Noname manuscript No.
(will be inserted by the editor)

Quantum Meet-in-the-Middle Attack on 7-round Feistel Construction

Yinsong, Xu · Zheng, Yuan*

Received: date / Accepted: date

Abstract Quantum attacks on Feistel constructions have attracted much attention from worldwide cryptologists. To reduce the time complexity of quantum attacks on 7-round Feistel construction, we propose a novel quantum meet-in-the-middle (QMITM) attack in Q1 model. Inspired by Hosoyamada *et al.*'s work [17], we have introduced quantum computing in the offline computation of the classic meet-in-the-middle (MITM) attack [22]. In the attack, the differential characteristic of 7-round Feistel construction is given via 5-round distinguisher firstly. And then we propose a quantum claw finding algorithm based on quantum walk, which speeds up the process of finding a match in the offline computation phase. The keys in 7-round Feistel construction could be recovered by the match at last. Compared with quantum attacks in Q2 model, our attack reduces the time complexity from $O(2^{3n/4})$ to $O(2^{2n/3})$, and is significantly better than classic attacks. Moreover, our attack belongs to Q1 model, which is more practical than Q2 model.

Keywords Quantum meet-in-the-middle attack · 7-round Feistel construction · Quanutm claw finding algorithm · 5-round distinguisher · Quantum walk

Y. Xu
School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, 100876, China.
Beijing Electronic Science and Technology Institute, Beijing, 100070, China.
E-mail: mugongxys@foxmail.com

Z. Yuan
*Corresponding Author
School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, 100876, China.
Beijing Electronic Science and Technology Institute, Beijing, 100070, China.
School of Cyberspace Science and Technology, University of Science and Technology of China, Hefei, 230026, China.
E-mail: zyuan@tsinghua.edu.cn

1 Introduction

Since the Shor's algorithm [1] was proposed, public key cryptosystems have been broken, such as RSA and ECC. Researchers hope to use new quantum algorithms to break other cryptosystems, especially the symmetric cryptosystems. They use Grover algorithm [2] to reduce the cost of exhaustive search on an n-bit key from 2^n to $2^{n/2}$ in quantum setting early. However, it is observed that doubling the key length of one block cipher could achieve a similar amount of security against quantum attackers.

In 2010, Kuwakado *et al.* [3] proposed a quantum 3-round distinguisher of Feistel construction based on the Simon's algorithm [4], which reduces the time complexity of key recovery from $O(2^n)$ to $O(n)$. Then, more and more specific constructions have been evaluated under post-quantum security, e.g. against Even-Mansour cipher [5], CBC-like MACs [6,7], AEZ [8], AES-COPA [9], FX construction [10], Feistel constructions [11–17], etc. Kaplan divides these quantum attacks into two attack models according to the adversary's ability: Q1 model and Q2 model [18].

Q1 model: The adversary is allowed to make classical online queries and perform quantum computation offline.

Q2 model: The adversary is allowed to make quantum superposition online queries on quantum cryptographic oracle. That is, oracles allow queries in quantum superposition states and return the results as quantum superposition states.

It can be seen that the adversary is more practical in Q1 model, and more powerful in Q2 model. In other words, if the adversary is not permitted access to quantum cryptographic oracle, the attack in Q2 model would not work.

Among these quantum attacks, the quantum attack on r-round Feistel constructions is a hot topic, and many ways have been proposed to reduce time complexity (as shown in Table 1). Note that, Feistel ciphers have many popular constructions. The construction we discussed in this paper is displayed in Fig.1, in which an n-bit state is divided into $n/2$-bit halves denoted by a_i and b_i, and the state is updated by iteratively applying the following two operations:

$$a_{i+1} \leftarrow b_i \qquad b_{i+1} \leftarrow a_i \oplus F(k_i \oplus b_i)$$

where F is a public function.

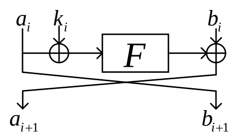

Fig. 1: Our target Feistel construction (called Feistel-KF [15]).

The first approach is to use quantum slide attack based on Simon's algorithm in Q2 model, which can reduce time complexity from $O(2^{rn/2})$ to $O(n^2)$

Table 1: Quantum attacks on Feistel constructions

Ref.	Setting	Round	Time Complexity
[11]	Q2	4	$O(n)$
[12]	Q2	4	$O(n^2)$
[13]	Q2	r $(r > 3)$	$O(2^{0.25nr-0.75n})$
[15]	Q2	r $(r > 4)$	$O(2^{0.25nr-n})$
[16]	Q2	3	$O(n^2)$
[17]	Q1	6	$O(2^n)$
Ours	Q1	7	$O(2^{2n/3})$

or even $O(n)$ when $r = 4$ [11,12]. Secondly, Grover's algorithm and quantum 3-round Feistel distinguisher are combined in Q2 model, whose time complexity is $O(2^{0.25nr-0.75n})$ [13, 14]. Furthermore, Ito *et al.* [15] proposed a new quantum 4-round Feistel distinguisher based on Simon's algorithm in Q2 model. Compared with the front one, its time complexity is reduced by a factor of $2^{0.25n}$. Next, Xie *et al.* [16] present a quantum algorithm for finding the linear structures of a function. And based on it, Xie further proposed new quantum distinguishers for 3-round Feistel scheme. This work provides a novel attack idea, but its complextity is $O(n^2)$, higher than Kuwakado *et al.*'s work [3]. Different from the above attacks in Q2 model, Hosoyamada *et al.*'s attack on 6-round Feistel construction is in Q1 model [17]. Hosoyamada proposed a quantum claw finding algorithm to find a match between two lookup tables in the classical meet-in-the-middle attacks, which can speed-up the search by Grover's algorithm.

Our Contribution. Inspired by Hosoyamada *et al.*'s work [17], we propose a new quantum meet-in-the-middle attack (QMITM) to reduce the time complexity of attacks on 7-round Feistel construction in Q1 model. In the classic meet-in-the-middle attacks on 7-round Feistel construction, a match pair should be found from two different lookup tables that constructed in the offline computation phase. Therefore, we convert this problem into finding a claw between two functions f and g, i.e., *claw finding problem.* Normally, a claw between functions f and g is defined as a pair (x, y) such that $f(x) = g(y)$.

Moreover, whether in the classical or the quantum setting, how to find a match pair consumes a major amount of time in the entire attack. We need to improve the claw finding algorithm by quantum algorithms to reduce time complexity. Different from Hosoyamada's quantum claw finding algorithm based on Grover's algorithm, our quantum claw finding algorithm is based on quantum walk algorithm, which is the variant of Zhang's algorithm for *(m,n) 2-subset finding problem* [19].

Analysis shows that the time complexity of our attack is $O(2^{2n/3})$, which is lower than the best quantum attack - $O(2^{3n/4})$ [15]. Although our attack needs $O(n2^{n/2})$ qubits higher than quantum attacks in Q2 model, our attack is more practical in Q1 model. Besides, we reduce the queries and adjust truncated differentials during the distinguisher analysis, which reduces the memory complexity by $2^{n/4}$ compared with classic attack [23].

Algorithm 1 Quantum Walk on Set S in Set A

Input: State $|S, x_S, y\rangle$ and A with $S \subseteq A$, and $y \in A - S$, where $|S| = r$, $|A| = N$, and x_S contains the variable values x_i for all $i \in S$.

1: $|S, x_S, y\rangle \to |S, x_S\rangle \left((-1 + \frac{2}{N-r})\,|y\rangle + \frac{2}{N-r} \sum_{y' \in A-S-\{y\}} |y'\rangle\right)$.

2: $|S, x_S, y\rangle \to \left|S \cup \{y\}, x_{S\cup\{y\}}, y\right\rangle$ by querying x_y and inserting it corresponding to y.

3: $|S, x_S, y\rangle \to |S, x_S\rangle \left((-1 + \frac{2}{r+1})\,|y\rangle + \frac{2}{r+1} \sum_{y' \in S-\{y\}} |y'\rangle\right)$.

4: $|S, x_S, y\rangle \to \left|S - \{y\}, x_{S-\{y\}}, y\right\rangle$ by erasing x_y in x_S and removing y from S.

This paper is organized as follows. Sect. 2 provides a brief description of quantum walk algorithm for element distinctness and claw finding problems, and 5-round distinguisher. And our quantum meet-in-the-middle attack on 7-round Feistel construction is presented in Sects. 3. Then, the comparison with classical and other quantum attacks is analyzed in Sect. 4, followed by a conclusion in Sect. 5.

2 Preliminaries

In this section, we would briefly introduce how quantum walk algorithm to solve element distinctness and claw finding problems, and show the 5-round distinguisher of Feistel construction.

2.1 Quantum Walk Algorithm for Element Distinctness and Claw Finding

Problem 1 ELEMENT DISTINCTNESS [20]: Given numbers x_1, ... , $x_N \in [N]$, are they all distinct?

In the classical setting, the natural solution to the element distinctness problem is done by sorting, which requires $O(N)$ queries. In the quantum setting, Buhrman et al. [21] gave the first nontrivial quantum algorithm that uses $O(N^{3/4})$ queries. Ambainis [20] then designed a new quantum algorithm inspired of quantum walk, which only needs $O(N^{2/3})$ queries and is proved to be the lower bound [19].

In Ambainis's algorithm (Algorithm 2), its working state is a superposition state in the form of $|S, x_S, y\rangle$, where S is a r-size subset of $[N]$, x_S contains the variable values x_i for all $i \in S$, and y is an index not in S. Note that, **Quantum Walk** (Algorithm 1) is used as an basic tool in Algorithm 2. Let $I = \{i_1, \ldots, i_k\}$ be the k-tuple of equal elments, i.e., $x_{i_1} = \ldots = x_{i_k}$. In Algorithm 1, steps 1-2 insert the index $y \in I$ into S, and steps 3-4 remove one index $y \notin I$ in S.

Then, Zhang [19] applied Ambainis's algorithm to solve the following problems (Problems 2 and 3) to give Algorithm 3, which meets the requirements of Theorem 1.

Problem 2 (m,n) 2-SUBSET FINDING [19]: Given x_1, ... , $x_N \in [M]$, two sets of indices J_1, $J_2 \subseteq [N]$ with $|J_1| = m$, $|J_2| = n$, and a relation $R \subseteq$

Algorithm 2 k-Element Distinctness

Input: x_1, ..., $x_N \in [M]$, with there exists at most one k-size set $I = \{i_1, \dots, i_k\} \subseteq [N]$ *s.t.* $x_{i_1} = \dots = x_{i_k}$.
Output: I and $x_I = \{x_{i_1}, \dots, x_{i_k}\}$ if they exist; otherwise reject.

1. Create the initial state $\frac{1}{\sqrt{\binom{N}{r}(N-r)}} \sum_{S \subseteq [N], |S|=r, y \in [N]-S} |S\rangle |y\rangle$.
2. Query all x_i $(i \in S)$ to get $\frac{1}{\sqrt{\binom{N}{r}(N-r)}} \sum_{S \subseteq [N], |S|=r, y \in [N]-S} |S, x_i, y\rangle$.
3. Repeat $O((\frac{N}{r})^{k/2})$ times:
   (a) Apply the phase flip $|S, x_i, y\rangle \to -|S, x_i, y\rangle$ when $I \subseteq S$.
   (b) Do **Quantum Walk** on S in $[N]$ for $O(\sqrt{r})$ times.
4. Measure the result state and give the corresponding answer.

Algorithm 3 (m,n) 2-Subset Finding

Input: x_1, ... , $x_N \in [M]$. J_1, $J_2 \subseteq [N]$, $|J_1| = m$, $|J_2| = n$. $R \subseteq [M] \times [M]$ *s.t.* there exists at most one pair of $(x_{j_1}, x_{j_2}) \in R$ with $j_1 \in J_1$, $j_2 \in J_2$ and $j_1 \neq j_2$.
Output: Output the pair (j_1, j_2), otherwise reject.

1. Create the initial state $|\psi_{start}\rangle = \frac{1}{\sqrt{T}} \sum_{S_b \subseteq J_b, |S_b|=r_b, y_b \in J_b - S_b} \left|S_1, x_{S_1}, y_1, S_2, x_{S_2}, y_2\right\rangle$, where $T = \binom{m}{r_1}\binom{n}{r_2}(m-r_1)(n-r_2)$ and $b = 1, 2$.
2. Repeat following operations $O(\sqrt{\frac{mn}{r_1 r_2}})$ times
   (a) Check whether the pair (j_1, j_2) is in $S_1 \times S_2$. If it exits, do the following phase flip: $\left|S_1, x_{S_1}, y_1, S_2, x_{S_2}, y_2\right\rangle \to -\left|S_1, x_{S_1}, y_1, S_2, x_{S_2}, y_2\right\rangle$.
   (b) Do **Quantum Walk** on S_1 in J_1 for $t_1 = \lceil \frac{\pi}{4}\sqrt{r_1} \rceil$ times.
   Do **Quantum Walk** on S_2 in J_2 for $t_2 = \lceil \frac{\pi}{8}\sqrt{r_2} \rceil$ times.
3. Measure the resulting state and give the corresponding answer.

$[M] \times [M]$, with the promise that there exists at most one pair of $(x_{j_1}, x_{j_2}) \in R$ *s.t.* $j_1 \in J_1$, $j_2 \in J_2$ and $j_1 \neq j_2$. Output the unique pair if it exists, and reject otherwise.

Problem 3 CLAW FINDING [19]: The above problem with the restrictions that R is the *Equality* relation and $J_1 \cap J_2 = \emptyset$.

Theorem 1 *[19] Algorithm 3 outputs desired results correctly in the function-evaluation, and we can pick r_1 and r_2 to make number of queries be*

$$\begin{cases} O((mn)^{1/3}) & if \quad \sqrt{n} \leq m \leq n^2 & r_1 = r_2 = (mn)^{1/3} \\ O(\sqrt{n}) & if \quad m < \sqrt{n} & r_1 = m, r_2 \in [m, (mn)^{1/3}] \\ O(\sqrt{m}) & if \quad m > n^2 & r_1 \in [n, (mn)^{1/3}], r_2 = n \end{cases}$$

2.2 5-round Distinguisher

The 5-round distinguisher is the differential characteristics of 5-round Feistel construction proposed by Guo *et al.* [22], which is described by Lemma 1 and Proposition 1. And the differential characteristic is shown in Fig. 2, where v_i $(0 \leq i \leq 6)$ is the input or output of each round, X, X', 0 and X'' represent differences.

Lemma 1 *[22] Let X and X', where $X \neq X'$, be two non-zero branch differences. If a 5-round Feistel encrypts a pair of plaintexts (m, m') with a difference $0||X$ to a pair of ciphertexts with difference $0||X'$, then the number of possible internal state values of the three middle rounds that correspond to the plaintext m is limited to $2^{n/2}$ on average.*

Proof With input difference $0||X$ and output difference $0||X'$ is shown in Fig. 2, we can see that after the first round, the input difference $(0, X)$ must become a state difference $(X, 0)$. Similarly, after the inversion of the last round the output difference $(0, X')$ becomes $(0, X')$. Guo denotes output difference $\Delta F_3^O = X'' = X \oplus X'$ and input difference $\Delta F_3^I = \Delta$ at 3rd round. Then, both output differences ΔF_2^O and ΔF_4^O have the same difference Δ. To summarize, we get that for each fixed Δ, the input and output differences of the round functions at rounds 2, 3 and 4 are fixed. Therefore, there exists one state value (one solution) that satisfies such input-output difference in each of the three rounds. Therefore, there exists one state value (one solution) that satisfies such input-output difference in each of the three rounds. As Δ can take at most $2^{n/2}$ different values, the internal state values of the three middle rounds can assume $2^{n/2}$ different values.

Proposition 1 *[22] Let (m, m') be a pair of plaintexts that conforms to the 5-round distinguisher and $F^{\Delta}(m, \delta) = Trunc_{n/2}(F(m) \oplus F(m \oplus (0||\delta)))$, where F is 5-round Feistel, $Trunc_{n/2}$ is the truncation of the first $n/2$ bits, $m \in \{0, 1\}^n$ and $\delta \in \{0, 1\}^{n/2}$. Let $\delta_j = 1, \ldots, 2^b - 1$, $b \geq 1$ form b-δ-sequence. Then, the sequence $F^{\Delta}(m, \delta_j)$, $\delta_j = 1, \ldots, 2^b - 1$ can assume only $2^{n/2}$ possible values.*

Proof $F^{\Delta}(m, \delta)$ gives the output difference of the left half of the pair of the ciphertexts, produced by encryption of a pair of plaintexts $(m, m \oplus (0||\delta))$ with 5 round Feistel. Assume the difference δ is fixed, and let us consider a new pair of plaintexts $F^{\Delta}(m, \delta_j)$. Finally, $\Delta v_5 = \Delta F_4^O \oplus \Delta F_2^O$. In summary, for an arbitrary δ_j, the mapping from δ_j to Δv_5 becomes deterministic (as long as Δ is fixed). Therefore, the mapping is determined from the value of Δ, X and X', and acts independently of the value of m. Since Δ takes at most $2^{n/2}$ values, the number of sequences of Δv_5 is limited to $2^{n/2}$. The detailed proof can be seen in Ref. [22].

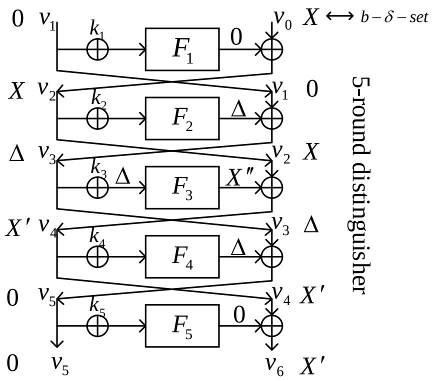

Fig. 2: The differential characteristic of the 5-round Feistel construction. v_i $(0 \leq i \leq 6)$ is the input or output of each round, X, X', 0, Δ and X'' represent the input or output differences of the round functions F_i $(0 \leq i \leq 5)$. The difference X belongs to $b-\delta-set$, i.e., has a set of 2^b state values that are all different in b state bits (the active bits) and are all equal in the remaining state bits (the inactive bits). And k_i $(0 \leq i \leq 5)$ is the $n/2$-bit key of each round.

3 Quantum Meet-in-the-Middle Attack on 7-round Feistel Construction

Our proposed quantum MITM attack still belongs to the Q1 model, i.e., the adversary can use a quantum computer to perform offline computation, and can only perform classic online queries. Similar to classic meet-in-the-middle attack on the 6-round Feistel construction, our attack is mainly divided into three phases: the online query phase, the offline computation phase and the recovery of keys. And the differential characteristic of the 7-round Feistel construction is shown as below.

3.1 Online Query Phase

Firstly, we choose two plaintext sets in the form of $\{v_0||0, v_0||1, ..., v_0||2^{n/2}-1\}$ and $\{v_0 \oplus X||0, v_0 \oplus X||1, ..., v_0 \oplus X||2^{n/2}-1\}$, where n represents the length of one block and X contains b active bits. After querying these plaintexts, 2^n pairs of ciphertexts with truncated differential X' and Y (as shown in Fig. 3) are obtained. The truncated differences X' and Y satisfy the following Proposition 2.

Proposition 2 *Let* $\{v_0||0, v_0||1, ..., v_0||2^{n/2}-1\}$ *and* $\{v_0 \oplus X||0, v_0 \oplus X||1, ..., v_0 \oplus X||2^{n/2}-1\}$ *be the input plaintexts of the 7-round Feistel construction. Let us choose* $2^{n/2}$ *choices of* X' $(X' \neq X)$*. When* X' *is fixed, the difference* Y *also has* $2^{n/2}$ *possible values.*

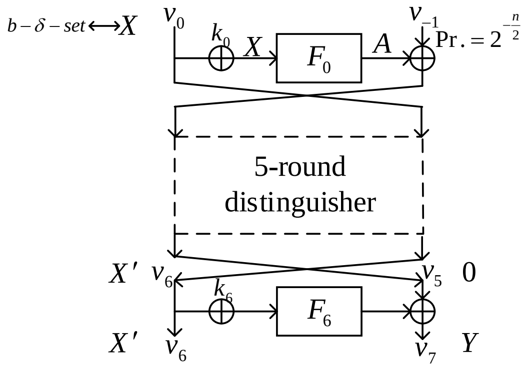

Fig. 3: The differential characteristic of the 7-round Feistel construction. A is the output difference of F_0, and Y is the difference at v_7.

Proof Since $\{v_0||0, v_0||1, ..., v_0||2^{n/2}-1\}$ and $\{v_0 \oplus X||0, v_0 \oplus X||1, ..., v_0 \oplus X||2^{n/2}-1\}$ can construct 2^n pairs, Δv_{-1} has $2^{n/2}$ values. And the posibility of $\Delta v_{-1} = F_0(v_0 \oplus k_0) \oplus F_0(v_0 \oplus k_0 \oplus X)$ is $2^{-\frac{n}{2}}$. Therefore, there are $2^{n/2}$ pairs of plaintexts (m, m') that satisfy the input difference of 5-round distinguisher, i.e., (v_1, v_1') $(v_1 \oplus v'_1 = 0)$ corresponding to (m, m') has $2^{n/2}$ possible values. $v_2 = F_1(v_1 \oplus k_1) \oplus v_0$ also has $2^{n/2}$ possible values. Then, $\Delta = F_2(v_2 \oplus k_2) \oplus F_2(v_2 \oplus X \oplus k_2)$ has $2^{n/2}$ possible values too. And according to the proof of Lemma 1, for $2^{n/2}$ possible values Δ, there exists one state value (one solution) that satisfies such input-output difference in the three rounds of 5-round distinguisher. On the other hand, the possibility of that a pair of plaintexts (m, m') with a difference $0||X$ to a pair of ciphertexts with difference $0||X'$ is $2^{-n/2}$. So, in the $2^{n/2}$ pairs of plaintexts that satisfy the input difference of 5-round distinguisher, there is one pair that satisfies the differential characteristic of 5-round distinguisher. When X' is fixed, and since Lemma 1 gives that v_3 and v_4 are determined by Δ, $v_6 = v_4 \oplus F_5(v_5 \oplus k_5) = v_4 \oplus F_5(v_3 \oplus F_4(v_4 \oplus k_4) \oplus k_5)$ is also determined by Δ. Due to that Δ has $2^{n/2}$ possible values, v_6 also has $2^{n/2}$ possible values. And $Y = \Delta v_7 = F_6(v_6 \oplus k_6 \oplus X') \oplus F_6(v_6 \oplus k_6) \oplus \Delta v_5$ $(\Delta v_5 = 0$ in 5-round distinguisher), the difference Y also has $2^{n/2}$ possible values.

Therefore, the difference between the collected 2^n pairs of ciphertexts will have 2^n possible values. Since X has b active bits, it needs to be queried $2^{n/2+1+b}$ times in total. The time complexity of the entire online query phase is $2^{n/2}$.

3.2 Offline Computation Phase

After getting the 2^n pairs of plaintexts and its corresponding ciphertexts, we need to calculate the difference sequences $\Delta v_5 - sequences$ of each pair with different $X(X \in \{X, X \oplus 1, X \oplus 2, \cdots, X \oplus 2^b - 1\})$. Suppose there is a pair of plaintext $m_1 = (v_0, v_{-1})$ and $m_2 = (v_0 \oplus X, v'_{-1})$, and its corresponding ciphertexts $c_1 = (v_6, v_7)$, $c_2 = (v_6 \oplus X', v_7 \oplus Y)$. The difference sequences $\Delta v_5 - sequences$ consist of the differences Δv_5 between $\{m_2^1 = (v_0 \oplus X \oplus 1, v'_{-1})$, $m_2^2 = (v_0 \oplus X \oplus 2, v'_{-1})$,..., $m_2^{2^b-1} = (v_0 \oplus X \oplus 2^b - 1, v'_{-1})\}$ and $m_2 = (v_0 \oplus X, v'_{-1})$, respectively. It is also the differences Δv_5 between $\{m_1^1 = (v_0 \oplus 1, v_{-1})$, $m_1^2 = (v_0 \oplus 2, v_{-1})$, ..., $m_1^{2^b-1} = (v_0 \oplus 2^b - 1, v_{-1})\}$ and $m_1 = (v_0, v_{-1})$, respectively, i.e.,

$$\Delta v_5 - sequences = \Delta v_5^{m_1 \leftrightarrow m_1^1} || \Delta v_5^{m_1 \leftrightarrow m_1^2} || \cdots || \Delta v_5^{m_1 \leftrightarrow m_1^{2^b-1}}, \tag{1}$$

where $\Delta v_5^{m_1 \leftrightarrow m_1^j} (j \in \{1, 2, \cdots, 2^b - 1\})$ is the difference between m_1^j and m_1 at v_5.

$$\Delta v_5^{m_1 \leftrightarrow m_1^j} = F_6(v_6 \oplus k_6) \oplus F_6(v_6^j \oplus k_6) \oplus v_7 \oplus v_7^j, \tag{2}$$

where k_6 is the key of the round function F_6, $c_1^j = (v_6^j, v_7^j)$ is the ciphertext of m_1^j.

However, the key k_6 is not given. To guess k_6, the differences Δv_5 of all pairs are set to 0. Then, we can get the input F_6^I of the function F_6 by Eq. 3.

$$F_6(F_6^I) \oplus F_6(F_6^I \oplus X') = \Delta v_5 \oplus Y = Y. \tag{3}$$

The key k_6 can be obtained by Eq. 4.

$$k_6 = F_6^I \oplus v_6. \tag{4}$$

In order to facilitate subsequent computation and reduce time complexity, we need to encode these information into the quantum superposition state and compute on the state directly. Let X' be fixed. The computation process is shown as below.

Step 1. Prepare the initial state $|0\rangle^{\otimes n/2} |0\rangle^{\otimes n/2} |0\rangle^{\otimes 2^b n}$.

Step 2. Apply Hadamard transforms $H^{\otimes n/2}$ on the first register to obtain the quantum superposition

$$\sum_{x \in \{0,1\}^{n/2}} \frac{1}{\sqrt{2^{n/2}}} |x\rangle |0\rangle^{\otimes n/2} |0\rangle^{\otimes 2^b n}. \tag{5}$$

Step 3. A quantum query to the function f maps this to the state

$$\sum_{x \in \{0,1\}^{n/2}} \frac{1}{\sqrt{2^{n/2}}} |x\rangle |f(x)\rangle |0\rangle^{\otimes 2^b n} = \sum \frac{1}{\sqrt{2^{n/2}}} \left| F_6^I \right\rangle |Y\rangle |0\rangle^{\otimes 2^b n}, \tag{6}$$

where the function f is based on Eq. 3, i.e.,

$$f(x) = F_6(x) \oplus F_6(x \oplus X') = Y. \tag{7}$$

Step 4. Encode the ciphertexts into the state accroding to different Y values, i.e.

$$\sum \frac{1}{\sqrt{2^{n/2}}} \left| F_6^I \right\rangle |Y\rangle |v_6\rangle |v_7\rangle \left| v_6^1 \right\rangle \left| v_7^1 \right\rangle \cdots \left| v_6^{2^b-1} \right\rangle \left| v_7^{2^b-1} \right\rangle. \tag{8}$$

Step 5. Perform XOR operation on $\left|F_6^I\right\rangle$ and $\left|v_6\right\rangle$, save results on the first $n/2$ qubits, i.e.,

$$\begin{aligned}&\sum \frac{1}{\sqrt{2^{n/2}}}\left|F_6^I \oplus v_6\right\rangle|Y\rangle\left|v_6\right\rangle\left|v_7\right\rangle\left|v_6^1\right\rangle\left|v_7^1\right\rangle \cdots\left|v_6^{2^b-1}\right\rangle\left|v_7^{2^b-1}\right\rangle \\ &=\sum \frac{1}{\sqrt{2^{n/2}}}\left|k_6\right\rangle|Y\rangle\left|v_6\right\rangle\left|v_7\right\rangle\left|v_6^1\right\rangle\left|v_7^1\right\rangle \cdots\left|v_6^{2^b-1}\right\rangle\left|v_7^{2^b-1}\right\rangle .\end{aligned} \tag{9}$$

Step 6. According to Eq. 2, all $\Delta v_5^{m_1 \leftrightarrow m_1^j}$ can be otained. And discard the extra useless qubits to get

$$\left|\Delta v_5-sequences\right\rangle=\sum \frac{1}{\sqrt{2^{n/2}}}|Y\rangle\left|\Delta v_5^{m_1 \leftrightarrow m_1^1}\right\rangle\left|\Delta v_5^{m_1 \leftrightarrow m_1^2}\right\rangle \cdots\left|\Delta v_5^{m_1 \leftrightarrow m_1^{2^b-1}}\right\rangle . \tag{10}$$

The above is the whole process of calculating $\Delta v_5-sequences$, we construct the whole process into a function $g:\{0,1\}^{n/2} \times\{0,1\}^{n/2} \rightarrow\{0,1\}^{(2^b-1)n/2}$, i.e., $g: X' \times Y \rightarrow \Delta v_5-sequences$. Since the process of constructing the function g does not have a search and comparison process, and the time to construct a quantum circuit is generally not considered in quantum cryptoanalysis, the time complexity is only $O(1)$ with $O(n)$ qubits.

On the other hand, in addition to computing $\Delta v_5-sequences$ by collected plaintexts and ciphertexts, it is also necessary to compute $\Delta v_5-sequences$ through the 5-round distinguisher. That is, we need to compute the differences Δv_5 between $\{v_1||v_0 \oplus 1, v_1||v_0 \oplus 2, \cdots, v_1||v_0 \oplus 2^b-1\}$ and $v_1||v_0$. Suppose there is a pair $(v_1||v_0, v_1||v_0 \oplus j)(j \in\{1,2, \cdots, 2^b-1\})$, t_2, t_3 and t_4 are the inputs of F_2, F_3 and F_4 respectively. The values of t_2, t_3 and t_4 can be obtained with diferent X' and Δ by Algorithm 4.

When X' and Δ are fixed, $\Delta v_0=j$, the output difference ΔF_1^O of the first round function F_1 in the 5-round distinguisher is always 0. So, $\Delta v_2=\Delta v_0=j$. In the second round of 5-round distinguisher, due to $F_2^I=t_2$ and $\Delta F_2^I=j$, then, $\Delta F_2^O=F_2(t_2) \oplus F_2(t_2 \oplus j)$. And in the 3rd round of 5-round distinguisher, $\Delta F_3^O=F_3(t_3) \oplus F_3(t_3 \oplus \Delta F_2^O)$ because of $F_3^I=t_3$ and $\Delta F_3^I=\Delta F_2^O$. In the next round, $\Delta v_4=\Delta F_3^O \oplus j$, the output difference of 4th round is $\Delta F_4^O=F_4(t_4) \oplus F_3(t_4 \oplus \Delta F_3^O)$. Finally, we can get the difference of $(v_1||v_0, v_1||v_0 \oplus j)$ at v_5, i.e., $\Delta v_5^j=\Delta F_4^O \oplus \Delta F_2^O$. By repeating this procedure, we can get $\Delta v_5-sequences$ with fixed X' and Δ.

$$\Delta v_5-sequences=\Delta v_5^1||\Delta v_5^2|| \cdots||\Delta v_5^{2^b-1} \tag{11}$$

Moreover, the $\Delta v_5-sequences$ with different X' and Δ should all be computed. The whole process can be defined as a new function $f:\{0,1\}^{n/2} \times\{0,1\}^{n/2} \rightarrow\{0,1\}^{(2^b-1)n/2}$, i.e., $f: X' \times \Delta \rightarrow \Delta v_5-sequences$. Note that, to search $t_{2,3,4}$, we can apply with Grover's algorithm with time complexity of $O(2^{n/4})$, using $O(n)$ qubits. And the parallelized Grover search on all values of $t_{2,3,4}$, can parallelly run $O(2^{n/2})$ independent small processors with $O(n 2^{n/2})$ qubits. Therefore, the time complexity of constructing function f is $O(2^{n/4})$ and it needs $O(n 2^{n/2})$ qubits.

Algorithm 4 The process of searching t_2, t_2 and t_4.

1. for $t_2 = 0, 1, \cdots, 2^{n/2} - 1$ do
   if $\Delta = F_2(t_2) \oplus F_2(t_2 \oplus X)$
      Output t_2; Break;
2. for $t_3 = 0, 1, \cdots, 2^{n/2} - 1$ do
   if $X'' = F_3(t_3) \oplus F_3(t_3 \oplus \Delta)$
      Output t_3; Break;
3. for $t_4 = 0, 1, \cdots, 2^{n/2} - 1$ do
   if $\Delta = F_4(t_4) \oplus F_4(t_4 \oplus X')$
      Output t_4; Break;

For now, we need to find a claw between the functions f and g, which can be implemented by Algorithm 5. The process of Algorithm 5 is as same as Algorithm 3. However, we need to make a little change about the domains of f and g, to meet the requirements of the input in Algorithm 3. Due to $X' \times \Delta$, $X' \times Y \in \{0,1\}^n$, and to ensure $\{X' \times \Delta\} \cap \{X' \times Y\} = \emptyset$, we need to insert 0 or 1 at the front of the bit string, i.e., to get the sets $J_0 = \{0|X' \times \Delta\}$ and $J_1 = \{1|X' \times Y\}$. But, the inserted bit would not participate in the calculation of functions f and g.

Correctness proof of Algorithm 5: Firstly, if there is no desired pair, then Algorithm 5 actually does nothing. Thus, we cannot find the desired pair after the measurement and will reject.

On the other hand, if there is a pair, we would show that we can find it using Lemma 2 (proved by Ambainis [20]). Suppose $(0|j_0, 1|j_1) \in J_0 \times J_1$ is the desired pair. We define a 3-dimensional subspace

$$\tilde{H}_0 = \{|\psi_{j,l}\rangle : j = 0, \ldots, k; l = 0, 1; (j,l) \neq (k,1)\} \tag{12}$$

where $|\psi_{j,l}\rangle$ is the superposition of states $\{|S_0, \{f(j_0)\}, y_0\rangle$: $|S| = r_0, y_0 \in J_0 - S_0$, $j = |S_0 \cap \{0|j_0\}|\}$ (with $l = 1$ if $y_0 = 0|j_0$, otherwise $l = 0$). W_0 (the operator of **Quantum Walk** on S_0 in J_0) has 3 eigenvalues when restricted on $\tilde{H}_0$. One of them is 1, whose corresponding eigenvector is $|\psi_{start,0}\rangle = \frac{1}{\sqrt{\binom{m}{r_0}(m-r_0)}} \sum_{S_0 \subseteq J_0, |S_0|=r_0, y_0 \in J_0 - S_0} |S_0, \{f(j_0)\}, y_0\rangle$. And the other two eigenvalues are $e^{\pm i\theta_0}$ ($\theta_0 = (2 + o(1))/\sqrt{r_0}$). Therefore, $W_0^{t_0}$ has 3 eigenvalues: 1 (corresponding eigenvector $|\psi_{start,0}\rangle$) and $e^{\pm i\theta'_0}$ ($\theta'_0 = \frac{\pi}{2} + o(1)$).

Similarly, $\tilde{H}_1$ is defined, as well as W_1, $|\psi_{start,1}\rangle$ and θ_1. So, $W_1^{t_1}$ has 3 eigenvalues: 1 (corresponding eigenvector $|\psi_{start,1}\rangle$) and $e^{\pm i\theta'_1}$ ($\theta'_1 = \frac{\pi}{4} + o(1)$). The Step 2(b) restricted on $\tilde{H}_0 \otimes \tilde{H}_1$ is the operation $W = (I_0 \otimes W_1)(I_1 \otimes W_0)$. So, W has 9 eigenvalues $\{e^{i(b_0\theta'_0 + b_1\theta'_1)} : b_0, b_1 \in \{-1, 0, 1\}\}$. One of the eigenvalue is 1 and its corresponding eigenvector is $|\psi_{start,0}\rangle \otimes |\psi_{start,1}\rangle$, which is exactly the $|\psi_{start}\rangle$ in Algorithm 5. And all the other 8 eigenvalues are in the form of $e^{i\theta}$ ($\theta \in [\pi/4 - o(1), 2\pi - \pi/4 + o(1)]$). Finally, we can find $\alpha = \langle\psi_{start}|\psi_{good}\rangle = \sqrt{\Pr_{|S_0|=r_0,|S_1|=r_1}[(0|j_0, 1|j_1) \in S_0 \times S_1]} = O(\sqrt{\frac{r_0 r_1}{mn}})$.

Algorithm 5 Quantum claw finding between the functions f and g.

Input: $J_0 = \{0|X' \times \Delta\}$, $J_1 = \{1|X' \times Y\}$, $|J_0| = |J_1| = n$. There exists at most one pair of $f(j_0) = g(j_1)$ with $0|j_0 \in J_0$, $1|j_1 \in J_1$ and $0|j_1 \neq 1|j_2$.
Output: Output the pair $(0|j_0, 1|j_1)$, otherwise reject.

1. Create the initial state $|\psi_{start}\rangle = \frac{1}{\sqrt{T}} \sum_{S_b \subseteq J_b, |S_b| = r_b, y_b \in J_b - S_b} |S_0, \{f(j_0)\}, y_0, S_1, \{g(j_1)\}, y_1\rangle$, where $T = \binom{m}{r_0}\binom{n}{r_1}(m - r_0)(n - r_1)$ and $b = 0, 1$.
2. Repeat following operations $O(\sqrt{\frac{mn}{r_0 r_1}})$ times
   (a) Check whether the pair $(0|j_0, 1|j_1)$ is in $S_0 \times S_1$. If it exits, do the following phase flip:$|S_0, \{f(j_0)\}, y_0, S_1, \{g(j_1)\}, y_1\rangle \to - |S_0, \{f(j_0)\}, y_0, S_1, \{g(j_1)\}, y_1\rangle$.
   (b) Do **Quantum Walk** on S_0 in J_0 for $t_0 = \lceil \frac{\pi}{4}\sqrt{r_0} \rceil$ times.
   Do **Quantum Walk** on S_1 in J_1 for $t_1 = \lceil \frac{\pi}{8}\sqrt{r_1} \rceil$ times.
3. Measure the resulting state and give the corresponding answer.

So, the number of iterations in Step 2 is $\frac{1}{\alpha} = O(\sqrt{\frac{mn}{r_0 r_1}})$, and the correctness holds by Lemma 2.

Lemma 2 *[20] Let $\mathcal{H}$ be a finite-dimensional Hilbert space and $|\psi_1\rangle$, ..., $|\psi_m\rangle$ be an orthonormal basis for $\mathcal{H}$. Let $|\psi_{good}\rangle$, $|\psi_{start}\rangle$ be two states in $\mathcal{H}$ which are superpositions of $|\psi_1\rangle$, ..., $|\psi_m\rangle$ with real amplitudes and $\langle\psi_{good} \mid \psi_{start}\rangle = \alpha$. Let U_1, U_2 be unitary transformations on $\mathcal{H}$ with the following properties:*

1. *U_1 is the transformation that flips the phase on $|\psi_{good}\rangle$ ($U_1 |\psi_{good}\rangle = -|\psi_{good}\rangle$) and leaves any state orthogonal to $|\psi_{good}\rangle$ unchanged.*
2. *U_2 is a transformation which is described by a real-valued $m \times m$ matrix in the basis $|\psi_1\rangle$, ..., $|\psi_m\rangle$. Moreover, $U_2 |\psi_{start}\rangle = |\psi_{start}\rangle$, and if $|\psi\rangle$ is an eigenvector of U_2 perpendicular to $|\psi_{start}\rangle$, then $U_2 |\psi\rangle = e^{i\theta} |\psi\rangle$ for $\theta \in [\epsilon, 2\pi - \epsilon]$, $\theta \neq \pi$ (where ϵ is a constant, $\epsilon > 0$).*

Then, there exists $t = O(\frac{1}{\alpha})$ such that $\left|\left\langle \psi_{good} | (U_2 U_1)^t | \psi_{start} \right\rangle\right| = \Omega(1)$.

3.3 Recovery of k_0 and k_6

After obtaining the claw (X', Y, Δ), we can recover the keys k_0 and k_6. The key k_6 can be recovered by Eq. 3 and 4 with known X', Y and v_6(v_6 can be found according to the known X' and Y.). Similarly, to recover k_0, we need to find the plaintext pair $m_1 = (v_0, v_{-1})$ and $m_2 = (v_0 \oplus X, v'_{-1})$ corresponding to the known X' and Y. Then, the input F_0^I of the function F_0 can be obtained when $m_1 = (v_0, v_{-1})$ as the input of 7-round Feistel construction.

$$F_0(F_0^I) \oplus F_0(F_0^I \oplus X) = v_{-1} \oplus v'_{-1} \tag{13}$$

And the key k_0 can be recovered as below,

$$k_0 = F_0^I \oplus v_0. \tag{14}$$

However, among these $2^{n/2}$ candidates for k_0 (k_6), only one candidate k_0 (k_6) is correct while the remaining are false positives. To find the correct keys, we need to make sure that the set $\{\Delta v_5 - sequences\}$ contains all the possible stored sequences of differences. When b is too small, the recovery will wrongly yield to valid key candidates k_0 and k_6. On the other hand, When b is too large, the construction of $\{\Delta v_5 - sequences\}$ will require more queries. Therefore, we need to find an optimal value of b. One recovery yields a false positive with probability $2^{n/2}/2^{n2^b/2} = 2^{n(1-2^b)/2}$ as there are $2^{n/2}$ valid sequences of 2^b elements among the $2^{n2^b/2}$ theoretically possible ones. Thus, we want $2^{n(2-2^b)/2} \ll 1$ that only the correct key k_0 (k_6) results in a stored element, and $b \geq 2$.

4 Analysis and Comparision

The entire our QMITM attack on 7-round Feistel construction consists of 3 phases, the online query phase, the offline computation phase and the key recovery phase. But the resources are mainly consumed by the first two phases. During the online query phase, our proposed attack only needs $2^{n/2+1+b}$ queries, and then store these plaintexts and ciphertexts. So, the time and memory complexities of this phase are both $O(2^{n/2})$.

In the offline computation phase, the time complexities of constructing functions f and g are $O(2^{n/4})$ and $O(1)$, in addition to $O(n2^{n/2})$ and $O(n)$ qubits, respectively. Then, the complexity of searching a claw between f and g (i.e. Algorithm 5) is similar to Algorithm 3, which is $O(2^{2n/3})$ with $O(n)$ qubits. In summary, the time and memory complexities of our attack are $O(2^{2n/3})$ and $2^{n/2}$, respectively. And $O(n2^{n/2})$ qubits would be consumed.

On the other hand, the detailed comparison with other classic meet-in-the-middle attacks [22,23] and the quantum attacks [11,15] are shown in Table 2. In the classical setting, although Guo *et al.* [22] did not specify the complexity of the attack on 7-round Feistel construction with key size of n, they pointed out that the distinguisher for $4 + s + 1$ rounds uses the same strategy as the one on $4 + s$ rounds ($s \geq 2$), and the time complexity is $\mathcal{T}_7 = 2^{n/2}\times\ \mathcal{T}_6$, where $\mathcal{T}_6 = 2^{3n/4}$ is the time complexity for 6-round Feistel. And the memory complexity is similar to time complexity. Then, Zhao *et al.*'s classic meet-in-the-middle attack on 7-round Feistel construction is extended from Guo's attack, which use pairs sieve procedure to reduce complexity [23]. Its time complexity is $O(2^n)$, and it needs to store $2^{n/4}$ structures of $2^{n/2}$ plaintexts and ciphertexts.

For quantum attacks, although the quantum slide attacks [11, 12] have achieved non-travil results, they currently only attacks at most 4-round Feistel. Besides, Hosoyamada *et al.*'s work [17] only gives the attack on 6-round one. And Xie [16] provides a new and inspirational approach to implement a new quantum 3-round Feistel distinguisher by Bernstein–Vazirani algorithm, whose time complexity is a little higher than the distinguisher based on Simon's algorithm. Therefore, we could not compare with them. Then, Dong *et al.*'s attack [13] is under Q2 model, which analyzes the relationship between the last 4 round keys and ciphertexts based on quantum 3-round distinguisher, and uses Grover's algorithm to search correct last 4 round keys. Its time complexity is $O(2^n)$. Since the attack directly performs quantum computations on $O(n^2)$ qubits, no classic memory storage is required. Furthermore, Ito *et al.* gives a new quantum 4-round Feistel distinguisher. So, the attack only needs to search 3 round keys by Grover's algorithm, whose time complexity is $O(2^{3n/4})$. The other complexities are as same as Dong *et al.*'s attack.

It is obvious that our time complexity is the lowest. And our attack belongs to Q1 model, which is more practical than Q2 model. Since our attack is in Q1 model, we need classic memory to store queried data and a large number of qubits to perform repeated computation. Nevertheless, we think it is deserved, especially when the adversary is not permitted to access the quantum cryptographic oracle.

Table 2: Comparision with classic and quantum attacks on 7-round Feistel construction.

	Setting	Time	Memory	Qubits
Guo *et al.*'s attack [22]	Classic	$2^{5n/4}$	$2^{5n/4}$	-
Zhao *et al.*'s attack [23]	Classic	2^n	$2^{3n/4}$	-
Dong *et al.*'s attack [13]	Q2	2^n	-	n^2
Ito *et al.*'s attack [15]	Q2	$2^{3n/4}$	-	n^2
Our attack	Q1	$2^{2n/3}$	$2^{n/2}$	$n2^{n/2}$

5 Conclusion

To reduce the time complexity of classic and quantum attacks on 7-round Feistel construction, we propose a quantum meet-in-the-middle attack combining the quantum claw finding algorithm and the 5-round distinguisher in Q1 model. Our proposed attack reduce the time complexity from $O(2^{3n/4})$ to $O(2^{2n/3})$ comparing with the best quantum attack [15], and less than classic attacks. Although our attack consumes more qubits, it belongs to the Q1 model, which is more practical and deserved.

Furthermore, we hope to carry out quantum meet-in-the-middle attacks on more multi-rounds Feistel constructions. Because there are not only 5-round distinguisher, but also 7-round distinguisher, 8-round distinguisher, etc. Combining these distinguishers with quantum claw finding algorithms, or even other quantum algorithms may achieve good attack results.

Acknowledgements This work was supported by the 13th Five-Year Plan National Cryptography Development Fund (MMJJ20180217).

References

1. Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.
2. Lov K Grover. Quantum computers can search arbitrarily large databases by a single query. *Physical review letters*, 79(23):4709, 1997.
3. Hidenori Kuwakado and Masakatu Morii. Quantum distinguisher between the 3-round feistel cipher and the random permutation. In *2010 IEEE International Symposium on Information Theory*, pages 2682–2685. IEEE, 2010.
4. Daniel R Simon. On the power of quantum computation. *SIAM journal on computing*, 26(5):1474–1483, 1997.
5. Hidenori Kuwakado and Masakatu Morii. Security on the quantum-type even-mansour cipher. In *2012 International Symposium on Information Theory and its Applications*, pages 312–316. IEEE, 2012.
6. Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia. Breaking symmetric cryptosystems using quantum period finding. In *Crypto 2016-36th Annual International Cryptology Conference*, pages 207–237. Springer, 2016.
7. Thomas Santoli and Christian Schaffner. Using simon's algorithm to attack symmetric-key cryptographic primitives. *arXiv preprint arXiv:1603.07856*, 2016.
8. Tairong Shi, Chenhui Jin, and Jie Guan. Collision attacks against aez-prf for authenticated encryption aez. *China Communications*, 15(2):46–53, 2018.
9. Yinsong Xu, Wenjie Liu, and Wenbin Yu. Quantum forgery attacks on copa, aes-copa and marble authenticated encryption algorithms. *Quantum Information Processing*, 20(4):1–21, 2021.
10. Gregor Leander and Alexander May. Grover meets simon–quantumly attacking the fx-construction. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 161–178. Springer, 2017.
11. Xiaoyang Dong, Bingyou Dong, and Xiaoyun Wang. Quantum attacks on some feistel block ciphers. *Designs, Codes and Cryptography*, 88:1–25, 2020.
12. Xavier Bonnetain, María Naya-Plasencia, and André Schrottenloher. On quantum slide attacks. In Kenneth G. Paterson and Douglas Stebila, editors, *Selected Areas in Cryptography – SAC 2019*, pages 492–519, Cham, 2020. Springer International Publishing.
13. Xiaoyang Dong and Xiaoyun Wang. Quantum key-recovery attack on feistel structures. *Science China Information Sciences*, 61(10):1–7, 2018.
14. Xiaoyang Dong, Zheng Li, and Xiaoyun Wang. Quantum cryptanalysis on some generalized feistel schemes. *Science China Information Sciences*, 62(2):22501, 2019.
15. Gembu Ito, Akinori Hosoyamada, Ryutaroh Matsumoto, Yu Sasaki, and Tetsu Iwata. Quantum chosen-ciphertext attacks against feistel ciphers. In Mitsuru Matsui, editor, *Topics in Cryptology – CT-RSA 2019*, pages 391–411, Cham, 2019. Springer International Publishing.
16. Huiqin Xie and Li Yang. Using bernstein–vazirani algorithm to attack block ciphers. *Designs, Codes and Cryptography*, 87(5):1161–1182, 2019.
17. Akinori Hosoyamada and Yu Sasaki. Quantum demiric-selçuk meet-in-the-middle attacks: applications to 6-round generic feistel constructions. In *International Conference on Security and Cryptography for Networks*, pages 386–403. Springer, 2018.
18. Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia. Quantum differential and linear cryptanalysis. *IACR Transactions on Symmetric Cryptology*, 2016(1):71–94, 2016.
19. Shengyu Zhang. Promised and distributed quantum search. In Lusheng Wang, editor, *Computing and Combinatorics*, pages 430–439, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
20. Andris Ambainis. Quantum walk algorithm for element distinctness. *SIAM Journal on Computing*, 37(1):210–239, 2007.
21. Harry Buhrman, Christoph Dürr, Mark Heiligman, Peter Høyer, Frédéric Magniez, Miklos Santha, and Ronald de Wolf. Quantum algorithms for element distinctness. *SIAM Journal on Computing*, 34(6):1324–1330, 2005.
22. Jian Guo, Jérémy Jean, Ivica Nikolić, and Yu Sasaki. Extended meet-in-the-middle attacks on some feistel constructions. *Designs, Codes and Cryptography*, 80(3):587–618, 2016.

23. Shibin Zhao, Xiaohan Duan, Yuanhao Deng, Zhiniang Peng, and Junhu Zhu. Improved meet-in-the-middle attacks on generic feistel constructions. *IEEE Access*, 7:34416–34424, 2019.