EPJ.org

EPJ Quantum Technology
a SpringerOpen Journal

RESEARCH Open Access

Quantum differential cryptanalysis based on Bernstein-Vazirani algorithm

Rong-Xue Xu[1], Hong-Wei Sun[1*], Ke-Jia Zhang[2], Gang Du[1] and Dan-Dan Li[3]

[*]Correspondence:
sunhw@hlju.edu.cn
[1]School of Computer and Big Data (School of Cybersecurity), Heilongjiang University, Harbin 150080, China
Full list of author information is available at the end of the article

Abstract

Recent research has demonstrated the potential of quantum algorithms to exploit vulnerabilities in various popular constructions, such as certain block ciphers like Feistel, Even-Mansour, and multiple MACs, within the superposition query model. In this study, we delve into the security of block ciphers against quantum threats, particularly investigating their susceptibility to cryptanalysis techniques, notably exploring quantum adaptations of differential cryptanalysis. Initially, we introduce a BV-based quantum algorithm for identifying linear structures with a complexity of $O(n)$, where n denotes the number of bits in the function. Subsequently, we illustrate the application of this algorithm in devising quantum differential cryptanalysis techniques, including quantum differential cryptanalysis, quantum small probability differential cryptanalysis, and quantum impossible differential cryptanalysis, demonstrating polynomial acceleration compared to prior approaches. By treating the encryption function as a unified entity, our algorithm circumvents the traditional challenge of extending differential paths in differential cryptanalysis.

Keywords: Block cipher; Differential cryptanalysis; Quantum cryptanalysis; BV algorithm

1 Introduction

Modern cryptography relies on the concept of computational security, and the level of security provided by a cryptographic system can be expressed as the amount of computing resources required to break it. However, based on the assumption of computational complexity, it is difficult for people to break existing cryptographic algorithms within a limited time under the current computing power. This is the cornerstone of the security of the classical cryptosystem. However, due to the existence of quantum algorithms [1–4], the security of this cryptographic system has been dramatically affected. For example, Shor's algorithm [1] will greatly impact the currently widely used RSA cryptographic systems [5], and Grover's algorithm [2] provides quadratic acceleration for exhaustive key search (the key length is reduced by half). Therefore, it is of great significance to the development of cryptography to evaluate the specific threats of quantum computing to various cryptographic systems and provide a reference for the design and analysis of cryptographic algorithms that can resist quantum attacks.

Quantum attacks against symmetric crypto primitives The influence of large-scale universal quantum computers is apparent in many public-key schemes, whereas the impact on symmetric cryptography seems less significant. Since symmetric cryptographic algorithms do not depend on trapdoor functions, they appear capable of evading the exponential speedup of quantum computers compared to classical algorithms for an extended period. Initially, it was widely believed that only Grover's algorithm [2], offering a quadratic acceleration of the exhaustive search problem, could leverage quantum resources to target symmetric cryptosystems. This change occurred following the introduction of Simon-based attacks proposed by Kuwakado and Morii. [6, 7]. They demonstrated that Even-Mansour and 3-round Feistel constructions could be broken in polynomial time. Subsequently, various generic constructions were also found to be vulnerable to different quantum algorithms [8–16]. These included attacks based on the Simon algorithm [17], the Grover-meets-Simon algorithm [18], and the Bernstein-Vazirani (BV) algorithm [19], among others.

Quantum differential cryptanalysis Differential cryptanalysis [20] is pivotal in penetrating modern cryptosystems, particularly block ciphers, manifesting in various forms like truncated differential attacks and impossible differentials. Concurrently, quantum computing, grounded in quantum mechanics, has emerged, showcasing superior acceleration over classical computing in certain domains. Consequently, the application potentials of quantum algorithms in differential cryptanalysis become conceivable. In the preliminary phase of differential cryptanalysis, Yang et al. [21, 22] introduced methodologies based on the BV algorithm for discovering diverse differences: quantum differential cryptanalysis, quantum small probability differential cryptanalysis, quantum impossible differential cryptanalysis, and quantum truncated differential cryptanalysis. These approaches can identify desired differences within time $O(\mathrm{poly}(\omega))$ (ω denotes the length of the round key). For the subsequent stage of differential cryptanalysis, i.e., deriving the key from the known difference, Zhou et al. [23] proposed a corresponding quantum version based on quantum search and quantum counting algorithms, achieving a quadratic speedup. In 2017, Kaplan et al. [24] showed that it is usually possible to use quantum computations to obtain a quadratic speedup for these attack techniques, but the situation must be nuanced: they cannot get a quadratic speedup for all variants of the attacks, such as truncated differential cryptanalysis.

Our contributions In this paper, we reassess the BV algorithm and investigate methods to streamline the complexity of BV-based quantum differential attacks on block ciphers. Our focus is on addressing two unresolved inquiries posed by Xie et al. [22].

1. Can an alternative method be employed to directly identify the linear structure of the vector function, bypassing the step of searching linear structures for each component function and then intersecting them, as done by Xie et al.?

   We introduce a novel BV-based quantum algorithm for identifying linear structures of a vector function. This algorithm can efficiently pinpoint approximate linear structures of the vector function with just $O(n)$ quantum queries. Compared to the previous quantum linear structure finding algorithm [22], our approach achieves a quadratic acceleration.

2. How can we further streamline the complexity of quantum differential attacks while maintaining the probability of success?

   We improve three applications of the quantum linear structure finding algorithm in the realm of differential cryptanalysis [22], encompassing quantum differential cryptanalysis, quantum small probability differential cryptanalysis, and quantum impossible differential cryptanalysis. Our BV-based attacks yield a polynomial acceleration compared to some relevant findings, reducing the complexity from $O(n^3q^2(n)\log(n))/O(n^4l^2(n)q^2(n))/O(n^2)$ to $O(q^2(n))/O(nl^2(n)q^2(n))/O(n)$, respectively.

Organization The paper follows this structure: Sect. 2 introduces essential notations, definitions, and key technical lemmas. In Sect. 3, we present a novel quantum algorithm for identifying approximate linear structures of a vector function. Section 4 outlines three methodologies for conducting quantum differential cryptanalysis. Finally, conclusions are drawn in Sect. 5.

2 Preliminaries

We define F_2 as the prime field with elements 0 and 1, denoted $\{0,1\}$. The n-dimensional vector space over F_2 is represented as F_2^n, equivalent to $\{0,1\}^n$. The collection of all functions mapping F_2^m to F_2^n is denoted by $C_{m,n}$. For $n=1$, this set is denoted by B_m. The symbol "$\oplus$" signifies XOR (addition in F_2^n), while "$\cdot$" denotes the scalar product of bit-strings viewed as n-bit vectors.

2.1 Linear structure

Below, we define the concept of linear structures for a mapping F from $\{0,1\}^m$ to $\{0,1\}^n$.

Definition 1 ([25]) Given a function $F \in C_{m,n}$, the linear structure is a vector $a \in \{0,1\}^m$ satisfying the equation:

$$F(x \oplus a) \oplus F(x) = i, \text{ for all } x \in \{0,1\}^m \tag{1}$$

where $i \in \{0,1\}^n$ is a constant vector. Let U_F represent the set of all linear structures of function F, and U_F^i be defined as:

$$U_F^i = \{a \in F_2^m | F(x \oplus a) \oplus F(x) = i, \text{ for all } x \in \{0,1\}^m\} \tag{2}$$

And, $U_F = \bigcup_i U_F^i$. Given any $a \in \{0,1\}^m$ and $i \in \{0,1\}^n$, define:

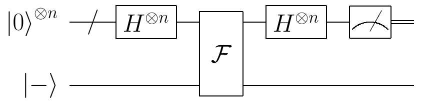
$$V_{F,a}^i = \frac{|\{x \in F_2^m | F(x \oplus a) \oplus F(x) = i\}|}{2^m} \tag{3}$$

This parameter quantifies the deviation of a from a linear structure of the function F. We then introduce the following definition.

Definition 2 ([22, 26, 27]) For a given function $F \in C_{m,n}$, the ϵ-approximate linear structure is a vector $a \in \{0,1\}^m$ satisfying:

$$\frac{|\{x \in F_2^m | F(x \oplus a) \oplus F(x) = i\}|}{2^m} > 1 - \epsilon \tag{4}$$

Figure 1 BV algorithm



where ϵ is a negligible parameter.

Definition 3 ([22, 26, 27]) The Walsh spectrum of a Boolean function $f : F_2^m \to F_2$ is defined as the function $S_f : F_2^m \to Z$ by:

$$S_f(\omega) = \frac{1}{2^m}\Sigma_{x\in F_2^m}(-1)^{f(x)+\omega\cdot x} \tag{5}$$

where the dot product $\omega \cdot x = \omega_1 x_1 \oplus \omega_2 x_2 \oplus \cdots \oplus \omega_m x_m$ is defined for any $\omega = (\omega_1, \dots, \omega_m)$ and $x = (x_1, \dots, x_m)$.

Let's denote the support of S_f as $\mathrm{supp}(S_f)$, defined as $\mathrm{supp}(S_f) = \{\omega \in F_2^m | S_f(\omega) \neq 0\}$. Then, Lemma 1 [28] elucidates how the Walsh spectrum enables the determination of linear structures within a Boolean function.

Lemma 1 ([28]) *Given a function $f \in B_m$. For $\forall i \in \{0, 1\}$, we have that*

$$U_f^i = \{a \in F_2^m | a \cdot \omega = i, \forall \omega \in supp(S_f)\} \tag{6}$$

Lemma 1 demonstrates that with a sufficiently large subset W of $\mathrm{supp}(S_f)$, linear structures of f can be obtained by solving the linear equations $\{a \cdot \omega = i | \omega \in W\}$.

2.2 Bernstein-Vazirani algorithm

In 1993, Bernstein and Vazirani [19] proposed a quantum algorithm for determining the slope of an affine function (BV problem). Utilizing quantum superposition of queries, represented as $\Sigma_{x,y}\lambda_{x,y}|x\rangle|y\rangle \mapsto \Sigma_{x,y}\lambda_{x,y}|x\rangle|y \oplus f(x)\rangle$, the BV algorithm exhibits linear speedup in query complexity compared to classical methods. Concretely, the BV problem is defined as follows:

Bernstein-Vazirani problem [19]: Given a oracle O_f[1] : $\{0, 1\}^n \to \{0, 1\}$ defined by $f(x) = a \cdot x \bmod 2$ for some hidden $a \in \{0, 1\}^n$, the objective is to determine a.

The BV algorithm (see Fig. 1) addresses the abovementioned problem using the following four quantum steps.

1. Prepare quantum state $|\psi_0\rangle = |0\rangle^{\otimes n}|1\rangle$, apply $H^{\otimes(n+1)}$

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle|-\rangle \tag{7}$$

Here, $H^{\otimes n}|0\rangle^{\otimes n} = [\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)]^{\otimes n} = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{N-1}|x\rangle$.

[1] While we can input x to the oracle and request the computation of $f(x)$, the internal computation process remains inaccessible.

2. Apply oracle O_f

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} |x\rangle|-\rangle \tag{8}$$

   For function $f(x) \in \{0,1\}$, the impact of oracle access operation $U_f : |x\rangle|y\rangle \to |x\rangle|y + f(x)\rangle$ on $|x\rangle|-\rangle$: $U_f|x\rangle(|0\rangle - |1\rangle) = (-1)^{f(x)}|x\rangle(|0\rangle - |1\rangle)$. Easily verifiable, the conditions hold true when $f(x)$ equals 0 or 1.

3. Apply Hadamard operation $H^{\otimes n}$ (erase the last qubit)

$$\begin{aligned} |\psi_3\rangle &= H^{\otimes n} \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} |x\rangle \\ &= \frac{1}{\sqrt{2^n}} [\frac{1}{\sqrt{2^n}} \sum_{x,y\in\{0,1\}^n} (-1)^{y\cdot x} |y\rangle\langle x|] \sum_{x\in\{0,1\}^n} (-1)^{f(x)} |x\rangle \\ &= \sum_{y\in\{0,1\}^n} [\frac{1}{2^n} \sum_{x\in\{0,1\}^n} (-1)^{f(x)+y\cdot x}]|y\rangle \\ &= \sum_{y\in\{0,1\}^n} S_f(y)|y\rangle \end{aligned} \tag{9}$$

4. Measure state, output a

$$p(y) = [\frac{1}{2^n} \sum_{x\in\{0,1\}^n} (-1)^{f(x)+y\cdot x}]^2 = [\frac{1}{2^n} \sum_{x\in\{0,1\}^n} (-1)^{x\cdot(a\oplus y)}]^2 = \delta_{a,y} \tag{10}$$

   Clearly, $\delta_{a,y} = \frac{1}{\sqrt{2^n}} \Sigma_{x\in F_2^n} (-1)^{f(x)+y\cdot x} = S_f(y)$, where $\delta_{a,y} = 1$ if $a = y$ and 0 otherwise. Then, the probability of obtaining a after measurement is 1.

The classical time complexity for solving this problem optimally is $O(n)$. However, the BV algorithm [19] achieves significant acceleration with a complexity of only $O(1)$. We refer to [29] for a qiskit implementation with a small example. Thus, it serves as an efficient tool for identifying linear structures within a vector function.

3 Quantum linear structure finding algorithm

In this section, we introduce a novel quantum linear structure search algorithm, leveraging the BV algorithm as a subroutine. In our subsequent applications, the accessible functions no longer adhere to the BV problem, meaning they are not of the specific form of $f(x) = a \cdot x$. In this scenario, the problem condition, represented by the linear function $f(x) = a \cdot x$, is relaxed to the general form of a Boolean function $f(x) : \{0,1\}^n \to \{0,1\}$. Based on the Eq. (9) above, applying the BV algorithm to the mentioned function at this stage results in a probability $S_f^2(y)$ of obtaining vector y, indicating it consistently returns vector $y \in supp(S_f)$.

New observations We now turn to the vector function. Given the BV algorithm's exclusive operation on Boolean functions, we construct the following function for the vector function $F = (f_1, f_2, \ldots, f_n)$:

$$\phi := f_1 \oplus f_2 \oplus \cdots \oplus f_n \tag{11}$$

```
Algorithm 1 FindStruct
Input: Function $F = (f_1, f_2, \ldots, f_n) \in C_{n,n}$; $\phi := f_1 \oplus f_2 \oplus \cdots \oplus f_n$; Initialize $W$ as empty
Output: Return $(a, i)$
Repeat $cn$ times
    Run the BV algorithm on $\phi$                                   $\triangleright y$
    Set $W = W \cup \{y\}$
EndRepeat
if $\{y \cdot a = i_\phi | y \in W, i_\phi \in \{0,1\}\}$ has no solution then
    return “failure"
else
    return the nonzero solution $a$ and calculate $i = F(x \oplus a) \oplus F(x)$    $\triangleright (a, i)$
end if
```

At this stage, $\phi \in B_n$ represents a Boolean function. It's apparent that when function F exhibits a linear structure, the newly formulated function ϕ also inherits a linear structure and is equivalent. Then, we have the following proposition.

Proposition 1 *If there exists a function ϕ such that $\phi := f_1 \oplus f_2 \oplus \cdots \oplus f_n$, then a being the linear structure of vector function $F = (f_1, f_2, \ldots, f_n)$ is both necessary and sufficient for a to be the linear structure of ϕ.*

Proof Function F has a linear structure a.

$$
\begin{aligned}
&\Leftrightarrow F(x \oplus a) \oplus F(x) = i, \forall x \in \{0,1\}^n \\
&\Leftrightarrow f_j(x \oplus a) \oplus f_j(x) = i_j, j = 1, 2, \ldots, n \\
&\Leftrightarrow f_1(x \oplus a) \oplus f_1(x) \oplus \cdots \oplus f_n(x \oplus a) \oplus f_n(x) = i_\phi, i_\phi = i_1 \oplus \cdots \oplus i_n \\
&\Leftrightarrow \phi(x \oplus a) \oplus \phi(x) = i_\phi, i_\phi \in \{0,1\}
\end{aligned}
\tag{12}
$$

Then, function ϕ has a linear structure a. □

Proposition 1 suggests that solving the linear structure problem of the vector function F can be reduced to solving the linear structure problem of Boolean functions ϕ. When the BV algorithm is applied to the given function ϕ, it consistently yields a vector $y \in supp(S_\phi)$. Lemma 1 establishes the algebraic connection between this output and the linear structure a ($\{y \cdot a = i_\phi | i_\phi \in \{0,1\}\}$), facilitating the reconstruction of the vector function's linear structure. The above process can be summarized as Algorithm 1.

Using the proposed algorithm, we can polynomially solve the approximate linear structure problem of vector functions. In our subsequent cryptographic analysis applications, collisions beyond the function's linear structure will impact the algorithm's success rate. Therefore, before drawing conclusions, we must first define the parameter (the maximum collision ratio)

$$
\varepsilon(F, a) = \frac{1}{2^n} \max_{1 \le j \le n} \max_{t \in F_2^n \setminus U_f} \max_{i_j \in F_2} |\{x \in F_2^n | F_j(x \oplus t) \oplus F_j(x) = i_j\}| \tag{13}
$$

This parameter $\varepsilon(F,a) < 1$ quantifies the proximity between collisions and linear structures in the vector function. As it increases, collisions in the function approach the linear structure, exert a greater impact on the algorithm; conversely, the impact diminishes. Specifically, we provide the following two theorems to illustrate this (see Appendix for the proof).

Theorem 1 *If $\varepsilon(F,a) \le p_0 < 1$, Algorithm 1, after cn quantum queries, can still successfully recover the linear structure (a,i) with a probability of at least $1 - (2p_0^c)^n$.*

Theorem 1 suggests that with a small parameter $\varepsilon(F,a)$, increasing the number of queries can mitigate the effect of collisions on the algorithm's success probability. In essence, when the parameter c is adequately large (usually, selecting $c > \log_{p_0} 2^{-1}$ is adequate), the algorithm's success probability exponentially converges to 1 as n grows. At this point, the proposed algorithm can find the linear structure (including periods) of the vector function in $O(n)$ time.

Theorem 2 *When Algorithm 1 is executed on the vector function $F = (f_1, f_2, \ldots, f_n)$ for cn times, if Algorithm 1 returns vector (a,i), then for any ϵ $(0 < \epsilon < 1)$, we have*

$$Pr[\frac{|\{x \in F_2^n | F(x \oplus a) \oplus F(x) = i\}|}{2^n} > 1 - \epsilon] > 1 - e^{-2cn\epsilon^2} \tag{14}$$

Theorem 2 suggests that when the collision ratio is high, Algorithm 1 will output the approximate linear structure of the function, denoted as ϵ-approximate linear structure.

Truncate outputs of quantum oracles We begin by constructing quantum oracle capable of implementing each component function $f_j : \{0,1\}^n \to \{0,1\}$ $(1 \le j \le n)$ within the vector function $F = (f_1, f_2, \ldots, f_n)$. This process allows us to build a quantum oracle for the function ϕ. This necessitates employing quantum truncation techniques [30]. In the following, we will outline specific construction methods.

Given the function F, we can efficiently access the quantum encryption oracle $O_F : |x\rangle|y\rangle \mapsto |x\rangle|y \oplus O_F(x)\rangle$. Now, we need to use this oracle to simulate the quantum oracle of the component functions f_j. To do this, we define the unitary operation $O'_{f_j} := (I_n \otimes H^{\otimes(j-1)} \otimes I \otimes H^{\otimes(n-j)}) \cdot O \cdot (I_n \otimes H^{\otimes(j-1)} \otimes I \otimes H^{\otimes(n-j)})$. Specifically, we have:

$$\begin{aligned} O'_{f_j}|x\rangle|0\rangle|0\rangle|0\rangle &= (I_n \otimes H^{\otimes(j-1)} \otimes I \otimes H^{\otimes(n-j)}) \cdot O_F \\ &\quad \cdot (I_n \otimes H^{\otimes(j-1)} \otimes I \otimes H^{\otimes(n-j)})|x\rangle|0\rangle|0\rangle|0\rangle \\ &= (I_n \otimes H^{\otimes(j-1)} \otimes I \otimes H^{\otimes(n-j)}) \cdot O_F|x\rangle|+\rangle|0\rangle|+\rangle \\ &= (I_n \otimes H^{\otimes(j-1)} \otimes I \otimes H^{\otimes(n-j)})|x\rangle[\frac{1}{\sqrt{2}}\Sigma_u|u \oplus O_{f_1}(x)\rangle] \\ &\quad \cdots |O_{f_j}(x)\rangle \cdots [\frac{1}{\sqrt{2}}\sum_v |v \oplus O_{f_n}(x)\rangle] \\ &= (I_n \otimes H^{\otimes(j-1)} \otimes I \otimes H^{\otimes(n-j)})|x\rangle|+\rangle \cdots |O_{f_j}(x)\rangle \cdots |+\rangle \\ &= |x\rangle|0\rangle|O_{f_j}(x)\rangle|0\rangle \end{aligned} \tag{15}$$

Therefore, we can simulate $O_\phi : |x\rangle|0\rangle \to |x\rangle|O_{f_1} \oplus O_{f_2} \oplus \cdots \oplus O_{f_n}\rangle$ with the complete encryption oracle O_F using ancilla qubits:

$$\begin{aligned} |x\rangle|0\rangle &\overset{O_{f_j}}{\Rightarrow} |x\rangle|O_{f_1}\rangle|O_{f_2}\rangle\cdots|O_{f_n}\rangle|0\rangle \\ &\Rightarrow |x\rangle|O_{f_1}\rangle|O_{f_2}\rangle\cdots|O_{f_n}\rangle|O_{f_1} \oplus O_{f_2} \oplus \cdots \oplus O_{f_n}\rangle \\ &\overset{O_{f_j}}{\Rightarrow} |x\rangle|0\rangle|O_{f_1} \oplus O_{f_2} \oplus \cdots \oplus O_{f_n}\rangle \end{aligned} \tag{16}$$

Related work The first approach to trying to find linear structures with the BV algorithm was proposed by Li and Yang [26]. Later, Xie and Yang [22] extended the aforementioned algorithm to the case of vector functions, while also providing some cryptographic analysis applications, including differential analysis, related-key analysis, and others. Specifically, considering that the BV algorithm only operates on Boolean functions, they needed to determine the linear structures of each component function to derive the linear structure of the vector function (with $O(n^2)$). In contrast, we constructed a Boolean function based on the vector function and analyzed the algebraic relationships between function linear structures, thereby providing a solution to the problem. This significantly reduces the complexity of solving the problem, achieving quadratic acceleration.

4 Quantum differential cryptanalysis

In this section, we show how our proposed algorithms significantly decrease the query complexity of classical differential analysis. Compared to the method by Xie and Yang [22], our attacks, including quantum differential attacks, small-probability differential attacks, and impossible differential attacks, achieve polynomial acceleration. Notably, the query complexity of our differential attacks decreases from $O(n^3q^2(n)\log(n))/O(n^4l^2(n)q^2(n))/O(n^2)$ to $O(q^2(n))/O(nl^2(n)q^2(n))/O(n)$.

4.1 Applications to the quantum differential cryptanalysis

Differential Cryptanalysis [20] is a cryptographic analysis technique used to compromise encryption algorithms' security. It analyzes the output differences resulting from small input data variations to infer the key or other encryption parameters. Typically, this technique observes input differences and deduces key information from them. It can be applied to attack numerous symmetric-key encryption algorithms, including DES and AES.

Differential cryptanalysis is a chosen-plaintext attack that relies exclusively on the resulting ciphertexts. Here, we analyze the encryption function $E : \{0,1\}^n \to \{0,1\}^n$ of an iterated design with r rounds, and we use $E^{(r)}$ to denote a reduced version with r rounds (i.e., $E := E^{(r)}$). Let $E_k^{(r-1)}$ be the encryption function of the first $r-1$ rounds, and k denotes the key involved in these rounds. Differential cryptanalysis consists of the following two steps:

1. Analyze the target encryption algorithm $E^{(r-1)}$ to identify differential characteristics;
2. Utilizing these identified characteristics, attackers endeavor to deduce the key or other critical information of the target encryption algorithm.

In this study, we explore quantum versions of differential cryptanalysis, and introduce the described function:

$$F : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^n$$

Algorithm 2 Improved FindStruct

```
Input: Encryption function $F = (f_1, f_2, \ldots, f_n) \in C_{n+d,n}$; $\phi(x) := f_1 \oplus f_2 \oplus \cdots \oplus f_n$; Initialize
    the set $W$ as empty
Output: Return $(a, i)$
Repeat $cn$ times
    Run the BV algorithm on $\phi$                         $\triangleright y = y_1 y_2 \cdots y_n y_{n+1} \cdots y_{n+d}$
    Set $W = W \cup \{y_1 y_2 \cdots y_n\}$
EndRepeat
if $\{y \cdot a = i_\phi | y \in W, i_\phi \in \{0,1\}\}$ has no solution then
    return “failure"
else
    return the nonzero solution $a$ and calculate $i = F(x \oplus a) \oplus F(x)$      $\triangleright (a, i)$
end if
```

$$(m, k) \mapsto E_k^{(r-1)}(m) \tag{17}$$

Based on the above Eq. (17), we can regard the target encryption algorithm as a function F. Specifically, since we cannot access the key information of the target encryption algorithm, we need to input both the key and the plaintext m as the function's parameters. In other words, under the key $k \in \{0,1\}^d$, the target encryption algorithm maps the plaintext m to the ciphertext $c = E_k^{(r-1)}(m)$, denoted as F. In this scenario, if we apply Algorithm 1 to function F and manage to find an approximate linear structure of function F, then this linear structure corresponds to a high probability differential characteristic of the target encryption algorithm.

The identified linear structure in the aforementioned process corresponds to a differential characteristic under the related-key condition. In general, we need to identify a differential characteristic under the same key (i.e., the correct key of the target encryption algorithm). In this case, we only need to set the last d bits of the function input (corresponding to the key part) to 0. Concretely, we give Algorithm 2.

Using Algorithm 2, we can identify the linear structure of function F with a query complexity of $O(n)$. The recovered linear structure (a, i) then undergoes validation using multiple sets of inputs x and $x \oplus a$. If validation fails, Algorithm 2 is rerun until the correct linear structure is found. This process aligns with a differential attack on the target encryption algorithm $E^{(r-1)}$. Specifically, Algorithm 2 can efficiently find the differential characteristic of the target encryption algorithm (for most keys) in polynomial time, as summarized in the following theorem.

Theorem 3 *When Algorithm 2 runs on the encryption function $F = (f_1, f_2, \ldots, f_n)$ for cn iterations, yielding vector (a, i), and for key $k \in K'$ where $K' \subseteq K$ and $|K'|/|K| \geq 1 - \frac{1}{q(n)}$, we have*

$$Pr[\frac{|x \in F_2^n | E^{(r-1)}(x \oplus a) \oplus E^{(r-1)}(x) = i|}{2^n} > 1 - \epsilon] > 1 - e^{\frac{-2cn\epsilon^2}{q^2(n)}} \tag{18}$$

where $q(n)$ is any polynomial in n, representing the block length.

Proof In Algorithm 2, we set the differentials of the key part to 0 to identify the differential characteristics of the target encryption algorithm under the same key. That is, we implicitly assume that $(a\|0)\cdot(y_1y_2\cdots y_{n+d})=i$, i.e., $a\cdot(y_1y_2\cdots y_n)=i$. When we regard the limited $(a\|0)$ as the output, Algorithm 2 degenerates into Algorithm 1. Then, from the above Theorem 2, it follows that

$$Pr[\frac{|x\in F_2^{n+d}|F(x\oplus a\|0)\oplus F(x)=i|}{2^{n+d}}>1-\epsilon_0] \tag{19}$$

holds with the probability greater than $1-e^{-2cn\epsilon_0^2}$. Let

$$V(k)=\frac{|\{x\in F_2^n|E_k^{(r-1)}(x\oplus a)\oplus E_k^{(r-1)}(x)=i\}|}{2^n} \tag{20}$$

Equation (19) indicates that if we consider $V(k)$ as a variable, then for any $k\in\{0,1\}^m$, we have $E_k(V(k))>1-\epsilon_0$. Consequently, for any $q(n)$, we have

$$Pr_k[V(k)>1-q(n)\epsilon_0]>1-\frac{1}{q(n)} \tag{21}$$

The above Eq. (21) indicates that for the majority of keys (with $(1-\frac{1}{q(n)})$ of keys, denoted as K'), we can find a high-probability differential characteristic of the target encryption algorithm. In other words, for any $k\in K'$, we have

$$Pr[V(k)>1-q(n)\epsilon_0]>1-e^{-2cn\epsilon_0^2} \tag{22}$$

Moreover for $\epsilon=q(n)\epsilon_0$, we have

$$Pr[\frac{|x\in F_2^n|E^{(r-1)}(x\oplus a)\oplus E^{(r-1)}(x)=i|}{2^n}>1-\epsilon]>1-e^{\frac{-2cn\epsilon^2}{q^2(n)}} \tag{23}$$

This concludes the proof. □

After $cn=\frac{1}{2}c_1^2q^2(n)ln(c_2)$ steps of Algorithm 2, we can obtain the differential of $E_k^{(r-1)}$. Then, from the above Theorem 3, it follows that there exist constants c_1, c_2 that satisfy

$$Pr[\frac{|x\in F_2^n|E^{(r-1)}(x\oplus a)\oplus E^{(r-1)}(x)=i_1i_2\cdots i_n|}{2^n}>1-\frac{1}{c_1}]>1-\frac{1}{c_2} \tag{24}$$

for $k\in K'$. We show that with this strategy, the adversary can obtain the differential of $E_k^{(r-1)}$ with complexity $O(q^2(n))$. Afterward, it is likely that classical attacks that recover the subkey in the last round. Our algorithm is applied in the first phase. Considering the known best quantum differential algorithm [22], our algorithm is a polynomial speedup with complexity dropping from $O(n^3q^2(n)\ln(n))$ to $O(q^2(n))$.

In traditional differential analysis, as the number of rounds increases, the number of active S-boxes also increases, significantly reducing the likelihood of high-probability differential paths. The resulting complexity in the encryption function further complicates the identification of these high-probability paths, limiting the feasible number of attack rounds. With quantum algorithms, however, the target encryption can be treated as a black box, enabling the use of the BV algorithm to partially mitigate the complexity introduced by additional rounds.

Notably, our algorithm currently applies only to partial keys. Determining whether high-probability differential characteristics applicable to all keys can be identified remains a key challenge [22]. Under related-key conditions, the proposed attack may identify differential features valid for all keys, as the final bit of the linear structure generated by Algorithm 2 aligns with the key's algebraic structure. However, caution is required in practical analysis, as specific encryption algorithms have unique key scheduling. For an attacker to succeed, quantum queries on two related-key black boxes would be necessary, and the key schedule may not yield round keys matching this requirement.

Our approach provides a general framework for attacking symmetric cryptosystems, with its effectiveness largely determined by the differential distribution of the target algorithm. The core of the quantum differential algorithm lies in exploiting the algebraic structure of the target encryption algorithm. Specifically, the attacker constructs a linear structure function and employs Algorithm 1 to recover its linear characteristics for conducting the attack. For example, for the Even-Mansour construction $Enc(m) = P(m \oplus k_1) \oplus k_2$, we can consider the function $f(x) = P(x \oplus k_1) \oplus k_2 \oplus P(x)$, this correspond to a high order differential. The complexity of Algorithm 2 primarily depends on the number of Oracle calls in step 2, specifically the need for set W to include $n-1$ independent vectors. For $n = 12$, finding such $n-1$ independent vectors requires an expected 12.613 calls to the BV algorithm subroutine [31]. Then, the complexity of the proposed quantum differential algorithm is $O(n)$. This study offers insights into the application of quantum algorithms to differential analysis and supports the exploration of block cipher design principles in quantum computing contexts.

4.2 Applications to the quantum small probability differential cryptanalysis

Differential cryptanalysis using small-probability differentials was introduced in Ref. [32, 33], examining the propagation of input differences within the encryption function and their impact on output differences. This approach, as described in Ref. [32, 33], can be utilized similarly to general differentials for conducting a differential attack. Xie and Yang [22] have proposed the feasibility of executing quantum small-probability differential cryptanalysis by analyzing each component function of the encryption function individually. Hereafter, we illustrate a more efficient method to uncover small-probability differential characteristics of the target encryption algorithm using Algorithm 2, introducing the corresponding function:

$$\begin{aligned} F: \{0,1\}^n \times \{0,1\}^d &\to \{0,1\}^n \\ (m,k) &\to E_k^{(r-1)}(m) \end{aligned} \tag{25}$$

Using the mentioned function F, Algorithm 1 identifies small-probability differential characteristics (referred to as $(\Delta x, \Delta y)$) of the target encryption algorithm $E_k^{(r-1)}$. Subsequently, these characteristics are employed to validate the correctness of the guessed key and recover the accurate key of the target encryption algorithm. Our quantum small-probability differential attack mainly involves two processes (quantum part and classical part):

1. Find differential characteristic:

   Given the target encryption algorithm $E^{(r-1)}$, we define the function $F(m,k) = E_k^{(r-1)}$. After attackers execute Algorithm 2 $nl^2(n)q^2(n)$ times, they are highly likely to obtain the linear structure (a,i), where $q(n)$ and $l(n)$ represent any two polynomials about n. Then we set $b = (\overline{i_1 i_2 \cdots i_n})$, where $\overline{i_j} = i_j \oplus 1$. Thus, this process demonstrates that we can find a small-probability differential characteristic (a, b) of the target encryption algorithm in polynomial time.

2. Construct key-recovery attack:

   Using the identified small-probability differential characteristics, we can conduct a key-recovery attack on the target encryption algorithm. The attacker selects $l^2(n)$ plaintext differential pairs based on these characteristics and obtains corresponding ciphertext differential pairs $\Delta y^{(1)}, \Delta y^{(2)}, \ldots, \Delta y^{(l^2(n))}$ through encryption (for all possible keys), where $\Delta y^{(i)} = (\Delta y_1^{(i)}, \Delta y_2^{(i)}, \ldots, \Delta y_n^{(i)})$, $i = 1, 2, \ldots, l^2(n)$. Concurrently, all input pairs and their corresponding output differentials are logged. In essence, by analyzing the statistical distribution of output differentials and recording the occurrences of specific differentials ($\Delta y_j^i = b_j, j = 1, 2, \ldots, n$) as ($C_s$), attempts are made to deduce the key of the target encryption algorithm. Keys associated with lower ratio $\lambda_s = C_s / nl^2(n)$ may indicate potential correctness.

We can identify small-probability differential characteristics of the target encryption algorithm for most keys, as mentioned earlier. Under related-key conditions, such characteristics are accessible for any key. To be precise, the following theorem holds.

Theorem 4 *For most keys (with* $1 - \frac{1}{q(n)}$ *of keys, denoted as* K'*) in the first* $(r-1)$ *rounds of the target encryption algorithm, we have*

$$Pr[\lambda_s \geq \frac{1}{l(n)}] \leq 3e^{-n/2}$$

where s represents the correct key in the final round of the target encryption algorithm and $l(n)$ *is any polynomial in n.*

Proof By applying the above Theorem 2, it follows that

$$Pr[\frac{|x \in F_2^{n+d} | F(x \oplus a \| 0) \oplus F(x) = b|}{2^{n+d}} \leq \epsilon] \tag{26}$$

holds with the probability greater than $1 - exp(-2cn\epsilon^2)$. Clearly, we have

$$V(k) = \frac{|\{x \in F_2^n | E_k^{(r-1)}(x \oplus a) \oplus E_k^{(r-1)}(x) = b\}|}{2^n} \tag{27}$$

Equation (27) indicates that if we consider $V(k)$ as a variable, then for any $k \in \{0,1\}^d$, we have $E_k(V(k)) \leq \epsilon$. Consequently, for any polynomial $q(n)$ of n, we have

$$Pr_k[V(k) \leq q(n)\epsilon] \geq 1 - \frac{1}{q(n)} \tag{28}$$

The above Eq. (28) indicates that for the majority of keys (with $(1 - \frac{1}{q(n)})$ of keys, denoted as K'), we can find a small-probability differential characteristic of the target encryption algorithm. In other words, for any $k \in K'$, we have

$$Pr[\frac{|x \in F_2^n | E_k^{(r-1)}(x \oplus a) \oplus E_k^{(r-1)}(x) = b|}{2^n} \leq q(n)\epsilon] \tag{29}$$

holds with the probability greater than $1 - exp(-2cn\epsilon^2)$. Let $\epsilon = \frac{1}{2l(n)q(n)}$, $cn = nl^2(n)q^2(n)$, we have

$$Pr[\frac{|x \in F_2^n|E_k^{(r-1)}(x \oplus a) \oplus E_k^{(r-1)}(x) = b|}{2^n} \leq \frac{1}{2l(n)}] \tag{30}$$

holds with the probability greater than $1 - exp(-n/2)$. Then,

$$Pr_x[E_{kj}^{(r-1)}(x \oplus a) \oplus E_{kj}^{(r-1)}(x) = b_j] \leq \frac{1}{2l(n)} \tag{31}$$

holds for all $j = 1, 2, \ldots, n$. Next, we let Y be a random variable

$$Y = Y(i,j) \begin{cases} 1, & \triangle y_j^{(i)} = b_j \\ 0, & \triangle y_j^{(i)} \neq b_j \end{cases} \tag{32}$$

For every $i = 1, 2, \ldots, l^2(n)$, Eq. (31) indicates $E(Y) \leq \frac{1}{l^2(n)}$ except a negligible probability. Then, using Hoeffding's inequality, we have that

$$Pr[\frac{\Sigma_{i,j}Y}{nl^2(n)} \geq \frac{1}{2l(n)} + \delta] \leq 2e^{-2nl^2(n)\delta^2} + e^{-n/2} \tag{33}$$

Moreover for $\delta = \frac{1}{2l(n)}$, $\Sigma_{i,j}Y/nl^2(n) = \lambda_s$, we have

$$Pr[\lambda_s \geq \frac{1}{l(n)}] \leq 3e^{-n/2} \tag{34}$$

which completes the proof of Theorem 4. □

According to Theorem 4, $\lambda_s < \frac{1}{l(n)}$ for the correct key s. It's important to verify the recovered key and attempt to decrypt other ciphertexts for confirmation. If verification fails, restart the attack process. Our quantum small-probability differential attack requires only $nl^2(n)q^2(n)$ quantum queries. Compared to Xie and Yang's algorithm [22], ours achieves a polynomial speedup, reducing complexity from $n^4l^2(n)q^2(n)$ to $nl^2(n)q^2(n)$.

4.3 Applications to the quantum impossible differential cryptanalysis

Impossible differential analysis extends cryptographic differential analysis to overcome limitations and enhance attack efficacy for certain cryptographic algorithms, particularly block ciphers. It capitalizes on the property where certain input pair differences are unlikely to transform into specific output differences, even after several encryption rounds. That is, for an impossible differential characteristic $(\Delta x, \Delta y)$ of the target encryption algorithm $E_k^{(r-1)} : \{0,1\}^n \to \{0,1\}^n$, we have

$$E_k^{(r-1)}(x \oplus \Delta x) + E_k^{(r-1)}(x) \neq \Delta y, \forall x \in F_2^n \tag{35}$$

Similar to differential analysis, impossible differential analysis is also divided into two steps. Firstly, we identify an impossible differential characteristic $(\Delta x, \Delta y)$ based on the target encryption algorithm $E_k^{(r-1)}$, preventing the derivation of output difference Δy from

Algorithm 3 Quantum Impossible Differential Cryptanalysis

```
Input: Function $F = (f_1, f_2, \dots, f_n) \in C_{n+d,n}$; $\phi(x) := f_1 \oplus f_2 \oplus \cdots \oplus f_n$; Initialize $W$ as empty
Output: Return $(a, i_1 i_2 \cdots \overline{i_j} \cdots i_n)$
Repeat $cn$ times
    Run the BV algorithm on $\phi$                                  ▷$y = y_1 y_2 \cdots y_n y_{n+1} \cdots y_{n+d}$
    Set $W = W \cup \{y_1 y_2 \cdots y_n\}$
EndRepeat
if if $\{y \cdot a = i_\phi | y \in W, i_\phi \in \{0,1\}\}$ has no solution then
    return “failure"
else
    return the nonzero solution $A^{\overline{i_\phi}}$ for $i_\phi \in \{0,1\}$          ▷$A^{\overline{i_\phi}}$
end if
Find $a^{i_\phi}$, and calculate $i_1 i_2 \cdots i_n = E^{(r-1)}(x \oplus a) \oplus E^{(r-1)}(x)$      ▷$(a^{i_\phi}, i)$
Generate a random $j \in \{1, 2, \dots, n\}$ such that $i_1 \oplus i_2 \oplus \cdots \oplus \overline{i_j} \oplus \cdots \oplus i_n = \overline{i_\phi}$
▷$(a, i_1 i_2 \cdots \overline{i_j} \cdots i_n)$
```

input difference Δx during the encryption process. Secondly, leveraging the identified impossible differential characteristic, we attempt to deduce the correct key s for the target encryption algorithm.

Our attack primarily targets the initial phase of impossible differential analysis, aiming to identify the impossible differential characteristic of the target encryption algorithm efficiently. Based on the target encryption algorithm, we construct a function

$$\begin{aligned} F : \{0,1\}^n \times \{0,1\}^d &\to \{0,1\}^n \\ (m, k) &\mapsto E_k^{(r-1)}(m) \end{aligned} \tag{36}$$

We can find the impossible linear structure of the above function using Algorithm 1. This structure corresponds to the impossible differential characteristics of the target encryption algorithm. We encapsulate this attack process in Algorithm 3.

According to Theorem 1, for $\varepsilon(F, a) \le p_0 < 1$, if Algorithm 3 returns vector $(a, i_1 i_2 \cdots \overline{i_j} \cdots i_n)$ after $O(n)$ queries, then $(a, i_1 i_2 \cdots \overline{i_j} \cdots i_n)$ corresponds to an impossible linear structure of function F, where $\varepsilon(F, a) := \max_{1 \le j \le n} \varepsilon(f_j, a)$. This corresponds to an impossible differential attack, where $(a, \times \times \cdots \overline{i_j} \cdots \times)$ represents a characteristic of the target encryption algorithm, with “×” denotes either bit can be 0 or 1. Specifically, we state the following theorem.

Theorem 5 *Given a vector function F with $\varepsilon(F, a) \le p_0 < 1$ for some constant p_0, if Algorithm 3 returns vector $(a, i_1 i_2 \cdots \overline{i_j} \cdots i_n)$ after cn queries, for any key $k \in \{0,1\}^d$ and $i_1 \cdots i_{j-1} i_{j+1} \cdots i_n \in \{0,1\}$, we have*

$$F(x \oplus a \| 0) \oplus F(x) \neq i_1 i_2 \cdots \overline{i_j} \cdots i_n, \forall x \in F_2^{n+d} \tag{37}$$

except for a negligible probability. This indicates $(a, i_1 i_2 \cdots \overline{i_j} \cdots i_n)$ as an improbable differential characteristic of the target encryption algorithm.

Proof Based on Theorem 1, it follows that $Pr[a \in U_F^i] > 1-(2p_0^c)^n$. Thus $Pr[F(x \oplus a\|0) \oplus F(x) \neq i_1 i_2 \cdots \overline{i_j} \cdots i_n, \forall x \in F_2^{n+d}] > 1-(2p_0^c)^n$. This indicates for all $k \in \{0,1\}^d$,

$$Pr[E^{(r-1)}(x \oplus a) \oplus E^{(r-1)}(x) \neq i_1 i_2 \cdots \overline{i_j} \cdots i_n, \forall x \in F_2^n] > 1-(2p_0^c)^n \tag{38}$$

Then, the conclusion holds. □

Theorem 5 demonstrates that Algorithm 3 effectively identifies the impossible differential characteristics of the target encryption algorithm $E^{(r-1)}$. Compared with the approach of Xie and Yang [22], our algorithm achieves a quadratic speedup. Unlike previously discussed methods, this impossible differential path applies to all keys, whereas high/small-probability differential paths are partial-key.

It is important to note that Algorithm 3 is designed to find impossible differentials focused on specific bits. Although its application scope is restricted to particular impossible differential paths, the algorithm introduces a novel perspective for impossible differential analysis. Traditional impossible differential analysis faces the challenge of extending the differential path, limiting the attackable rounds. By considering the initial $r-1$ rounds as a single unit, our approach successfully bypasses this limitation.

5 Conclusion

This paper introduces a novel quantum algorithm for finding approximate linear structures of vector functions. Furthermore, recognizing that the linear structures within an encryption function often correspond to high-probability differentials, we improve three methodologies for conducting differential cryptanalysis. Note that, all three approaches leverage quantum algorithms with polynomial time complexity. Future research should focus on advancing quantum algorithms for differential and low-probability differential cryptanalysis to comprehensively cover the entire key space while simplifying implementation, such as through circuit optimization. Achieving practical applicability in the near term remains an open challenge.

Appendix A: Proof of Theorem 1

Theorem 1 *If* $\varepsilon(F,a) \le p_0 < 1$, *Algorithm 1, after cn quantum queries, can still successfully recover the linear structure* (a,i) *with a probability of at least* $1-(2p_0^c)^n$.

Proof Before proving Theorem 1, first we will review a lemma.

Lemma 2 ([28]) *For a given Boolean function* $f \in B_n$ *and any* $a \in \{0,1\}^n$, $i \in \{0,1\}$, *we have*

$$\sum_{\omega \cdot a = i} S_f^2(\omega) = \frac{|\{x \in F_2^n | f(x \oplus a) \oplus f(x) = i\}|}{2^n} \tag{A1}$$

Based on the lemma provided, when executing algorithm subroutine (step 2) for the constructed function ϕ, the probability of obtaining the vector y satisfying the condition $y \cdot a = i_\phi$ is

$$Pr[y \cdot a = i_\phi] = \sum_{y \cdot a = i_\phi} S_\phi^2(y) = \frac{|\{x \in F_2^n | \phi(x \oplus a) \oplus \phi(x) = i_\phi\}|}{2^n} \tag{A2}$$

For convenience in our subsequent proof, we set $y \cdot a = 0$. The parameter $\varepsilon(F,a) \leq p_0 < 1$ implies $\varepsilon(\phi,a) \leq p_0 < 1$. More precisely:

$$\begin{aligned}
\varepsilon(\phi,a) &= \frac{1}{2^n} \max_{t \in F_2^n \setminus U_f} |\{x \in F_2^n | \phi(x \oplus t) \oplus \phi(x) = 0\}| \\
&\leq \frac{1}{2^n} \max_{1 \leq j \leq n} \max_{t \in F_2^n \setminus U_f} |\{x \in F_2^n | f_j(x \oplus t) \oplus f_j(x) = 0\}| \\
&= \varepsilon(F,a) \\
&\leq p_0
\end{aligned} \tag{A3}$$

In the subroutine of Algorithm 1 (step 2), running the BV algorithm on function ϕ initially yields a vector y_1 orthogonal to a, and subsequent runs yield another vector y_2 orthogonal to a. Repeating this step cn times, with high probability, yields $n-1$ orthogonal and independent vectors $y_1, y_2, \ldots, y_{n-1} \in \{0,1\}^n$, thus obtaining a unique non-zero solution. More precisely:

$$\begin{aligned}
p_{fail} &= Pr[\exists t \in F_2^n \setminus \{0,a\} s.t., y_1 \cdot t = y_2 \cdot t = \cdots = y_{cn} \cdot t = 0] \\
&\leq \Sigma_{t \in F_2^n \setminus \{0,a\}} Pr[y_1 \cdot t = y_2 \cdot t = \cdots = y_{cn} \cdot t = 0] \\
&\leq \Sigma_{t \in F_2^n \setminus \{0,a\}} (Pr[y_1 \cdot t = 0] \cdot Pr[y_2 \cdot t = 0] \cdot \cdots \cdot Pr[y_{cn} \cdot t = 0]) \\
&\leq 2^n \max_{t \in F_2^n \setminus \{0,a\}} (Pr[y \cdot t] = 0)^{cn} \\
&\leq (2p_0^c)^n
\end{aligned} \tag{A4}$$

After cn queries, Algorithm 1 has a probability of success

$$p_{succ} = 1 - p_{fail} \geq 1 - (2p_0^c)^n \tag{A5}$$

in recovering the linear structure a of function ϕ, as shown in Eq. (A4). After successfully recovering the linear structure a, (a,i) can be obtained through simple calculations, where $i = F(x \oplus a) \oplus F(x)$. Thus, the conclusion holds. □

Appendix B: Proof of Theorem 2

Theorem 2 *When Algorithm 1 is executed on the vector function $F = (f_1, f_2, \ldots, f_n)$ for cn times, if Algorithm 1 returns vector (a,i), then for any ϵ $(0 < \epsilon < 1)$, we have*

$$Pr[\frac{|\{x \in F_2^n | F(x \oplus a) \oplus F(x) = i\}|}{2^n} > 1 - \epsilon] > 1 - e^{-2cn\epsilon^2} \tag{B1}$$

Proof Before proving Theorem 2, first we will review Hoeffding's inequality.

Hoeffding's inequality [34] The empirical expectation $\overline{X}$ of a set of independent and identically distributed random variables $X_i \in [a_i, b_i]$, $i = 1, 2, \ldots, n$, satisfies the inequality

$$P(|\overline{X} - E[\overline{X}]| \geq t) \leq e^{-\frac{2n^2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}} \tag{B2}$$

where $t > 0$ denotes the deviation between the sum of random variables and its expectation.

According to Lemma 2, in the execution of the algorithm subroutine (step 2) on the function ϕ, the probability of obtaining a vector y satisfying $y \cdot a = i_\phi$ is

$$Pr[y \cdot a = i_\phi] = \frac{|\{x \in F_2^n | \phi(x \oplus a) \oplus \phi(x) = i_\phi\}|}{2^n} = p \tag{B3}$$

Then

$$Pr[y \cdot a = \overline{i_\phi}] = \frac{|\{x \in F_2^n | \phi(x \oplus a) \oplus \phi(x) = \overline{i_\phi}\}|}{2^n} = 1 - p = q \tag{B4}$$

where $p, q \in [0, 1]$ represent the corresponding probability. Next, we set X be a variable

$$X(y) = \begin{cases} 0, & y \cdot a = i_\phi \\ 1, & y \cdot a = \overline{i_\phi} \end{cases} \tag{B5}$$

The expectation of this variable is $E(X) = 1 \cdot q = q = 1 - p$, as evident from the formula above. In the subroutine of Algorithm 1 (step 2), a vector y_1 is obtained in the first execution. By repeating this step cn times, cn vectors $y_1, y_2, \ldots, y_{cn}$ are obtained, corresponding to cn independent and identically distributed random variables $X_1, X_2, \ldots, X_{cn}$. Then, from Hoeffding's inequality [34], we can get $Pr[q \geq \epsilon] \leq e^{-2cn\epsilon^2}$. More precisely:

$$Pr[q - \frac{1}{cn} \Sigma_{l=1}^{cn} X_l \geq \epsilon] \leq e^{-2cn\epsilon^2} \tag{B6}$$

Algorithm 1 cannot output vector (a, i) when $Y_l = 1$ $(l = 1, 2, \ldots, n)$, implying $\frac{1}{cn} \Sigma_{l=1}^{cn} Y_l = 0$. Therefore, the above Eq. (B6) can be simply rewritten as

$$Pr[q \geq \epsilon] \leq e^{-2cn\epsilon^2} \tag{B7}$$

From Eq. (B7), it follows

$$Pr[1 - p < \epsilon] = Pr(1 - \epsilon < p \leq 1) > 1 - e^{-2cn\epsilon^2} \tag{B8}$$

Thus the probability that

$$Pr[y \cdot a = i_\phi] = \frac{|\{x \in F_2^n | \phi(x \oplus a) \oplus \phi(x) = i_\phi\}|}{2^n} > 1 - \epsilon \tag{B9}$$

holds is greater than $1 - e^{-2cn\epsilon^2}$. Thus the conclusion holds. □

Abbreviations

BV Algorithm, Bernstein-Vazirani Algorithm; MACs, Message Authentication Codes; DES, Data Encryption Standard; AES, Advanced Encryption Standard.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant Nos. 62471070 and 62271234), the Fundamental Research Funds for Heilongjiang Universities, Open Foundation of State Key Laboratory of Public Big Data (Guizhou University) under Grant No. PBD2022-16, Double First-Class Project for Collaborative Innovation Achievements in Disciplines Construction in Heilongjiang Province under Grant No. LJGXCG2022-054.

Author contributions

Rongxue Xu and Hongwei Sun wrote the main manuscript text, and Kejia Zhang, Gang Du and Dandan Li designed all the algorithms. All authors reviewed the manuscript.

Funding

This work is supported by the National Natural Science Foundation of China (Grant Nos. 62471070 and 62271234), the Fundamental Research Funds for Heilongjiang Universities, Open Foundation of State Key Laboratory of Public Big Data (Guizhou University) under Grant No. PBD2022-16, Double First-Class Project for Collaborative Innovation Achievements in Disciplines Construction in Heilongjiang Province under Grant No. LJGXCG2022-054.

Data Availability

No datasets were generated or analysed during the current study.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

The Author confirms: that the work described has not been published before; that it is not under consideration for publication elsewhere; that its publication has been approved by all co-authors.

Competing interests

The authors declare no competing interests.

Author details

[1]School of Computer and Big Data (School of Cybersecurity), Heilongjiang University, Harbin 150080, China. [2]School of Mathematical Science, Heilongjiang University, Harbin 150080, China. [3]School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing 100876, China.

Received: 29 April 2024 Accepted: 3 December 2024 Published online: 18 December 2024

References

1. Shor PW. Algorithms for quantum computation: discrete logarithms and factoring. In: 35th annual symposium on foundations of computer science. Los Alamitos: IEEE Comput. Soc.; 1994. p. 124–34.
2. Grover LK. A fast quantum mechanical algorithm for database search. In: Miller GL, editor. Proceedings of the twenty-eighth annual ACM symposium on the theory of computing. Philadelphia, Pennsylvania, USA, May 22-24, 1996. New York: ACM; 1996. p. 212–9.
3. Song Y, Wu Y, Wu S, Li D, Wen Q, Qin S, Gao F. A quantum federated learning framework for classical clients. Sci China, Phys Mech Astron. 2024;67:250311.
4. Song Y, Li J, Wu Y, Qin S, Wen Q, Gao F. A resource-efficient quantum convolutional neural network. Front Phys. 2024;12:1362690.
5. Rivest RL, Shamir A, Adleman L. A method for obtaining digital signatures and public-key cryptosystems. Commun ACM. 1978;21:120–6.
6. Kuwakado H, Morii M. Quantum distinguisher between the 3-round Feistel cipher and the random permutation. In: 2010 IEEE international symposium on information theory proceedings (ISIT), June 2010. 2010. p. 2682–5.
7. Kuwakado H, Morii M. Security on the quantum-type even-mansour cipher. In: ISITA. IEEE; 2012. p. 312–6.
8. Kaplan M, Leurent G, Leverrier A, et al. Breaking symmetric cryptosystems using quantum period finding. In: CRYPTO 2016, Part II. 2016. p. 207–37.
9. Sun HW, Wei CY, Cai BB, et al. Improved BV-based quantum attack on block ciphers. Quantum Inf Process. 2023;22:9. https://doi.org/10.1007/s11128-022-03752-x.
10. Sun HW, Cai BB, Qin SJ, et al. Quantum attacks on beyond-birthday-bound MACs. Phys A, Stat Mech Appl. 2023;625:129047.
11. Sun HW, Cai BB, Qin SJ, et al. Quantum attacks on type-1 generalized feistel schemes. Adv Quantum Technol. 2023;6(10):2300155.
12. Li Z, Cai B, Sun H, et al. Novel quantum circuit implementation of advanced encryption standard with low costs. Sci China, Phys Mech Astron. 2022;65:290311.
13. Dong X, Wang X. Quantum key-recovery attack on Feistel structures. Sci China Inf Sci. 2018;61(10):102501.
14. Dong X, Li Z, Wang X. Quantum cryptanalysis on some generalized Feistel schemes. Sci China Inf Sci. 2019;62(2):22501.
15. Dong X, Dong B, Wang X. Quantum attacks on some Feistel block ciphers. Des Codes Cryptogr. 2020;88(6):1179–203.
16. Chen H, Li Y, Abla P, et al. In: Quantum algorithm for finding impossible differentials and zero-correlation linear hulls of symmetric ciphers. Australasian conference on information security and privacy. Switzerland: Springer; 2023. p. 431–51.
17. Simon DR. On the power of quantum computation. SIAM J Comput. 1997;26(5):1474–83.
18. Leander G, Grover AM. Meets Simon - quantumly attacking the FX-construction. In: Advances in cryptology - ASIACRYPT. 2017. p. 161–78.
19. Bernstein E, Vazirani UV. Quantum complexity theory. SIAM J Comput. 1997;26(5):1411–73.
20. Biham E, Shamir A. Differential cryptanalysis of DES-like cryptosystems. J Cryptol. 1991;4:3–72. https://doi.org/10.1007/BF00630563.

21. Li H, Yang L. Quantum differential cryptanalysis to the block ciphers. In: Applications and techniques in information security: 6th international conference, ATIS 2015, proceedings 6. Beijing, China, November 4-6, 2015. Berlin: Springer; 2015. p. 44–51.
22. Xie H, Using YL. Bernstein-Vazirani algorithm to attack block ciphers. Des Codes Cryptogr. 2019;87:1161–82.
23. Zhou Q, Lu S, Zhang Z, et al. Quantum differential cryptanalysis. Quantum Inf Process. 2015;14:2101–9.
24. Leurent G, Kaplan M, Leverrier A, Naya-Plasencia M. Quantum differential and linear cryptanalysis. FSE 2017-Fast Software Encryption, Mar 2017, Tokyo, Japan.
25. O'connor L, Klapper A. Algebraic nonlinearity and its applications to cryptography. J Cryptol. 1994;7(4):213–27.
26. Li H, Yang L. A quantum algorithm to approximate the linear structures of Boolean functions. Math Struct Comput Sci. 2018;28:1–13.
27. Xie H, Yang L. A quantum related-key attack based on the Bernstein-Vazirani algorithm. Quantum Inf Process. 2020;19(8):1–20.
28. Dubuc S. Characterization of linear structures. Des Codes Cryptogr. 2001;22:33–45.
29. Tudorache AG, Manta VI, Caraiman S. Implementation of the Bernstein-Vazirani quantum algorithm using the qiskit framework. Bulletin of the Polytechnic Institute of Ia?i Electrical Engineering, Power Engineering, Electronics Section. 2021;67(2):31–40.
30. Hosoyamada A, Quantum SY. Demiric-Selçuk meet-in-the-middle attacks: applications to 6-round generic Feistel constructions. In: Security and cryptography for networks: 11th international conference, SCN 2018, proceedings, vol. 11. Amalfi, Italy, September 5-7, 2018. Berlin: Springer; 2018. p. 386–403.
31. Hao X, Zhang F, Wei Y, et al. Quantum period finding based on the Bernstein-Vazirani algorithm. Quantum Inf Comput. 2020;20(1–2):65–84.
32. Borst J, Knudsen LR, Rijmen V. Two attacks on reduced IDEA. In: International conference on the theory and applications of cryptographic techniques. Berlin: Springer; 1997. p. 1–13.
33. Knudsen LR, Rijmen V. On the decorrelated fast cipher (DFC) and its theory. In: International workshop on fast software encryption. Berlin: Springer. 1999. p. 81–94.
34. Hoeffding W. Probability inequalities for sums of bounded random variables. In: The collected works of Wassily Hoeffding. 1994. p. 409–26.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.