IMPOSSIBLE DIFFERENTIAL CRYPTANALYSIS OF FOX

Zhongming Wu, Xuejia Lai, Bo Zhu, and Yiyuan Luo

Department of Computer Science and Engineering,
Shanghai Jiaotong University, Shanghai 200240, P. R. China

ABSTRACT

Block ciphers are the very foundation of computer and information security. FOX, also known as IDEA NXT, is a family of block ciphers published in 2004 and is famous for its provable security to cryptanalysis. In this paper, we apply impossible differential cryptanalysis on FOX cipher. We find a 4-round impossible difference, by using which adversaries can attack 5, 6 and 7-round FOX64 with 2^{71}, 2^{135} and 2^{199} one-round encryptions respectively. Compared to the previous best attack with $2^{109.4}$, $2^{173.4}$ and $2^{237.4}$ full-round encryptions to 5, 6 and 7-round FOX64, the method in this paper is the best attack to FOX cipher. This attack can also be applied to 5-round FOX128 with 2^{135} one-round encryptions.

Index Terms— Security, cryptography, block cipher, FOX, impossible difference

1. INTRODUCTION

FOX, also known as IDEA NXT, is a family of block ciphers designed by Junod and Vaudenay in 2004 [1]. In 2005 it was announced by MediaCrypt under the name IDEA NXT. It is the successor of the IDEA algorithm and it uses an extended Lai-Massey scheme known for its robustness to cryptanalysis. Currently, two versions of FOX cipher has introduced. One is FOX64/k/r and the other is FOX128/k/r, and they have 64-bits and 128-bits block size respectively. Both of them have a variable key length ranging from 8 to 256 bits, the original design suggests these two ciphers should be iterated for 16 rounds.

The round function of FOX adopts Lai-Massey Scheme [2]. The Lai-Massey Scheme is a well-known structure in block cipher design, it has been proven to have a good pseudorandomness property under Luby-rackoff paradigm and decorrelation in hesitance properties.

Despite of the strong security properties of round function design, the key schedule algorithm seems secure too. The key schedule of FOX is more complex compared with other existing block ciphers, and FOX uses the round function as a compress function to generate subkeys from the master key. This structure is very difficult to acquire information about master key or other subkeys from one certain subkey, besides, finding a collision from the key schedule algorithm is difficult too.

The authors of FOX analyze the security of FOX against linear attacks, differential attacks, integral attacks, statistical attacks, slide attacks interpolation attacks and algebraic attacks [1]. The integral attack is currently the most efficient attack on FOX. In 2006, Wu made some improvement based on the original integral attacks [3], and the improved integral attack can break 4, 5, 6 and 7 round FOX64 with $2^{45.4}$, $2^{109.4}$, $2^{173.4}$ and $2^{237.4}$ full-round encryptions respectively and 2^9 chosen plaintexts. In 2008, Nakahara designed a key recovery attack on 2-round FOX and an impossible differential attack on 5 round FOX [4]. The impossible differential attack presented by Nakahara requires 2^{118} times of encryptions and 2^{36} chosen plaintexts.

Impossible differential attack [5] is a kind of cryptanalysis for block ciphers. Compared with the ordinary differential cryptanalysis, impossible differential cryptanalysis considers the differences that are impossible at some intermediate state of the cipher algorithm. In this paper, we present a 4-round impossible differential, and by using the structure properties of $f32$ in the round function, we design an impossible differential attack on FOX, the method can attack 5, 6 and 7-round FOX64 with 2^{71}, 2^{135} and 2^{199} one-round encryptions respectively, which is the best know attack to FOX cipher.

The paper is organized as follows: Section 2 briefly introduces the FOX block cipher and some properties needed by the attack. In Section 3, we present a 4 round impossible differential on FOX. The attack on FOX is introduced in Section 4. Section 5 gives the conclusion.

2. PRELIMINARIES

The members of FOX family are denoted as follows:

Name	Block Size	Key size	No. rounds
FOX64	64	128	16
FOX128	128	256	16
FOX64/k/r	64	k	r
FOX128/k/r	128	k	r

In FOX64/k/r and FOX128/k/r, the number of round r must satisfy $12 \leq r \leq 255$, and the key length k, which

is multiple of 8, must satisfy $0 \leq k \leq 256$.

The round function of FOX64 uses a modified Lai-Massey Scheme, which consists of 2 parts: nonlinear part and linear part. The nonlinear part, denoted as $f(x)$, divides the inputs into two halves and exclusive-or them together as the input of $f32$ function, then these two halves of inputs exclusive-or the output of $f32$ function respectively. The nonlinear part is represented as follow,

$$f(y_1 \| y_2) = (x_1 \oplus f32(x_1 \oplus x_2) \| x_2 \oplus f32(x_1 \oplus x_2))$$

In the linear part, FOX introduces the orthomorphism function called *or*. It is a function taking a 32-bit input $X_1 \| X_2$ and returning a 32-bit output $Y_1 \| Y_2$, where $Y_1 = X_2$ and $Y_2 = X_1 \oplus X_2$. The linear part denoted as $g(x)$, it divides the inputs into four parts and makes *or* function on first two parts, the other two parts remain the same. The linear part is represented as follow,

$$g(x_1 \| x_2 \| x_3 \| x_4) = (x_2 \| x_1 \oplus x_2 \| x_3 \| x_4)$$

The whole encryption process consists of 16 rounds, 15 of which contain linear and nonlinear functions, while the last round only contains the nonlinear part.

The function $f32$ is a byte-wise nonlinear function. It consists of three parts: a substitution part denoted as $sigma8$, a diffusion part denoted as $mu8$, and an adding round key part. Denote the subkey as $RK_0 \| RK_1$, the $f32$ function can be expressed as:

$$f32(x) = sigma8(mu8(sigma8(x \oplus RK_0)) \oplus RK_1) \oplus RK_0$$

The $sigma8$ consists of 4 parallel nonlinear *S-box* (see detailed description in [1]), $mu8$ considers the input $x_1 \| .. \| x_4$ as a vector over $GF(2^8)$ and multiplies it with a matrix. The matrix is:

$$M = \begin{pmatrix} 1 & 1 & 1 & \alpha \\ 1 & z & \alpha & 1 \\ z & \alpha & 1 & 1 \\ \alpha & 1 & z & 1 \end{pmatrix}$$

Then we discuss some properties of FOX and make use of them on the attacks.

Lemma 1 *Output difference of* S-box *is zero if and only if the input difference of* S-box *is zero.*

Proof: Since *S-box* of FOX is a bijective mapping and the size of input and output is the same, then the *S-box* is a permutation. Thus different inputs of *S-box* would get different outputs, and vice versa.

Lemma 2 *If only one of the four input differences of $mu4$ is non-zero, the four bytes of output differences are all non-zero.*

Proof: Denote the input as a vector $\alpha = \{a_1, a_2, a_3, a_4\}^T$ over $GF(2^8)$, denote the non-zero element in α as a_j. When

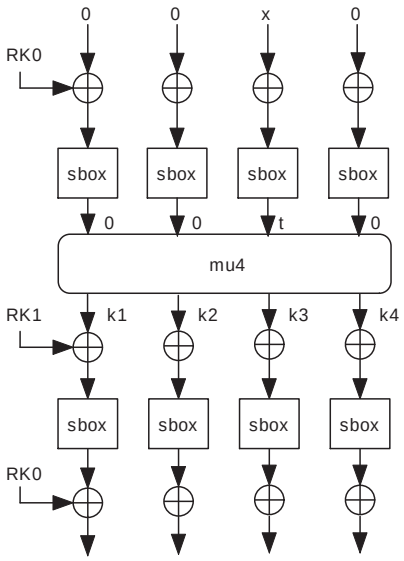

Fig. 1. Structure of $f32$. If only one input difference of the function $f32$ is non-zero, every byte of the output difference is distinct.

$a_k = 0, k \neq j, 1 \leq k \leq 4$, the result of matrix multiplication is:

$$\begin{aligned} M\alpha &= \{M_{1,j}a_j, M_{2,j}a_j, M_{3,j}a_j, M_{4,j}a_j\}^T \\ &= \{b_1, b_2, b_3, b_4\}^T \end{aligned}$$

Since $M_{i,j}$ and a_j are not zero, $b_i \neq 0$, for any i.

Lemma 3 *For the nonlinear part $f(x)$, the exclusive-or result of 1st and 2nd half of input is equal to exclusive-or result of 1st and 2nd half of output, and vice versa.*

Proof: From the high level structure of round function, it is easy to see that:

$$(x_1 \oplus f32(x_1 \oplus x_2)) \oplus (x_2 \oplus f32(x_1 \oplus x_2)) = x_1 \oplus x_2$$

Lemma 4 *When the input of $f32$ function is difference in only one byte, the output difference of $f32$ is distinct in every byte.*

Proof: See Fig. 1, without of loss of generality, we assume input difference of $f32$ is $0 \| 0 \| x \| 0, x \neq 0$, then the input difference of $mu4$ is 0 in 1st, 2nd and 4th bytes $(0 \| 0 \| t \| 0)$. According to Lemma 1, if input is different, then the output of *S-box* is different too, i.e. $t \neq 0$.

According to Lemma 2 the outputs of $mu4$ are different in each of four bytes ($k_1 \| k_2 \| k_3 \| k_4$ in Fig. 1). Since adding RK does not change the difference, the output difference of $f32$ is not equal to zero in all 4 bytes.

3. 4-ROUND IMPOSSIBLE DIFFERENTIAL OF FOX64

In the section, we will give the detailed description of the 4-round impossible differential.

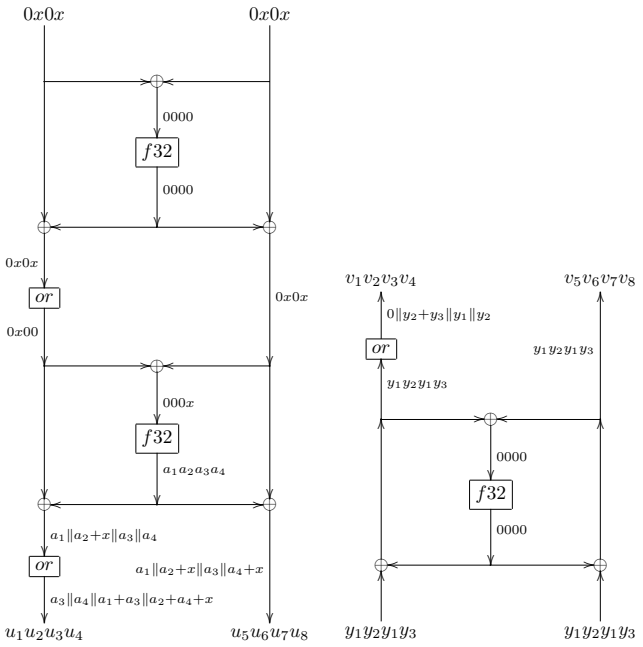

Fig. 2. Intermediate state of 4-Round impossible differential. The 1st and 2nd round states of the impossible differential are shown on left figure, the 4th round is shown on right.

Theorem 1 *In a r-round FOX64, when the input difference of* $(r-3)$*-th round is* $0\|x\|0\|x\|0\|x\|0\|x$*, the output difference of* r*-th round is impossible to be* $y_1\|y_2\|y_1\|y_3\|y_1\|y_2\|y_1\|y_3$*. Here* x*,* y_1*,* y_2 *and* y_3 *can be any value from* $0x00$ *to* $0xFF$*.*

Note that no matter what subkeys are selected, Theorem 1 always holds. In the following part, we will give the proof of the impossible differential.

Choose a pair of plaintext:

$$P_1 = p_1\|p_2\|p_3\|p_4\|p_5\|p_6\|p_7\|p_8$$
$$P_2 = p_1\|p_2'\|p_3\|p_4'\|p_5\|p_6'\|p_7\|p_8'$$

where $p_2 \oplus p_2' = p_4 \oplus p_4' = p_6 \oplus p_6' = p_8 \oplus p_8' = x \neq 0$. Thus the difference between the two plaintexts is $\Delta P = 0\|x\|0\|x\|0\|x\|0\|x$.

Because the input difference of $f32$ function of the 1st round is zero, the output difference of $f32$ is zero too. As a result the difference between outputs of the 1st round can be written as:

$$\Delta S_1 = 0\|x\|0\|0\|0\|x\|0\|x$$

The input difference of $f32$ of the 2nd round is $\Delta = 0\|0\|0\|x$. Denote the output difference of $f32$ in round 2 as $\Delta = a_1\|a_2\|a_3\|a_4$. According to Lemma 4, a_1, a_2, a_3 and a_4 are not equal to zero.

The output difference of round 2 can be written as:

$$\Delta S_2 = a_3\|a_4\|a_1 \oplus a_3\|a_2 \oplus a_4 \oplus x\|a_1\|x \oplus a_2\|a_3\|x \oplus a_4$$

Denote the difference of S_2 as $u_1..u_8$.

On the other hand, choose a pair of cipher text

$$C_1 = c_1\|c_2\|c_3\|c_4\|c_5\|c_6\|c_7\|c_8$$
$$C_2 = c_1'\|c_2'\|c_3'\|c_4'\|c_5'\|c_6'\|c_7'\|c_8'$$

satisfying

$$c_1 \oplus c_1' = c_3 \oplus c_3' = c_5 \oplus c_5' = c_7 \oplus c_7'$$
$$c_2 \oplus c_2' = c_6 \oplus c_6'$$
$$c_4 \oplus c_4' = c_8 \oplus c_8'$$

Denote the difference results of three formula above as y_1,y_2 and y_3 respectively. That is the difference in cipher text is:

$$\Delta S_4 = y_1\|y_2\|y_1\|y_3\|y_1\|y_2\|y_1\|y_3$$

Because the the round function in last round does not contain the *or* function, the input difference of $f32$ in round 4 is zero. Then the output differences of round 3 are the same as it is in round 4.

$$\Delta S_3 = y_1\|y_2\|y_1\|y_3\|y_1\|y_2\|y_1\|y_3$$

By applying the inverse *or* transform, the difference becomes:

$$\Delta T = 0\|y_2 \oplus y_3\|y_1\|y_2\|y_1\|y_2\|y_1\|y_3$$

Denote the difference of T as $v_1..v_8$.

According to the structure of FOX round function, $u_i \oplus u_{i+4} = v_i \oplus v_{i+4}$, $i = 1..4$, thus $u_3 \oplus u_7 = v_3 \oplus v_7$, that is $a_1 = 0$, which makes a conflict. Similarly, cipher texts pair with difference $y_1\|y_2\|y_3\|y_2\|y_1\|y_2\|y_3\|y_2$ would make a conflict too.

Note that a random pair of plaintexts would have a pair of cipher texts with above difference with probability about 2^{-39}.

4. IMPOSSIBLE DIFFERENTIAL ATTACK ON FOX

Select all plaintexts whose forms are $x_1\|x_2\|x_3\|x_4\|x_1 \oplus c_1\|x_5\|x_3 \oplus c_2\|x_4 \oplus c_3$, where c_1, c_2 and c_3 are constants and $x_1, \cdots, x_5$ can take the values from 0x00 to 0xFF respectively but $x_2 \oplus x_5 \neq 0$. Here we have $2^{40} - 2^{32}$ plaintexts. Every two of the plaintexts can construct a text pair for later use, and we have $(2^{40} - 2^{32})^2/2 \approx 2^{79}$ such pairs. The difference of each pair have the form $x_1\|x_2\|x_3\|x_4\|x_1\|x_5\|x_3\|x_4$.

Then verify corresponding cipher text pairs, and discard those pairs whose output difference is not as what is mentioned in the above section. There are about $2^{79}2^{-39} = 2^{40}$ pairs of texts remained. Note that the selecting plaintext phase requires about 2^{43} times one-round encryptions.

For every remained pair, it is easy to prove, when the output difference of $f32$ in the 1st round is $x_1\|x_2\|x_3\|x_2 \oplus x_4 \oplus x_5$, the input difference of round 2 is $0\|x_2 \oplus x_5\|0\|x_2 \oplus$

$x_5\|0\|x_2 \oplus x_5\|0\|x_2 \oplus x_5$, which is the input form of impossible difference mentioned above. Thus for every remaining pairs, when the 1st round subkey would cause the $f32$ in the 1st round have a output difference $x_1\|x_2\|x_3\|x_2 \oplus x_4 \oplus x_5$, we can conclude that this subkey is wrong.

For each pair of plaintext, we try to remove the wrong keys which can make the output of $f32$ function of 1st round to be $x_1\|x_2\|x_3\|x_2 \oplus x_4 \oplus x_5$.

Given one pair of plaintext P_1 and P_2, satisfying the condition above, remove those keys which can make the output of $f32$ function of 1st round to be $x_1\|x_2\|x_3\|x_2 \oplus x_4 \oplus x_5$.

1. Enumerate all possibility of the RK_0.
2. Calculate $mu4(Sbox(P_1 \oplus RK_0))$, $mu4(Sbox(P_2 \oplus RK_0))$, and $mu4(Sbox(P_1 \oplus RK_0)) \oplus mu4(Sbox(P_2 \oplus RK_0))$.
3. Assume the output difference of $f32$ is $\Delta C = x_1\|x_2\|x_3\|x_2 \oplus x_4 \oplus x_5$.
4. For each *S-box*, find the set $S = \{(T_1, T_2)|Sbox(T_1) \oplus Sbox(T_2) = \Delta C\}$ by looking up input-output difference table of each *S-box* with values from Step 2 and Step 3.
5. For each element in S, recover RK_1 from the result in step 2, and combine it with RK_0 together to be the wrong subkeys of 1st round.

The Step 1 requires 2^{32} times of guessing. Since each of four *S-box* affects one byte of texts, we can calculate the element of S byte by byte. Because the input-output difference table is not uniformly distributed, there may be several possible input pairs that would get the corresponding byte of ΔC in each *S-box*. Therefore, we should consider every four possible input pairs together in order to determine the elements of S. The time complexity of Step 4 is the same as looking up the input-output difference table of each *S-box*.

It is easy to prove, for any RK_0 there exists one RK_1 on average such that the combination of RK_0 and RK_1 is a wrong key. For every pairs of plaintexts, we can remove 2^{32} wrong keys on average. After analyzing 2^{39} pairs of plaintexts, there are about $2^{64}(1-2^{32})^{2^{39}} \approx 2^{64}e^{-2^7} \approx 2^{-118}$ wrong subkey remaining. The time complexity is about $2^{39}2^{32} = 2^{71}$ one-round encryptions, which is lower than the time complexity of the best known method of integral attack [3]. It is easy to extend the attack to 6 and 7 round FOX64 by simply guessing one or two round subkey, the corresponding complexities are about 2^{135} and 2^{199} one-round encryptions.

Similarly to the attacks of FOX64, there is an impossible differential pair in FOX128:

$$\Delta P = 0x0x\|0x0x\|0000\|0000$$
$$\Delta C = y_1y_2y_3y_4\|y_1y_2y_3y_4\|y_5y_6y_5y_7\|y_5y_6y_5y_7$$

The attack requires 2^{72} chosen plaintexts and 2^{135} one round encryptions.

5. CONCLUSION

In this paper, we use impossible differential attacks to analyze the security properties of FOX. We find a 4 round impossible differential in FOX functions and apply it on the impossible cryptanalysis of FOX. The attacks can break 5, 6 and 7 round FOX64 with 2^{71}, 2^{135} and 2^{199} one-round encryptions respectively, it requires 2^{40} chosen plaintexts. The attack presented in the paper is the best know method against FOX. The attack can also be applied to 5-round FOX128 with 2^{135} one round encryptions. However, the full-round FOX is still safe now. A comparison of known attacks on FOX are listed in the following table. The complexity is counted in terms of the number full-round encryptions.

Name	Round	No. Encry	Notes
FOX64	5	2^{69}	this paper
FOX64	5	$2^{109.4}$	[3]
FOX64	5	2^{118}	[4]
FOX64	6	2^{133}	this paper
FOX64	6	$2^{173.4}$	[3]
FOX64	7	2^{197}	this paper
FOX64	7	$2^{237.4}$	[3]
FOX128	5	2^{135}	this paper
FOX128	5	$2^{205.6}$	[3]

6. REFERENCES

[1] P. Junod and S. Vaudenay, "Fox: A new family of block ciphers," in *Selected Areas in Cryptography - SAC'04*, 2004, LNCS.

[2] S. Vaudenay, "On the lai-massey scheme," in *Advances in Cryptology - ASIACRYPT'99*, 1999, LNCS.

[3] W. Wu, W. Zhang, and D. Feng, "Improved integral cryptanalysis of fox block cipher," in *Information Security and Cryptology - ICISC'05*, 2005, LNCS.

[4] J. Nakahara, "An analysis of fox," in *Brazilian Symposium on Information and Computer System Security*, 2008.

[5] L. Knudsen, "Deal - a 128-bit block cipher," in *NIST AES Proposal*, 1998.