Proceedings of the 2023 International Conference on Mathematical Physics and Computational Simulation
DOI: 10.54254/2753-8818/11/20230402

The development of generic key recovery attack on Feistel structure

Zhouquan Jin

School of Cyber Science and Technology, Shandong University, Qingdao 266237, Shandong, China

student012@126.com

Abstract. Feistel structure was firstly proposed in 1973, and because its structure has a great avalanche effect and similar encryption and decryption, it was used in many encryption schemes, like DES, AES, CAST. According to the ambiguity of the intermediate state, Feistel structures are separately named as Feistel-1, Feistel-2 and Feistel-3.Even though some of efficient analysis were proposed to attack the Feistel structure such as differential cryptanalysis and linear attack, these attacks are only applicable to a given Feistel structure and cannot have a general analysis of all Feistel structures. To attack the general Feistel structure, splice and cut, key linearization, and meet-in-the-middle attack have been used to propose the general key recovery attack on various Feistel architectures. This paper summarizes these results and proposes the research direction of the MITM attack of the Feistel structure, especially for the generic key recovery on different round functions and combination with modern means, like the application of Simon algorithm, which can build 3-round distinguisher on the Feistel structure.

Keywords: cryptography, block cipher, meet-in-the-middle attack, generic key recovery attack.

1. Introduction

In 1973, while working on the architecture of Lucifer, Feistel suggested the now-famous Feistel structure [1]. Feistel structure is a block cipher structure, with n-bits input and n-bits output and usually $n/2$-bits permutations in a single round, which is called "branch". After each round, the Feistel structure switches its left branch and right branch to make each branch permutable. Due to the reason that Feistel structure has the similar encryption and decryption, and it has a well performance on the avalanche effect, Famous ciphers use Feistel structure, such as DES, Camellia and CAST.

Based on the clarity of the given encryption process, Feistel structure can be dived into 3 types, Feistel-1, Feistel-2 and Feistel-3, and because Feistel structure is widely used, a lot of ways to challenge its safety is produced. Differential analysis, linear analysis are commonly used in attacking the Feistel structure. However, these attacks depends on the specific permutations the Feistel structure has. Then, an approach known as generic key recovery is presented to crack the Feistel structure. The generic key recovery attack does not focus on the specific permutations the Feistel structure has, so some of the pre-computations can be done before attacking the specific encryption schemes. In recent years, several generic key recovery attacks have been proposed to attacking different types of the Feistel structures mentioned above. This paper is going to summarize the development of the Feistel-1 and Feistel-2 structures (especially when the length of key equals the length of the plaintext/ciphertext, denoted as

DOI: 10.54254/2753-8818/11/20230402

$k = n$).

2. The development of the generic key recovery attack on Feistel-1

Feistel-1 structure refers to the Feistel structure where the F-functions are randomly keyed.

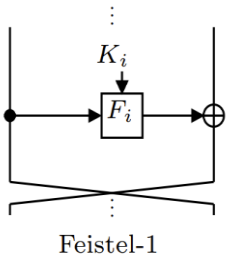

Figure 1. Feistel-1 structure [3].

2.1. Notations

Ψ^k:k-round Feistel structure
L_i, R_i: the Feistel-1 structure's 4-cycle input
Y_i: the output of the 2-round Feistel-1 structure
S_i, T_i: the output of the 4-round Feistel-1 structure
Z_i:the output of the 5-round Feistel-1 structure
m:the total numbers of the known input o
i:the i-th input, $1 \leq i \leq m$

2.2. A 5-round distinguisher used on Feistel-1 structure

In 2004, Patarin [2] proposed that based on the adaptive chosen plaintext attack (CPA-1), a 5-round distinguisher can be constructed when choosing $2^{n/2}$ plaintexts and with $O(2^{n/2})$ computations, and based on the known plaintext attacks (KPA), another 5-round distinguishers can be constructed with choosing $2^{3n/4}$ plaintexts and with $O(2^{3n/4})$ computations. Denote the 5-round Feistel structure as Ψ^5.

2.2.1. The CPA-1 distinguisher on Ψ^5. Conclusion: Fix that the $L_i = constant,\ \forall i, 1 \leq i \leq m$,then count the number N of (i,j) such that $S_i = S_j$ and $L_i \oplus T_i = L_i \oplus T_j$, the number of N is about double for the Ψ^5 compared with the random permutation, which builds the distinguisher.

Proof:

If $S_i = S_j$

$$L_i \oplus T_i = L_j \oplus T_j \Leftrightarrow L_i \oplus Z_i = L_j \oplus Z_j \Leftrightarrow f_1(R_1) \oplus f_3(Y_i) = f_1(R_1) \oplus f_3(Y_j)$$

$$\Leftrightarrow f_3\left(R_1 \oplus f_2\left(L_i \oplus f_1(R_1)\right)\right) = f_3\left(R_1 \oplus f_2\left(L_j \oplus f_1(R_1)\right)\right)$$

It means that $f_2(L_i \oplus f_1(R_1)) = f_2(L_j \oplus f_1(R_1))$, or if they are distinct, they have the distinct values, they would have the same images by f_3. Thus, the count number N is as twice much as when the plaintexts go through the Ψ^5 than the truly random permutations.

2.2.2. The KPA on the Ψ^5. Patarin [2] hypothesised that the CPA attack could become a KPA: instead of counting the number N such that $S_i = S_j, L_i \oplus T_i = L_j \oplus T_j$, the number N' will be incremented by one such that $(i,j), R_i = R_j, S_i = S_j\ and\ L_i \oplus T_i = L_j \oplus T_j$, and to make KPA efficient, the number of the chosen plaintexts should go up to $2^{3n/4}$ and with $O(2^{3n/4})$ computations. With this finding, Patarin [2] could reduce attackers' abilities but get the same result.

DOI: 10.54254/2753-8818/11/20230402

2.3. Attacking the 3-round Feistel-1 structure with a general key recovery strategy

Patarin [2] constructed a 5-round generic distinguisher. The distinguisher is too speculative for the generic key recovery attack. In 2013, Isobe and Shibutani [3] used All Subkeys Recovery (ASR) to attack 3-round Feistel-1 structure with broad key recovery. ASR attacks expand MITM attacks. ASR guesses subkeys instead of the master key using complex key scheduling methods.

2.3.1. The description of the all subkeys recovery attack. ASR is a two-way meet-in-the-middle attack. To be more specific, firstly, the attacker determines a t-bit matching state X, $X \in \{0,1\}^t$, and through the forward direction, the set of the subkeys, denoted as $\mathcal{K}_{(1)}$, and the function denoted as $\mathcal{F}_{(1)}$, so the matching state X is derived from $X = \mathcal{F}_{(1)}(P, \mathcal{K}_{(1)})$. Similarly, the state can also be computed from the back direction, denote the set of these subkeys as $\mathcal{F}_{(2)}$, and the matching state can be derived by $X = \mathcal{F}_{(2)}^{-1}(C, \mathcal{K}_{(2)})$, C is the ciphertext. Finally, the remaining subkeys, which are not included in computing the match state, are gathered and denoted as $\mathcal{K}_{(3)}$. Then it can be calculated that $|\mathcal{K}_{(1)}| + |\mathcal{K}_{(2)}| + |\mathcal{K}_{(3)}| = r \cdot n/2$, where r is the number of rounds, and n is the length of the keys. After the process, the attacker has $2^{r \cdot n/2 - t}$ key candidates left. Then, the attacker can brute force all candidate keys, and the total computations, denoted as C_{comp} using N plaintext/ciphertext pairs is estimated as:

$$C_{comp} = max\left(2^{|\mathcal{K}_{(1)}|}, 2^{|\mathcal{K}_{(2)}|}\right) \times N + 2^{r \cdot n/2 - N \cdot t} \quad (1)$$

The number of required plaintext/ciphertexts pairs is: $max(N, \lceil (r \cdot n/2 - N \cdot t)/n \rceil)$

The required memory is about: $min\left(2^{|\mathcal{K}_{(1)}|}, 2^{|\mathcal{K}_{(2)}|}\right) \cdot N$, and it is easy to see that cost of the ASR is less than the brute force 2^n.

2.3.2. The ASR used in the Feistel-1 structure. On the premise that the ciphertext is of the same length as the keys, denoted as $k = n$, the attacker can attack the 3-round Feistel-1 structure, the $|\mathcal{K}_{(1)}| = |\mathcal{K}_{(2)}| = 1 \cdot n/2$ bits, based on the equation computing the C_{comp}, the number of the computations need $2^{n/2+1}(2^{n/2} \times 2 + 2^{n/2})$ and about $(2 \times 2^{n/2})$ blocks memory, which is absolutely less than the computations used in the brute force attack 2^n, which means that the ASR on the 3-round Feistel-1 structure is efficient.

2.4. The improved ASR on the Feistel-1 structure

Isobe [4] improved the ASR-based generic key recovery attack on the 4-round Feistel-1 structure a year later. Isobe et al [4] controlled the value of the plaintexts, the left input branch, $L_1 = CON$, and the $L_2 = R_1 \oplus K_1'$, where $K_1' = F_1(K_1 \oplus CON)$ and because K_1'depends on the K_1, so it can be regarded as a new n/2-bit subkey K_1' is linearly inserted in the first round, and it is called the key linearization.

The splice and cut [5] technique allows us to divide the K_1' into $\mathcal{F}_{(1)}$ and $\mathcal{F}_{(2)}$ respectively. Therefore, to attack the 4-round Feistel-1 structure, guessing the K_1 as K_1' as the equivalence subkeys, and then $|\mathcal{K}_{(1)}| = |\mathcal{K}_{(2)}| = 3n/4$-bit, so $C_{comp} = 2^{3n/4+2}$ computations and $(3 \times 2^{3n/4})$ block memory is needed, in which the total complexity is much less than the brute force attacks (2^n), and this is the best result of the generic key recovery attack on the 4-round Feistel-1 structure.

3. The development of the generic key recovery attack on Feistel-2

The Feistel-2 structure allows the subkeys XORed before the F-function.

DOI: 10.54254/2753-8818/11/20230402

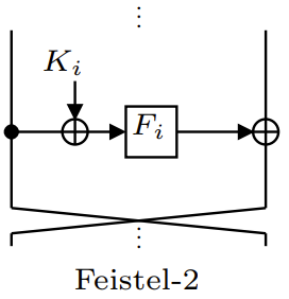

Figure 2. Feistel-2 structure [3].

It is easy to see that the Feistel-2 structure is not easier than the Feistel-1 structure, so the distinguisher proposed by Patarin [3] can be also used in the Feistel-2 structure.

3.1. Key recovery attack employing function reduction approach on a 5-round Feistel-2 structure

3.1.1. Function reduction. Denote that the L_{r+1} and R_{r+1} are the output of the r-round Feistel-2 structure, and they are represented by the function $\mathscr{F}_{\mathcal{L},r}, \mathscr{F}_{\mathcal{R},r}$ as $L_{r+1} = \mathscr{F}_{\mathcal{L},r}(\mathcal{K}_L, L_1|R_1)$ and $R_{r+1} = \mathscr{F}_{R,r}(\mathcal{K}_R, L_1|R_1)$, and the $\mathcal{K}_L, \mathcal{K}_R$ are sets of subkeys, so $|\mathcal{K}_L|$ and $|\mathcal{K}_R|$ can be computed as $|\mathcal{K}_L| = n/2 \times r$, $|\mathcal{K}_R| = n/2 \times (r-1)$, the Function Reduction Technique shows that when L_1 is fixed, $\mathcal{K}_L, \mathcal{K}_R$ contain at most $(n/2 \cdot r)$ and $(n/2 \cdot (r-2))$ subkey bits.

3.1.2. The use of the function reduction in the 5-round Feistel-2 structure. To implement the Function Reduction Method in a Feistel-2 framework with five rounds, Isobe *et al.* [4] fixes the $L_1 = CON_{\{1\}} and R_6 = CON_2$ as the arbitrary $(n/2)$-bit constants, and make the R_4 as the matching state used in the MITM attack, and then compute the equivalence keys $K_2' = K_2 \oplus K_1'$ and $K_1' = F(K_1 \oplus CON_1)$. In the backward computation, the equivalence keys are $K_4' = K_2 \oplus K_5'$ and $K_5' = F(K_5 \oplus CON_2)$. In addition, $2^{n/4}$ chosen plaintexts are used to make sure that $L_1 = CON_1$ and $R_6 = CON_2$. By using the ASR attack, K_2' and K_4' can be estimated by the complexity $C_{comp} = max(2^{n/2}, 2^{n/2}) \times 2$, and the same as K_3 and K_1'', where $K_1'' = K_1' \oplus K_5$. Finally after guessing the K_1 by $2^{n/2}$, all subkeys can be estimated, so the total complexity $2^{n/2+2}$, which is less than the brute force attack 2^n, is efficient.

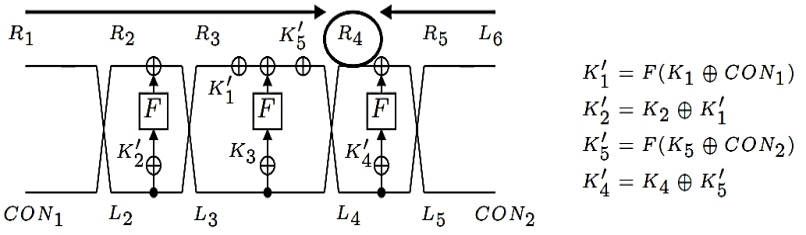

Figure 3. The 5-round Feistel-2 structure attack [3].

3.2. The Feistel-2 6-round generic key recovery attack

In 2016, Guo et al. built a 6-round key recovery attack and a Feistel-2 structure distinguisher. The attackers' rounds can be extended linearly as the key length increases. Guo's recovery attack combines differential analysis with generic key recovery to generate a precomputation table and make choices on its cost and size. Another advantage is that because Guo et al. uses the $b - \delta$-set techniques, the number of the plaintexts can be reduced because each two of the plaintexts can become a pair as the input.

DOI: 10.54254/2753-8818/11/20230402

3.2.1. 5-round distinguisher on the Feistel-2 structure. Guo $et\ al.$ proposed that when the input difference is fixed as $0||X$, while the output difference is fixed as $X'||0$, thus the number of impossible internal state values of the three middle rounds for the plaintexts is restricted to $2^{n/2}$, which means that the possible internal state can be tightened from $2^{3n/2}$ to $2^{n/2}$, where the distinguisher is built.

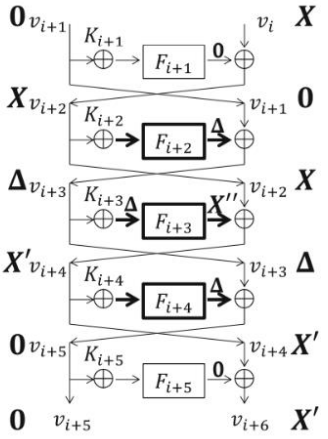

Figure 4. 5-round differential characteristic [6].

3.2.2. The 6-round key recovery attack. The 6-round key recovery assault extends the Feistel-2 structure's 5-round distinguisher. Firstly, Guo$et\ al.$constructed precomputation tables called T_2, T_4, which aims at finding the input corresponding to the differential input and output X, X'., so do the T_3. (To match the fixed internal state). After building the precomputation table, Guo $et\ al.$ ergodic all the differential, and each of the difference, Guo $et\ al.$ ergodics all the b active bits of the $b-\delta$set to compute all the corresponding Δv_5. Then Guo $et\ al.$ requires the oracle with the structure of $2^{n/2+1}$ plaintexts, and it can produce the 2^n pairs that matches the left input difference. To match the right branch difference, it depends on the property, which means to regard the first round of the encryption as the random permutations, so there will be $2^{-n/2}$ possibilities to match the right branch of the difference that the distinguisher needs, so $2^{n/2}$ candidates pairs will be left, and for each of them, to match the precomputation table and find the appropriate candidate subkey for K_0. After the process, Guo $et\ al.$ construct the $b-\delta$ set by modifying the active bits of v_0. By the candidate subkeys K_0, the author decrypts the candidate plaintexts and modify the left branch of the input, so that the value of the v_1 remains unchanged. With the knowledge, Δv_5 can be computed, and compared to the precomputation table T_δ. If matches, then K_0 is a correct subkey with high probability. Through this process, and the tradeoff of the data and time complexity, the ideal data complexity is $2^{3n/4}$ chosen plaintexts, while the time complexity is $2^{3n/4}$ encryptions, which are both less than the brute force attack.

3.3. The Feistel-2 7-round generic key recovery attack

Based on Guo et al.'s distinguisher, Zhao et al. [7] suggested a 7-round generic key recovery attack by using the method called impossible-differential sieve technique, Zhao et al. [7] found that with the same input proposed by Guo et al., after 7-round of the Feistel encryption, only $2^{n/2-1}$ distinct values left. With this finding, Zhao et al. [7] proposed the 7-round generic key recovery attack, and its complexity of data is $3\times 2^{n-2}$ chosen plaintexts, while the complexity of time is $3\times 2^{n-2}$ encryptions.

4. Conclusion

Developed from the basic key recovery attacks against the Feistel-1 and Feistel-2 structures, increasing rounds are being attacked, but one of the limitations is the connection between the distinguisher and the

DOI: 10.54254/2753-8818/11/20230402

real key recovery attack. Since the attacker does not know the internal state of the encryption, they can only collide with a lot of input pairs, which is wasteful. H. Kuwakado and M. Morii used post-quantum computation to identify the 3-round Feistel cypher from the random permutation [8]. Combining quantum and conventional distinguishers is another good idea.

References

[1] Feistel, Horst. "Cryptography and Computer Privacy. " Scientific American 228, no. 5 (1973): 15–23.

[2] Patarin, J. (2004). Security of Random Feistel Schemes with 5 or More Rounds. In: Franklin, M. (eds) Advances in Cryptology – CRYPTO 2004. CRYPTO 2004. Lecture Notes in Computer Science, vol 3152. Springer, Berlin, Heidelberg

[3] Isobe, T., Shibutani, K. (2013). All Subkeys Recovery Attack on Block Ciphers: Extending Meet-in-the-Middle Approach. In: Knudsen, L.R., Wu, H. (eds) Selected Areas in Cryptography. SAC 2012. Lecture Notes in Computer Science, vol 7707. Springer, Berlin, Heidelberg.

[4] Isobe, T., Shibutani, K. (2013). Generic Key Recovery Attack on Feistel Scheme. In: Sako, K., Sarkar, P. (eds) Advances in Cryptology - ASIACRYPT 2013. ASIACRYPT 2013. Lecture Notes in Computer Science, vol 8269. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-42033-7_24

[5] Aoki, K., Guo, J., Matusiewicz, K., Sasaki, Y., Wang, L. (2009). Preimages for Step-Reduced SHA-2. In: Matsui, M. (eds) Advances in Cryptology – ASIACRYPT 2009. ASIACRYPT 2009. Lecture Notes n Computer Science, vol 5912. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-10366-7_34

[6] Guo, J., Jean, J., Nikolić, I., Sasaki, Y. (2014). Meet-in-the-Middle Attacks on Generic Feistel Constructions. In: Sarkar, P., Iwata, T. (eds) Advances in Cryptology – ASIACRYPT 2014. ASIACRYPT 2014. Lecture Notes in Computer Science, vol 8873. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-45611-8_24

[7] S. Zhao, X. Duan, Y. Deng, Z. Peng and J. Zhu, "Improved Meet-in-the-Middle Attacks on Generic Feistel Constructions," in IEEE Access, vol. 7, pp. 34416-34424, 2019, doi: 10.1109/ACCESS.2019.2900765.

[8] H. Kuwakado and M. Morii, "Quantum distinguisher between the 3-round Feistel cipher and the random permutation," 2010 IEEE International Symposium on Information Theory, Austin, TX, USA, 2010, pp. 2682-2685, doi: 10.1109/ISIT.2010.5513654.