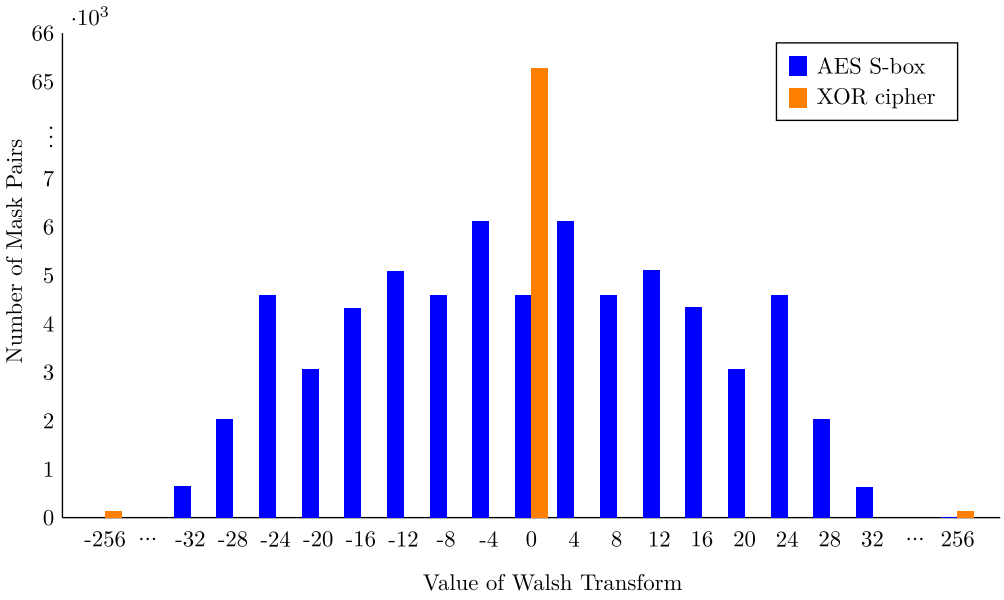

Figure 2. Distribution of the Walsh transform of the AES S-box and 8-bit XOR cipher.

5.1. Methodology

This study conducted controlled randomized experiments that executed the CEA repeatedly to measure the number of samples required to obtain a mask with an absolute bias greater than or equal to a fixed threshold $\tau \in [0, 1/2]$ concerning the utilized vectorial Boolean function. This approach aimed to validate Theorem 4 by comparing the experimental results with the theoretical prediction.

Selection of functions: We chose the AES S-box and an 8-bit XOR cipher as the target functions due to their different Walsh transform distributions. In particular, according to Figure 2, the distribution of the Walsh transforms of the AES S-box reminds us of a normal distribution, and the Walsh transforms of the XOR cipher are concentrated on zero, with relatively few exceptions. Further, as can be deduced from Theorem 4, according to their different Walsh transform distributions, the probabilities of success of these functions for finding a nontrivial mask with the lowest possible absolute bias differ significantly. Hence, by analyzing the XOR cipher and the AES S-box, we gain insight into the behavior of the CEA for measuring a mask with a prescribed absolute bias, which is relatively likely or unlikely, respectively.

Carrying out the experiments: Given the theoretical focus of this paper, we tested the CEA in a simulation using a classical computer. This simulation served as a practical application of our theoretical work, allowing us to observe the behavior of the CEA in a controlled environment.

Parameter initialization: Each function obtained a seed, defined as the sum of the first three digits of a transcendent number and a random integer (see the experimental setup for details), ensuring the exact reproducibility of the experiment. Further, the threshold value for the bias τ for nontrivial masks was determined based on the highest achievable absolute bias for each function.

Experiments and data collection: We conducted 3000 independent experiments each for the AES S-box and the XOR cipher. Each experiment repeatedly executed the CEA until a returned mask solved Problem 1 while storing each mask.

Analysis: Based on the results of the experiments, we used the Clopper–Pearson intervals [30] to provide evidence that the probability of the experiments differed only by a small constant from the probability in Theorem 4.

5.2. Experimental Setup

Algorithm 4 specifies the experiments performed for the AES S-box and the XOR cipher with the integer 89 as a fixed key, where f is a placeholder for the functions.

According to Algorithm 4, $\varsigma = \lfloor \pi \cdot 10^3 \rfloor + 0\text{xCA684ACA}$ denotes the seed for the AES S-box, and $\varsigma = \lfloor e \cdot 10^3 \rfloor + 0\text{xABA640E0}$ is the seed for the XOR cipher. The random seed values were generated using the free service from random.org. Further, the highest threshold value for τ was $1/16$ in the case of the AES S-box and $1/2$ in the case of the XOR cipher.

The experiments were implemented in Python using the Qiskit software (Version Number: 3) stack [31] for quantum computing, and the code is available on GitHub [32]. In order to optimize the code to run efficiently on standard consumer hardware, we addressed the implementation of the quantum oracle U_f using sparse matrices from SciPy [33]. We executed the experiments using an AMD Ryzen 7 PRO 6850U in about 5 min and 47 s.

During the simulation of the CEA, we initialized the quantum register with Qiskit and then realized the rest of the CEA in a custom simulation. To clarify, U_f was applied by manually multiplying U_f by the state vector. Subsequently, similarly to U_f, we manually applied $H^{\otimes(m+n)}$ to the state vector. Ultimately, we performed the measurement by sampling the mask according to the weights in the state vector using the standard random.choices function of Python.

Algorithm 4: Generic pseudocode description of the experiments

Input: $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$
Choose a seed ς for f;
Set $N = 3000$;
Set τ to the highest threshold value for f regarding Problem 1;
for $i \leftarrow 1$ **to** N **do**
    $\varsigma \leftarrow \varsigma + 1$;
    found $\leftarrow$ false;
    while not *found* **do**
        $(\alpha, \beta) \leftarrow$ execute CEA for f;
        store $(i, (\alpha, \beta), \mathrm{W}_f(\alpha, \beta), \varepsilon_f(\alpha, \beta))$;
        if $\left|\varepsilon_f(\alpha, \beta)\right| \geq \tau$ *and* $(\alpha, \beta) \neq (0, 0)$ **then**
            found $\leftarrow$ true ;
        end
    end
end

5.3. Results

Figure 3 displays a boxplot that presents the results of the experiments for the AES S-box and the XOR cipher, with the lower whisker set at one and the upper whisker set at 99, representing the number of CEA measurements required to obtain a mask that solved Problem 1. For the AES S-box, we had a minimum value of one, a lower quartile (25th percentile) of 18, a median (50th percentile) of 36, an upper quartile (75th percentile) of 53, and a maximum value of 99. In addition, Figure 4 illustrates the theoretical probability of Theorem 4 and the empirically determined probability from the experiments with the AES S-box to measure a mask that solves Problem 1 with r samples, whereby there is not much visual difference. Further, for the XOR cipher, we had 2982 experiments with one single measurement and 18 experiments with two measurements, while the empirically estimated

probability was close to the probability in Theorem 4. Overall, the results were in line with our theoretical work.

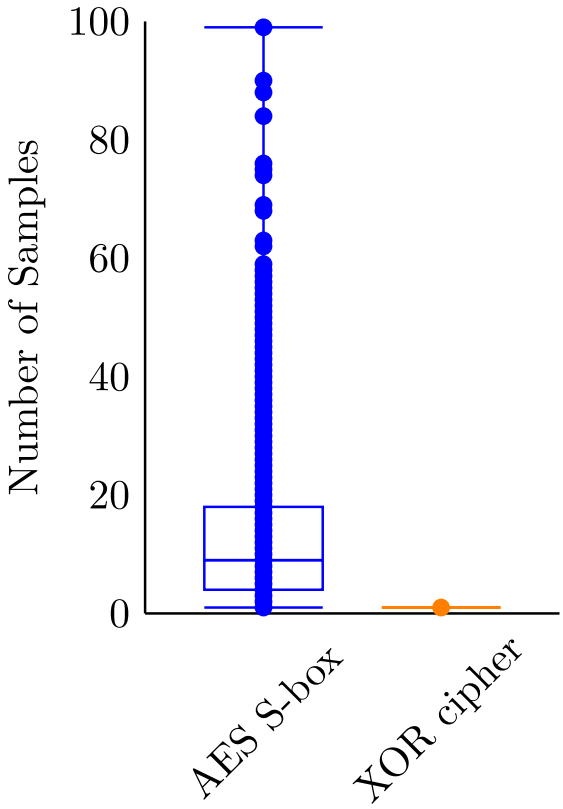

Figure 3. Boxplot of 3000 experiments, where each point marks the number of samples required for an experiment.

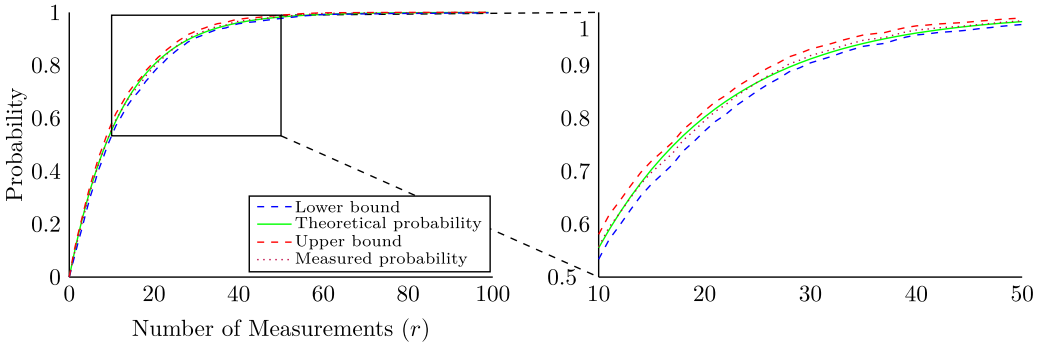

Figure 4. Plot of the Clopper–Pearson confidence intervals for the probabilities of the 3000 AES S-box experiments to solve Problem 1 using r samples and the probabilities of Theorem 4.

5.4. Analysis

To verify the significance of the estimated probabilities of the experiments, we used Clopper–Pearson confidence intervals [30] with a confidence level of $\gamma = 99\%$ each. In particular, we calculated confidence intervals for all the numbers r of sampled masks needed to measure a mask to solve Problem 1 for the AES S-box and the XOR cipher experiments. For the AES S-box, Figure 4 displays the confidence intervals, the relative frequencies of the experiments that solve Problem 1 with r samples, and the probabilities p_r from Theorem 4.

All measured probabilities lay within the respective confidence intervals. For the individual confidence intervals with $r \geq 10$ for the AES S-box and $r \geq 1$ for the XOR cipher, the probability of the experiments differed practically negligibly from the theoretical results.

We further highlight this fact by giving the absolute relative deviations of the boundaries of the confidence intervals from the theoretically obtained success probabilities for the 3000 AES S-box experiments in Figure 5. The absolute relative deviation is given by the term $|p_r - p'_r|/p_r$, where p'_r is one of the boundaries of the confidence interval. The plot shows the deviations using a logarithmic scale, illustrating the strictly monotonic decline with the increase in the count of samples measured until success r. Starting with $r = 10$, the absolute relative deviations became less than 5%, from which we considered them negligible. We observed a narrower behavior in the case of the experiments on the XOR cipher, where the absolute relative deviations were less than 1% for $r = 1$ and $r = 2$, which were the sample counts observed for solving Problem 1.

The deviations between the observed intervals did not directly indicate a potential bias or limitation in the experiments. In particular, the Clopper–Pearson interval is conservative [34], which means that high confidence ($\gamma = 99\%$) could lead to larger intervals considering the statistical variability in the experiments. In addition, the code developed for this study was rigorously tested during its creation and was written in Python. It reflected the noiseless execution of the CEA on an idealized quantum computer whilst having been executed on a classical machine. Therefore, there was no evidence of bias or limitations in this study. The deviations between the intervals depended on the initial seed ς, as it gave the random choices made by the standard pseudorandom number generator by Python, which corresponded to the quantum measurement results obtained by the CEA.

Considering the previous discussion, we may conclude that the associated confidence interval helps confirm Theorem 4 for the AES S-box and the XOR cipher, as it indicates that the theory also applies in general.

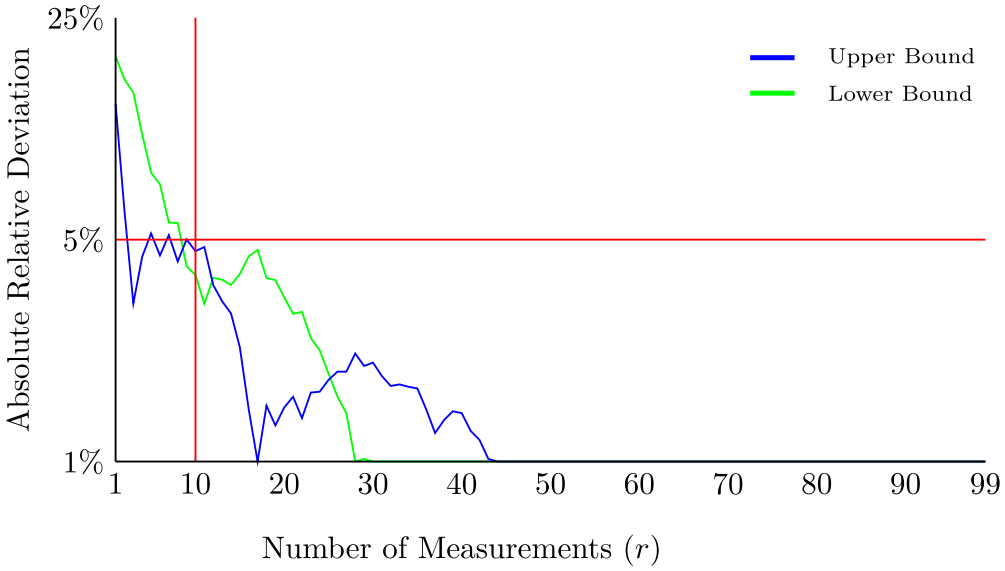

Figure 5. Plot of the absolute relative deviations of the boundaries of the Clopper–Pearson confidence intervals from the predictions obtained using Theorem 4 during 3000 AES S-box experiments.

6. Discussion

This section begins by exploring the applications of the CEA in linear cryptanalysis. Then, it asymptotically compares Algorithm 3, an approach using the CEA, with Algorithm 1, which uses the FWT. Finally, this section discusses the capabilities of the CEA for linear cryptanalysis.

The CEA has two main applications in linear cryptanalysis. When attacking a cipher with linear cryptanalysis, masks with a prescribed bias are necessary. Since the primary purpose of the CEA is to measure a biased mask, it can be applied to solve Problem 1. In addition to attacking a cipher, the CEA helps design ciphers. Specifically, using Algorithm 3 can probabilistically prove that there is no mask with a particular nonzero bias. In addition,

this paper emphasizes that proving the absence of masks with a particular bias using the CEA is an empirical approach that might only be acceptable in some scenarios.

Having explored the two main applications of the CEA in linear cryptanalysis, the next step here would be to compare the complexity of Algorithm 3, an algorithm that solves Problem 1 using the CEA, with Algorithm 1, which solves Problem 1 using the FWT. In contrast to Algorithm 1, as shown in Theorem 7, the time complexity of Algorithm 3 additionally depends on the threshold $\tau \in (0, 1/2]$ and the probability of success $(1 - \delta) \in (0, 1)$. Setting these variables to unrealistic low values would give a misleading comparison and conceal the potential of the CEA in linear cryptanalysis. Therefore, we specify Scenario 1, which defines the setting of our comparison.

Scenario 1. *A function $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$ with circuit depths $\mathcal{O}(T_f) = \text{poly}(n + m)$ and $T_{U_f} = \mathcal{O}(T_f)$, together with probability $\delta = 1/(m + n)^2$ and absolute bias $\tau \in (0, 1/2]$ with $\tau = const$, defines the comparison scenario for Algorithms 1 and 3.*

As specified in Scenario 1, the comparison scenario asymptotically fixes four variables, and the following discusses the reasonability behind these choices.

1. In practice, ciphers must be efficiently computable. Therefore, this paper concludes that assessing functions with $T_f = \text{poly}(m + n)$ using linear cryptanalysis is a reasonable assumption.
2. As mentioned, Kitaev et al. [28] (Thm. 7.3) prove that every function f is realizable as quantum oracle U_f with $T_{U_f} = \mathcal{O}(()T_f)$.
3. As Theorem 7 shows, the probability of failure δ increases the time and query complexity of Algorithm 3 only logarithmically. Hence, this paper conservatively sets $\delta = 1/(m + n)^2$ and argues that smaller values would be unreasonable. To clarify, if $m + n > 10$, then the success probability $(1 - \delta)$ of Algorithm 3 is greater than 0.99.
4. This paper focuses on linear cryptanalysis, specifically Matsui's Algorithms 1 and 2, whose data complexity, illustrated in Table 1, grows with smaller biases. However, the data complexity cannot exceed a specific boundary for practical reasons. To clarify, before attacking a cipher with Algorithms 1 or 2, it is necessary to collect plaintext–ciphertext pairs or at least ciphertexts for the same key, respectively. Since the available computational resources limit the size of manageable data complexities, collecting a vast quantity of data is impractical. Furthermore, recalling the relation between bias and data complexity, a limit for data complexity also limits the lower bound of the absolute bias used by Algorithms 1 or 2. Hence, this paper argues that $\tau = \text{const}$ is reasonable.

After addressing the rationality of variable choices in Scenario 1, this article compares Algorithm 1 with Algorithm 3 based on Scenario 1. For this purpose, this paper applies Scenario 1 to Theorems 2 and 7, whereby Table 3 displays the result. In addition, the following paragraphs detail the comparison of the time complexity, the oracle complexity, and the requirements for space and qubits.

In the scenario specified in Scenario 1, the time complexity of Algorithm 3 has an asymptotic advantage over Algorithm 1, reducing a linear factor to a logarithmic one. However, in contrast to Algorithm 1, Algorithm 3 has the potential to improve its time complexity. In detail, in the context of testing the bias of a mask (line 5), Algorithm 3 uses a direct approach to calculate the Walsh transform, which inflicts an exponential time complexity of $\mathcal{O}((m + n + T_f) \cdot 2^m)$. Applying a more efficient approach to line 5 would improve the time complexity of Algorithm 3. Nonetheless, finding an efficient approach for calculating the Walsh transform is beyond the scope of this paper and is considered future work, which is more detailed in Section 7.

As stated in Section 4, we do not differentiate between classical and quantum oracle complexity. The oracle complexity for Algorithm 1 has an asymptotic advantage of a logarithmic factor over Algorithm 3. Nevertheless, similar to the time complexity, using a

more efficient way of calculation, the Walsh transform in Algorithm 3 could improve its oracle complexity by potentially decreasing the number of evaluations of f.

Compared to the query complexity and neglecting the additional qubits, the space complexity of Algorithm 3 has an exponential advantage over that of Algorithm 1. In particular, Algorithm 1 has a space complexity of $\mathcal{O}(2^m)$. In contrast, Algorithm 3 has a space complexity of $\mathcal{O}(n+m)$, demonstrating an exponential advantage. Remarkably, for Algorithm 3, this advantage comes at the cost of only using $\mathcal{O}(m+n)$ qubits.

Table 3. Worst-case comparison of Algorithm 1 using the FWT and Algorithm 3 utilizing the CEA for solving Problem 1 in the scenario specified by Scenario 1.

Complexity	**Algorithm 1 (FWT)**	**Algorithm 3 (CEA)**
Time	$\mathcal{O}((n \cdot T_f + m) \cdot 2^{m+n})$	$\mathcal{O}(\log(m+n) \cdot T_f \cdot 2^{m+n})$
Oracle queries	$\mathcal{O}(2^{m+n})$	$\mathcal{O}(\log(m+n) \cdot 2^{m+n})$
Space complexity	$\mathcal{O}(2^m)$	$\mathcal{O}(m+n)$
Qubits	-	$\mathcal{O}(m+n)$

Remark: Let $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$ be a vectorial Boolean function, and T_f its time complexity.

Beyond the advantages shown by comparing Algorithm 1 with Algorithm 3, we highlight the potential of the CEA. The quantum computing approach with the CEA offers a probabilistic method for choosing biased masks. According to Theorem 6, the CEA can reduce the mask search space from $\mathcal{O}(2^{m+n})$ to $\mathcal{O}(\log(\delta^{-1}) \cdot 2^n/\tau^2)$, which, depending on the choice of $\tau \in (0, 1/2]$ and $\delta \in (0,1)$, is an exponential advantage. Therefore, the CEA enables a more direct search for biased masks while filtering masks with zero bias. Neglecting the unresolved issue of efficiently checking the biases, the CEA is useful in specific scenarios that profit from a reduced search space. Specifically, scenarios with a vectorial Boolean function $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$ with one nontrivial mask $(\alpha, \beta) \in \mathbb{F}_2^m \times \mathbb{F}_2^n$, a large constant absolute bias, e.g., $|\varepsilon_f(\alpha, \beta)| = 1/2 - c$ where $c > 0$ is negligibly small, and a time complexity of $T_f = \text{poly}(m+n)$. In the stated scenario, according to Theorem 6, the CEA can measure that mask with a probability of $1 - 1/(m+n)^2$, a time complexity of $\mathcal{O}(\log(m+n) \cdot T_f \cdot 2^n)$, and $\mathcal{O}(\log(m+n) \cdot 2^n)$ queries. Remarkably, in this scenario, the number of queries is only exponential in the number of output bits of f, which makes it applicable for functions with $n \leq 32$.

7. Conclusions

Finding masks for $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$ with a prescribed absolute bias for linear cryptanalysis is assumed to be challenging, and this paper showed that the CEA could support this search. In addition to formalizing the success probability of CEA, we provided an asymptotic analysis that proved its capabilities to measure masks with a prescribed absolute bias. The theory was confirmed through an empirical study using a quantum computer simulation on the AES S-box and the XOR cipher with a fixed key. Furthermore, applying a practical scenario (searching for masks with a large constant absolute bias), we demonstrated that the approach using the CEA hads an asymptotic advantage, reducing a linear factor to a logarithmic one in terms of time complexity and an exponential advantage in terms of space compared to a classical approach using the FWT (see Table 3 for details). The exponential time complexity in finding a mask with a prescribed absolute bias makes the CEA suitable for functions with $(m+n \leq 32)$. Nonetheless, neglecting the time complexity to check the bias of a mask, in specific scenarios, the CEA can reduce the search space from $\mathcal{O}(2^{m+n})$ to $\mathcal{O}(\log(m+n) \cdot 2^n)$ masks by executing the CEA only $\mathcal{O}(\log(m+n) \cdot 2^n)$ times.

Although CEA probabilistically guarantees the output of a mask with a prescribed absolute bias, it does not determine it. Efficiently determining the actual bias of a mask is a crucial next step in using the CEA to support linear cryptanalysis, but it is beyond the scope of this paper. We took a step towards solving this problem by rigorously giving the relationship between the bias of a mask and its associated Walsh transform, proving that a

direct computation of the bias must be of order $O((m+n+T_f)\cdot 2^m)$, where T_f is the time complexity of f. From the definition of the Walsh transform itself, it is not apparent how to reduce this exponential complexity; thus, it poses the main challenge to solve this problem.

Future directions might involve deeper studies of the properties of the Walsh transform, the development of efficient algorithms, and possibly quantum algorithms, or hardness proofs by reducing the problem of computing the Walsh transform of a function to a computationally hard problem. Reducing the exponent for obtaining a complexity term of, e.g., $O(2^{cm})$ with $c \in (0,1)$ would broaden the range of applicable functions with each bit. Specifically, a value of $c = 1/4$ would allow the determination of Walsh transforms of ciphers with 128 input bits, although we consider this value unattainable at this time, considering the state of the art. Determining the Walsh transform of masks and thus their biases will be the focus of our future work as we continue to explore the capabilities of the CEA in the context of linear cryptanalysis.

Author Contributions: Conceptualization: C.G.; data curation: C.G. and V.P.; formal analysis: C.G.; funding acquisition: M.M., F.M. and H.E.; investigation: C.G. and V.P.; methodology: C.G.; project administration: C.G., H.E. and F.M.; software: V.P. and C.G.; supervision: C.G., H.E. and M.M.; validation: C.G., V.P., H.E., F.M., H.H. and M.M.; visualization: V.P. and C.G.; writing—original draft: C.G.; writing—review and editing: C.G., V.P., H.E., F.M., H.H. and M.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the German Federal Ministry of Finance.

Data Availability Statement: The data of the empirical study are provided in a GitHub repository [32].

Acknowledgments: This article was written as part of the Qu-Gov project, which was commissioned by the German Federal Ministry of Finance. The authors want to extend their gratitude to Oliver Muth, Yvonne Ripke, and Andreas Wilke for their continuous encouragement and support. Furthermore, the authors wish to thank the reviewers from the journal for their valuable criticism and helpful suggestions, which considerably helped in improving the quality of this article.

Conflicts of Interest: Authors Holger Eble and Frank Morgner were employed by Bundesdruckerei GmbH. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

AES	Advanced Encryption Standard
CEA	Correlation Extraction Algorithm
DES	Data Encryption Standard
FEAL	Fast Data Encipherment Algorithm
FWT	Fast Walsh transform
RAM	Random Access Machine
SPN	Substitution–permutation network
RRAM	Real Random Access Machine

References

1. Gill, S.S.; Kumar, A.; Singh, H.; Singh, M.; Kaur, K.; Usman, M.; Buyya, R. Quantum computing: A taxonomy, systematic review and future directions. *Softw. Pract. Exp.* **2022**, *52*, 66–114. https://doi.org/10.1002/spe.3039.
2. Shor, P. Algorithms for quantum computation: Discrete logarithms and factoring. In Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, USA, 20–22 November 1994; pp. 124–134. https://doi.org/10.1109/SFCS.1994.365700.
3. Gerjuoy, E. Shor's factoring algorithm and modern cryptography. An illustration of the capabilities inherent in quantum computers. *Am. J. Phys.* **2005**, *73*, 521–540. https://doi.org/10.1119/1.1891170.
4. Proos, J.; Zalka, C. Shor's Discrete Logarithm Quantum Algorithm for Elliptic Curves. *Quantum Inf. Comput.* **2003**, *3*, 317–344.
5. Larasati, H.T.; Kim, H. Quantum Cryptanalysis Landscape of Shor's Algorithm for Elliptic Curve Discrete Logarithm Problem. In *Information Security Applications*; Kim, H., Ed.; Springer: Cham, Switzerland, 2021; pp. 91–104. https://doi.org/10.1007/978-3-030-89432-0_8.

6. Grassl, M.; Langenberg, B.; Roetteler, M.; Steinwandt, R. Applying Grover's Algorithm to AES: Quantum Resource Estimates. In *Post-Quantum Cryptography*; Takagi, T., Ed.; Springer: Cham, Switzerland, 2016; pp. 29–43. https://doi.org/10.1007/978-3-319-29360-8_3.
7. Hasija, T.; Ramkumar, K.R.; Kaur, A.; Mittal, S.; Singh, B. A Survey on NIST Selected Third Round Candidates for Post Quantum Cryptography. In Proceedings of the 2022 7th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 22–24 June 2022; pp. 737–743. https://doi.org/10.1109/ICCES54183.2022.9835864.
8. Burek, E.; Wroński, M.; Mańk, K.; Misztal, M. Algebraic Attacks on Block Ciphers Using Quantum Annealing. *IEEE Trans. Emerg. Top. Comput.* **2022**, *10*, 678–689. https://doi.org/10.1109/TETC.2022.3143152.
9. Chen, Y.A.; Gao, X.S. Quantum Algorithm for Boolean Equation Solving and Quantum Algebraic Attack on Cryptosystems. *J. Syst. Sci. Complex.* **2022**, *35*, 373–412. https://doi.org/10.1007/s11424-020-0028-6.
10. Ding, J.; Gheorghiu, V.; Gilyén, A.; Hallgren, S.; Li, J. Limitations of the Macaulay Matrix Approach for Using the HHL Algorithm to Solve Multivariate Polynomial Systems. *Quantum* **2023**, *7*, 1069. https://doi.org/10.22331/q-2023-07-26-1069.
11. Kaplan, M.; Leurent, G.; Leverrier, A.; Naya-Plasencia, M. Quantum Differential and Linear Cryptanalysis. *IACR Trans. Symmetric Cryptol.* **2016**, *2016*, 71–94. https://doi.org/10.13154/tosc.v2016.i1.71-94.
12. Zhou, Q.; Lu, S.; Zhang, Z.; Sun, J. Quantum Differential Cryptanalysis. *Quantum Inf. Process.* **2015**, *14*, 2101–2109. https://doi.org/10.1007/s11128-015-0983-3.
13. David, N.; Naya-Plasencia, M.; Schrottenloher, A. Quantum Impossible Differential Attacks: Applications to AES and SKINNY. *Des. Codes Cryptogr.* **2024**, *92*, 723–751. https://doi.org/10.1007/s10623-023-01280-y.
14. Zou, H.; Zou, J.; Luo, Y. New Results on Quantum Boomerang Attacks. *Quantum Inf. Process.* **2023**, *22*, 171. https://doi.org/10.1007/s11128-023-03921-6.
15. Xie, H.; Yang, L. Quantum Truncated Differential and Boomerang Attack. *Symmetry* **2024**, *16*, 1124. https://doi.org/10.3390/sym16091124.
16. Malviya, A.K.; Tiwari, N. Linear approximation of a vectorial Boolean function using quantum computing. *EPL (Europhys. Lett.)* **2020**, *132*, 40001. https://doi.org/10.1209/0295-5075/132/40001.
17. Hosoyamada, A. Quantum Speed-Up for Multidimensional (Zero Correlation) Linear Distinguishers. In *Advances in Cryptology—ASIACRYPT 2023*; Guo, J., Steinfeld, R., Eds.; Lecture Notes in Computer Science; Springer: Singapore, 2023; pp. 311–345. https://doi.org/10.1007/978-981-99-8727-6_11.
18. Malviya, A.K.; Tiwari, N. Quantum linear cryptanalysis on a toy cipher. *Pramana* **2023**, *97*, 63. https://doi.org/10.1007/s12043-023-02529-w.
19. Matsui, M.; Yamagishi, A. A New Method for Known Plaintext Attack of FEAL Cipher. In *Advances in Cryptology—EUROCRYPT' 92*; Rueppel, R.A., Ed.; Springer: Berlin/Heidelberg, Germany, 1993; pp. 81–91. https://doi.org/10.1007/3-540-47555-9_7.
20. Matsui, M. Linear Cryptanalysis Method for DES Cipher. In *Advances in Cryptology — EUROCRYPT '93*; Helleseth, T., Ed.; Springer: Berlin/Heidelberg, Germany, 1994; pp. 386–397. https://doi.org/10.1007/3-540-48285-7_33.
21. Kaliski, B.S.; Robshaw, M.J.B. Linear Cryptanalysis Using Multiple Approximations. In *Advances in Cryptology — CRYPTO '94*; Desmedt, Y.G., Ed.; Springer: Berlin/Heidelberg, Germany, 1994; pp. 26–39. https://doi.org/10.1007/3-540-48658-5_4.
22. Hermelin, M.; Cho, J.Y.; Nyberg, K. Multidimensional Extension of Matsui's Algorithm 2. In *Fast Software Encryption*; Dunkelman, O., Ed.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 209–227. https://doi.org/10.1007/978-3-642-03317-9_13.
23. Bogdanov, A.; Rijmen, V. Linear hulls with correlation zero and linear cryptanalysis of block ciphers. *Des. Codes Cryptogr.* **2014**, *70*, 369–383. https://doi.org/10.1007/s10623-012-9697-z.
24. Brown. A Recursive Algorithm for Sequency-Ordered Fast Walsh Transforms. *IEEE Trans. Comput.* **1977**, *C-26*, 819–822. https://doi.org/10.1109/TC.1977.1674921.
25. Cui, J.; Guo, J. Quantum cryptographic property testing of multi-output Boolean functions. *Quantum Inf. Process.* **2019**, *18*, 182. https://doi.org/10.1007/s11128-019-2299-1.
26. Li, H. Quantum Algorithms for the Resiliency of Vectorial Boolean Functions. *Int. J. Theor. Phys.* **2021**, *60*, 1565–1573. https://doi.org/10.1007/s10773-021-04779-z.
27. Carlet, C. Vectorial Boolean Functions for Cryptography. In *Boolean Models and Methods in Mathematics, Computer Science, and Engineering*; Hammer, P.L., Crama, Y., Eds.; Encyclopedia of Mathematics and Its Applications; Cambridge University Press: Cambridge, UK, 2010; pp. 398–470. https://doi.org/10.1017/CBO9780511780448.012.
28. Kitaev, A.Y.; Shen, A.H.; Vyalyi, M.N. *Classical and Quantum Computation*; AMS, American Mathematical Society: Providence, RI, USA, 2002; Volume 47, Chapter 2, p. 64.
29. Dworkin, M.J.; Turan, M.S.; Mouha, N. (Eds.) *Advanced Encryption Standard (AES)*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2023.
30. Clopper, C.J.; Pearson, E.S. The Use of Confidence or Fiducial Limits Illustrated in the Case of the Binomial. *Biometrika* **1934**, *26*, 404–413. https://doi.org/10.2307/2331986.
31. Javadi-Abhari, A.; Treinish, M.; Krsulich, K.; Wood, C.J.; Lishman, J.; Gacon, J.; Martiel, S.; Nation, P.D.; Bishop, L.S.; Cross, A.W.; et al. Quantum Computing with Qiskit. *arXiv* 2024, arXiv:2405.08810; Version Number: 3. https://doi.org/10.48550/ARXIV.2405.08810.
32. Graebnitz, C.; Margraf, M.; Pickel, V. Emperical-Study-Correlation-Extraction-Algorithm. Available online: https://github.com/JCTHRG/Emperical-Study-Correlation-Extraction-Algorithm (accessed on 13 August 2024).

33. Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* **2020**, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2.
34. Chen, X.; Zhou, K.; Aravena, J.L. On the Binomial Confidence Interval and Probabilistic Robust Control. *Automatica* **2004**, *40*, 1787–1789. https://doi.org/10.1016/j.automatica.2004.04.016.