New Cryptanalysis of Block Ciphers with Low Algebraic Degree

Bing Sun[1], Longjiang Qu[1], and Chao Li[1,2]

[1] Department of Mathematics and System Science, Science College of National University of Defense Technology, Changsha, China, 410073
[2] State Key Laboratory of Information Security, Graduate University of Chinese Academy of Sciences, China, 10039
happy_come@163.com, ljqu_happy@hotmail.com

Abstract. Improved interpolation attack and new integral attack are proposed in this paper, and they can be applied to block ciphers using round functions with low algebraic degree. In the new attacks, we can determine not only the degree of the polynomial, but also coefficients of some special terms. Thus instead of guessing the round keys one by one, we can get the round keys by solving some algebraic equations over finite field. The new methods are applied to $\mathcal{PURE}$ block cipher successfully. The improved interpolation attacks can recover the first round key of 8-round $\mathcal{PURE}$ in less than a second; r-round $\mathcal{PURE}$ with $r \leq 21$ is breakable with about 3^{r-2} chosen plaintexts and the time complexity is 3^{r-2} encryptions; 22-round $\mathcal{PURE}$ is breakable with both data and time complexities being about 3×3^{20}. The new integral attacks can break $\mathcal{PURE}$ with rounds up to 21 with 2^{32} encryptions and 22-round with 3×2^{32} encryptions. This means that $\mathcal{PURE}$ with up to 22 rounds is breakable on a personal computer.

Keywords: block cipher, Feistel cipher, interpolation attack, integral attack.

1 Introduction

For some ciphers, the round function can be described either by a low degree polynomial or by a quotient of two low degree polynomials over finite field with characteristic 2. These ciphers are breakable by using the interpolation attack, which was first introduced by Jakobsen and Knudsen at FSE'97[2]. This attack was generalized by K. Aoki at SAC'99[3], which is called the linear sum attack, and a method was presented that can efficiently evaluate the security of byte-oriented ciphers against interpolation attack. In [4], the authors pointed some mistakes in [2], and introduced a new method, root finding interpolation attack, to efficiently find all the equivalent keys of the cipher, and this attack can decrease the complexity of interpolation attack dramatically. To apply the interpolation attack, a finite field should be constructed first, in [5], the effect of the choice of the irreducible polynomial used to construct the finite field was studied and an explicit relation between the Lagrange interpolation formula and the Galois Field Fourier Transform was presented.

O. Dunkelman (Ed.): FSE 2009, LNCS 5665, pp. 180–192, 2009.

Interpolation attack can be applied to some ciphers which have provable securities against differential and linear cryptanalysis[15,16]. For example, in [2], a provable secure block cipher $\mathcal{PURE}$ was introduced, however, it can be broken by using interpolation attack. Later, interpolation attack was successfully applied to some simplified version of SNAKE[17,18]. However, the complexity of interpolation attack on 6-round $\mathcal{PURE}$ is 2^{36}, and it will increase when the round of the cipher becomes 7,8 and so on. In another word, it is not a real-world attack.

Integral cryptanalysis[7,8] considers the propagation of sums of (many) values. Thus it can be seen as a dual to differential cryptanalysis which considers the propagation of sums of only two values. It was first proposed in [6] but under a different name, that is square attack. A number of these ideas have been exploited, such as square attack[19,20], saturation attack[9], multiset attack[12,10], and higher order differential attack[11,13]. Integrals have a number of interesting features. They are especially well-suited to analysis of ciphers with primarily bijective components. Moreover, they exploit the simultaneous relationship between many encryptions, in contrast to differential cryptanalysis where one considers only pairs of encryptions. Consequently, integrals apply to a number of ciphers not vulnerable to differential and linear cryptanalysis. These features have made integrals an increasingly popular tool in recent cryptanalysis work.

Integral attacks are well-known to be effective against byte-oriented block ciphers. In [14], the authors outlined how to launch integral attacks against bit-based block ciphers. The new type of integral attack traces the propagation of the plaintext structure at bit-level by incorporating bit-pattern based notations. The new integral attack is applied to Noekeon, Serpent and Present reduced up to 5, 6 and 7 rounds, respectively.

In this paper, by using an algebraic method, an improved interpolation attack and a new integral attack are proposed. The complexity of interpolation attack can be decreased dramatically which leads to a real-world attack against $\mathcal{PURE}$ with up to 22 rounds. There are two improvements in this paper. The first one is an improvement of the original interpolation attack. Instead of guessing the last round key one by one, we find some algebraic equations that can efficiently find the round key. Another one is an extended integral cryptanalysis and it is somewhat like the square attack. In a square attack, value of $\sum_x f(x)$ is computed. And in our attack, value of $\sum_x x^i f(x)$ for some integer i is computed and this value can be either a constant or strongly related with only a few round-keys. Thus instead of guessing the last round key one by one, we can get the round keys by solving some algebraic equation $f_C(K) = 0$ over finite field, where C is an arbitrarily chosen constant.

The paper is organized as follows: Feistel Structure and basic attacks are presented in section 2. In section 3, we introduce the basic mathematical foundations that can efficiently improve the attacks. And the improved interpolation attack is presented in section 4. Then, in section 5, new integral cryptanalysis is presented. Results of attack against $\mathcal{PURE}$ are given in section 6. Section 7 makes the conclusion of this paper.

2 Feistel Structure and Basic Attacks

2.1 Feistel Structure

A Feistel network consists of r rounds, each of which is defined as follows. Denote by (L, R) the $2n$-bit input, set $\alpha_0 = L$ and $\beta_0 = R$, let $(\alpha_{i-1}, \beta_{i-1})$ be the input to the ith round, (α_i, β_i) and k_i be the output and the round key of the ith round, respectively. Then $(\alpha_i, \beta_i) = Round(\alpha_{i-1}, \beta_{i-1})$ is defined as:

$$\begin{cases} \alpha_i = & \beta_{i-1}, \\ \beta_i = & f(\beta_{i-1}, k_i) \oplus \alpha_{i-1}, \end{cases}$$

where f is the round function and in this paper, we always assume that $f(\beta_{i-1}, k_i) = f(\beta_{i-1} \oplus k_i)$. See Fig.1. After iterating *Round* r times, the ciphertext (C_L, C_R) is defined as (β_r, α_r).

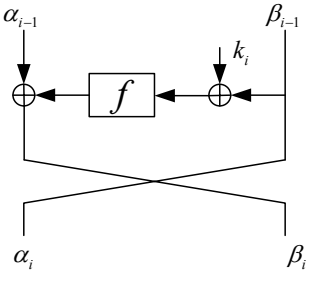

Fig. 1. Feistel Structure

2.2 Interpolation Attack on Block Ciphers

Let F be a field. Given $2t$ elements $x_1, \ldots, x_t, y_1, \ldots, y_t \in F$, where the x_is are distinct. According to Lagrange interpolation formula,

$$f(x) = \sum_{i=1}^{t} y_i \prod_{1 \le j \le t, j \ne i} \frac{x - x_j}{x_i - x_j}$$

is the only polynomial over F with degree at most $t-1$ such that $f(x_i) = y_i$ for $i = 1, \ldots, t$.

In an interpolation attack to an r-round Feistel cipher, we construct polynomials by using pairs of plaintexts and ciphertexts. The attacker first computes the degree of the output of $(r-1)$-th round, say N. Then he chooses $N+2$ plaintexts P_i and encrypts them, denote by C_i the corresponding ciphertexts. By guessing the last round key k^*, the attacker partially decrypts C_i one round back and gets D_i. Now, he uses (P_i, D_i) for $1 \le i \le N+1$ and applies the Lagrange interpolation formula to get the only polynomial $h(x)$ with degree at

most N such that $h(P_i) = D_i (1 \leq i \leq N+1)$. If $h(P_{N+2}) = D_{N+2}$, then put k^* as a candidate of the right key, otherwise, k^* is rejected. This process is repeated until the k^* is uniquely determined.

Assume k^* is an n-bit word, then the complexity of the interpolation attack is at least $(N+2) \times 2^n$, since to get the ciphertexts, it needs $N+2$ encryptions and 2^n partially decryptions for each ciphertext.

2.3 Integral Cryptanalysis

Let $(G, +)$ be a finite group and S be a subgroup of G. An integral over S is defined as the sum of all elements in S. That is,

$$\int S = \sum_{v \in S} v,$$

where the summation is defined in terms of the group operation for G.

In an integral attack, one tries to predict the values in the integral after a certain number of rounds. To be more exact, assume the input is x, and part or all of the output is $c(x)$, by computing $\sum_{x \in S} c(x)$, where S always denotes the finite field $\mathbb{F}_{2^t}$ for some integer t, one can distinguish the cipher from a random permutation. For example, in square attack, one adopts $\sum_{x \in S} c(x) = 0$ to efficiently find the round keys of a given cipher. But, if $\sum_{x \in S} c(x) = 0$, and let $h(x)$ be a nonlinear transformation, can we predict the value of $\sum_{x \in S} h(c(x))$? It seems that this is a difficult question if we cannot analyze h carefully.

Besides, most of the known integrals have the following form

$$\int (S, c) = \sum_{x \in S} c(x),$$

where x denotes the plaintext and c is the map from plaintext to ciphertext. However, in this paper, a new integral

$$\int (S, c, i) = \sum_{x \in S} x^i c(x)$$

for some integer i is proposed. This definition will facilitate our discussions in cryptanalysis.

3 Mathematical Foundation

3.1 Notations

The following notations will be used in this paper:

m	: degree of the round function
r	: rounds of the cipher
$2n$	: size of the plaintext/ciphertext
r_0	: $\lfloor \log_m (2^n - 1) \rfloor + 1$, the largest integer $\leq \log_m (2^n - 1) + 1$
$\deg(f)$	: degree of a polynomial f

To simplify the discussion, let the leading coefficient of $f(x)$ be 1:

$$f(x) = x^m \oplus \sum_{i=0}^{m-1} a_i x^i \in \mathbb{F}_{2^n}[x].$$

If $m = 1$ or $m = 2$, $f(x)$ is an affine function, thus we always assume $m \geq 3$.

3.2 Algebraic Analysis of Outputs of Feistel Cipher

By interpolation, an encryption algorithm can be seen as a polynomial function with the plaintext/ciphertext as its input/output. Thus, properties of this polynomial can be studied in order to get the information of the keys. If the round function has a low algebraic degree, then, the degree and some coefficients of special terms of the polynomial function between plaintexts and ciphertexts can be computed exactly.

Proposition 1. *Let $P = (C, x)$ be the input to an r-round Feistel cipher, where $C \in \mathbb{F}_{2^n}$ is a constant, $(\alpha_t, \beta_t) = (\alpha_t(x), \beta_t(x))$ be the output of the t-th round, if $1 \leq t \leq r-1$ and $m^{t-1} \leq 2^n - 1$, then*

$$\begin{cases} \deg \alpha_t = m^{t-1}, \\ \deg \beta_t = m^t, \end{cases}$$

where m is the degree of the round function. Furthermore, the leading coefficients of both $\alpha_t(x)$ and $\beta_t(x)$ are 1.

Proof. We can prove this proposition by induction.

If the input to the cipher is of the form $(\alpha_0, \beta_0) = (C, x)$ where C is a constant, then after the first round, $(\alpha_1, \beta_1) = (x, C \oplus f(x \oplus k_1))$. Therefore $\deg \alpha_1 = 1$, $\deg \beta_1 = \deg f = m$.

Assume $\deg \alpha_t = m^{t-1}, \deg \beta_t = m^t$, then

$$(\alpha_{t+1}, \beta_{t+1}) = (\beta_t, \alpha_t \oplus f(\beta_t \oplus k_t)),$$

thus $\deg \alpha_{t+1} = \deg \beta_t = m^t, \deg \beta_{t+1} = \deg \beta_t \times \deg f = m^{t+1}$. □

According to Proposition 1, (α_t, β_t) can be written in the following form:

$$(\alpha_t, \beta_t) = \left(x^{m^{t-1}} \oplus g_{t-1}(x), x^{m^t} \oplus g_t(x) \right), \tag{1}$$

where $g_i(x)$ is a polynomial with degree $< m^i$.

Proposition 1 determines the degree and leading coefficients of $\alpha_t(x)$ and $\beta_t(x)$. Now let's compute the coefficient of the term x^{m^t-1} in β_t, or equivalently, the leading coefficient of $g_t(x)$. This coefficient plays a very important role in the improvement of our new attacks. By induction, the following Proposition holds:

Proposition 2. *Assume $g_t(x) = \sum_{i=0}^{m^t-1} v_i x^i \in \mathbb{F}_{2^n}[x]$ is a polynomial defined as in (1), $m \equiv 1 \mod 2$ and $t \leq r_0 - 2$, then*

$$v_{m^t-1} = k_1 \oplus a_{m-1}.$$

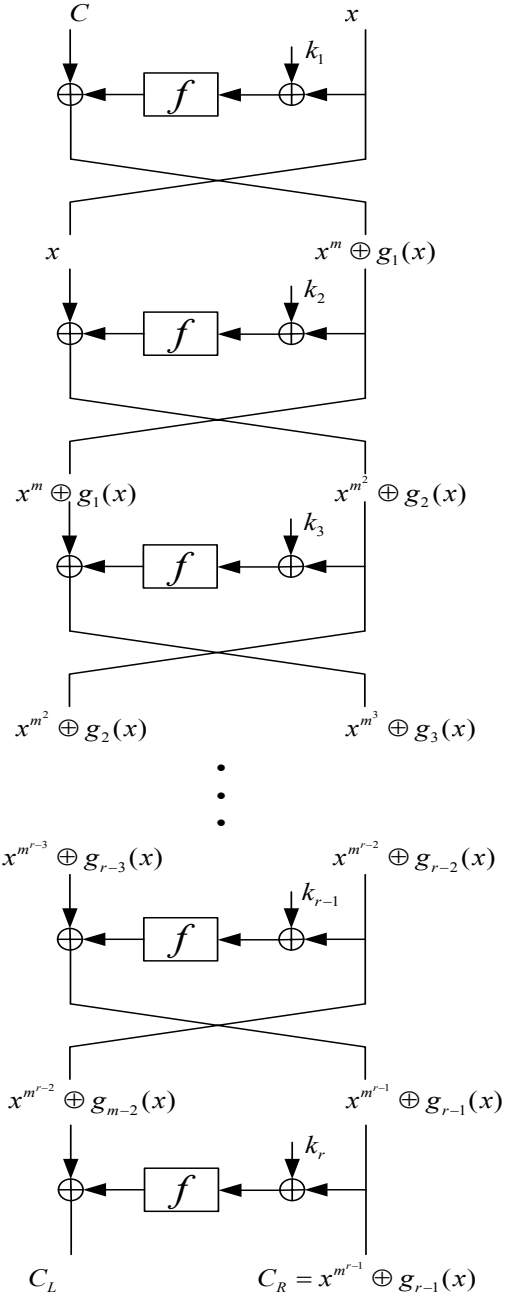

Fig. 2. Degrees of the output of each round

Proof. By computation, when $t = 1$:

$$\begin{aligned} x^m \oplus g_1(x) &= C \oplus f(x \oplus k_1) \\ &= (x \oplus k_1)^m \oplus a_{m-1}(x \oplus k_1)^{m-1} \cdots \\ &= x^m \oplus (k_1 \oplus a_{m-1})x^{m-1} \oplus \cdots \end{aligned}$$

Thus it is true for $t = 1$.

Assume $v_{m^t-1} = k_1 \oplus a_{m-1}$, then

$$\begin{aligned} x^{m^{t+1}} \oplus g_{t+1}(x) &= \alpha_t(x) \oplus f(\beta_t \oplus k_t) \\ &= x^{m^{t-1}} \oplus g_{t-1}(x) \oplus f(x^{m^t} \oplus g_t(x) \oplus k_t) \\ &= (x^{m^t} \oplus g_t(x) \oplus k_t)^m \oplus a_{m-1}(x^{m^t} \oplus g_t \oplus k_t)^{m-1} \oplus \cdots \\ &= (x^{m^t} \oplus v_{m^{t-1}} x^{m^t-1} \oplus \cdots)^m \oplus \cdots \\ &= x^{m^{t+1}} \oplus (m \times v_{m^{t-1}})x^{m^{t+1}-1} \oplus \cdots \end{aligned}$$

Thus $v_{m^{t+1}-1} = m \times v_{m^t-1} = k_1 \oplus a_{m-1}$, which ends our proof. □

4 Improved Interpolation Attack on Feistel Ciphers

4.1 Basic Properties of the Output of A Feistel Cipher

According to Proposition 1 and 2, we can determine not only the degree of the polynomial, but also coefficients of some special terms.

Theorem 1. *For an r-round $2n$-bit Feistel cipher, let the algebraic degree of the round function be an odd integer m, $r_0 = \lfloor \log_m (2^n - 1) \rfloor + 1$ and $r \leq r_0$. If the input to the cipher is of the form $P = (C, x)$ where $C \in \mathbb{F}_{2^n}$ is a constant, then the right half of the ciphertext is of the form $C_R(x) = x^{m^{r-1}} \oplus (k_1 \oplus a_{m-1})x^{m^{r-1}-1} \oplus q(x)$ where $q(x) \in \mathbb{F}_{2^n}[x]$ is a polynomial with degree $< m^{r-1} - 1$.*

Similar with Theorem 1, we can get the explicit expression of the output of an $(r_0 + 1)$-round Feistel cipher:

Theorem 2. *Let $r_0 = \lfloor \log_m (2^n - 1) \rfloor + 1$ and $r = r_0 + 1$, then for an r-round $2n$-bit Feistel cipher with the algebraic degree of round function being an odd integer m, if the input to the cipher is of the form $P = (x, C)$ where $C \in \mathbb{F}_{2^n}$ is a constant, then the right half of the ciphertext is of the form $C_R(x) = x^{m^{r-2}} \oplus (f(k_1 \oplus C) \oplus k_2 \oplus a_{m-1})x^{m^{r-2}-1} \oplus p(x)$ where $p(x) \in \mathbb{F}_{2^n}[x]$ is a polynomial with degree $< m^{r-2} - 1$.*

The above two Theorems have already been used in the original interpolation attack on $\mathcal{PURE}$, however, we use them in a different manner.

To improve the interpolation attack on Feistel ciphers with low algebraic degree, we always assume that the degree of the round function is an odd integer, that is $m \equiv 1 \mod 2$.

4.2 Improved Attack

In an interpolation attack, the attacker needs to guess the last round key, thus the complexity of the attack is at least $(N+2)\times 2^n$. In our improved interpolation attack, we can compute the first round key k_1 by only using the plaintexts and corresponding ciphertexts.

Feistel cipher with r round can be broken by the following attack:

Algorithm 1: Attack on Block Ciphers with $r \leq r_0$(I)

Step 1: Encrypt $P = (C, x)$ for $m^{r-1}+1$ different $x \in \mathbb{F}_{2^n}$ where $C \in \mathbb{F}_{2^n}$ is a constant. The corresponding ciphertexts are $(C_L(x), C_R(x))$;

Step 2: Compute $g(x) = ax^{m^{r-1}} \oplus sx^{m^{r-1}-1} \oplus \cdots \in \mathbb{F}_{2^n}[x]$ by interpolation such that $g(x) = C_R(x)$. According to Theorem 1, $k_1 = s \oplus a_{m-1}$.

Algorithm 1 needs $m^{r-1}+1$ encryptions, and to compute the interpolation polynomial, it needs $2 \times (m^{r-1}+1)$ word-memories to store (P_i, C_i). It is infeasible to mount a real-world attack when m^{r-1} is too large that a computer cannot store so many plaintexts/ciphertexts.

Algorithm 2 finds the first and second round keys by solving some algebraic equations over finite field instead of guessing the keys one by one.

Algorithm 2: Attack on Block Ciphers with $r \leq r_0 + 1$(I)

Step 1: Encrypt $P^{(1)} = (x, C_1)$ for $m^{r-2}+1$ different $x \in \mathbb{F}_{2^n}$ where $C_1 \in \mathbb{F}_{2^n}$ is a constant. The corresponding ciphertexts are $(C_L^{(1)}(x), C_R^{(1)}(x))$;

Step 2: Compute $g(x) = ax^{m^{r-2}} \oplus s_1 x^{m^{r-2}-1} \oplus \cdots \in \mathbb{F}_{2^n}[x]$ by interpolation such that $g(x) = C_R^{(1)}(x)$; thus $s_1 = f(k_1 \oplus C_1) \oplus k_2 \oplus a_{m-1}$;

Step 3: Choose another two constants C_2 and C_3, repeat Step 1 and Step 2, then we get $s_2 = f(k_1 \oplus C_2) \oplus k_2 \oplus a_{m-1}$, $s_3 = f(k_1 \oplus C_3) \oplus k_2 \oplus a_{m-1}$;

Step 4: By finding the common roots of the following equations, we get k_1 and k_2.

$$\begin{cases} s_1 = f(k_1 \oplus C_1) \oplus k_2 \oplus a_{m-1} \\ s_2 = f(k_1 \oplus C_2) \oplus k_2 \oplus a_{m-1} \\ s_3 = f(k_1 \oplus C_3) \oplus k_2 \oplus a_{m-1} \end{cases} \tag{2}$$

To find the solution of (2), set $h_{ij}(k_1) = f(k_1 \oplus C_i) \oplus f(k_1 \oplus C_j) \oplus s_i \oplus s_j$ for $1 \leq i < j \leq 3$. Compute $d(k_1) = \gcd(h_{12}(k_1), h_{13}(k_1), h_{23}(k_1))$, the greatest common divisor of $h_{12}(k_1), h_{13}(k_1)$ and $h_{23}(k_1)$, with great probability, $d(k_1) = k_1 \oplus K^*$ where K^* is a constant in $\mathbb{F}_{2^n}$. Thus $k_1 = K^*$, therefore, $k_2 = s_1 \oplus f(k_1 \oplus C_1) \oplus a_{m-1}$.

Comparing with the original interpolation attack, Algorithms 1 and 2 do not need to guess the key candidates. Thus the complexity of these attacks are $m^{r-1}+1$ for Algorithm 1 and $3\times m^{r-2}+3$ for Algorithm 2, number of plaintexts to be encrypted.

5 New Integral Cryptanalysis of Block Ciphers

For 2^n pairs $(x_i, y_i) \in \mathbb{F}_{2^n}^2$ where x_is are distinct, to find the polynomial $f(x)$ of degree $\leq 2^n - 1$ such that $y_i = f(x_i)$, we can use the Lagrange interpolation formula. However, there is another way to compute $f(x)$.

Theorem 3. *[1] Let* $f(x) = \sum_{i=0}^{2^n-1} a_i x^i \in \mathbb{F}_{2^n}[x]$ *be a polynomial with degree at most* $2^n - 1$*, then*

$$a_i = \begin{cases} \sum_{x \in \mathbb{F}_{2^n}} x^{2^n-1-i} f(x) & \text{if } i \neq 0 \mod 2^n - 1, \\ f(0) & \text{if } i = 0, \\ \sum_{x \in \mathbb{F}_{2^n}} f(x) & \text{if } i = 2^n - 1. \end{cases}$$

If m^{r-1} or m^{r-2} is too large that the computer can not store $m^{r-1} + 1$ or $m^{r-2} + 1$ pairs of plaintext and ciphertext, we can use the following methods. The two new methods below need almost no memories to compute the round keys of a Feistel cipher. However, they need more plaintexts/ciphertexts.

Algorithm 3: Attack on Block Ciphers with $r \leq r_0$(II)

Step 1: Encrypt $P^{(1)} = (C, x)$ for all $x \in \mathbb{F}_{2^n}$ where $C \in \mathbb{F}_{2^n}$ is a constant. The corresponding ciphertexts are $(C_L(x), C_R(x))$;

Step 2: Compute $s = \sum_{x \in \mathbb{F}_{2^n}} x^{2^n - m^{r-1}} C_R(x)$, thus $k_1 = s \oplus a_{m-1}$.

Algorithm 4: Attack on Block Ciphers with $r \leq r_0 + 1$(II)

Step 1: Encrypt $P^{(1)} = (x, C_1)$ for all $x \in \mathbb{F}_{2^n}$ where $C_1 \in \mathbb{F}_{2^n}$ is a constant. The corresponding ciphertexts are $(C_L^{(1)}(x), C_R^{(1)}(x))$;

Step 2: Compute $s_1 = \sum_{x \in \mathbb{F}_{2^n}} x^{2^n - m^{r-2}} C_R^{(1)}(x)$;

Step 3: Choose another two constants $C_2, C_3 \in \mathbb{F}_{2^n}$, repeat step 1 and step 2, and compute $s_2 = \sum_{x \in \mathbb{F}_{2^n}} x^{2^n - m^{r-2}} C_R^{(2)}(x)$, $s_3 = \sum_{x \in \mathbb{F}_{2^n}} x^{2^n - m^{r-2}} C_R^{(3)}(x)$;

Step 4: Find the solution of

$$\begin{cases} s_1 = f(k_1 \oplus C_1) \oplus k_2 \oplus a_{m-1}, \\ s_2 = f(k_1 \oplus C_2) \oplus k_2 \oplus a_{m-1}, \\ s_3 = f(k_1 \oplus C_3) \oplus k_2 \oplus a_{m-1}. \end{cases}$$

Comparing Algorithms 3 and 4 with the original interpolation attack, there are some merits of the improved attacks:

(1) There is no need to store plaintexts and corresponding ciphertexts while these data should be stored in the original interpolation attack[2] as well as Algorithms 1 and 2;

(2) There is no need to guess the key candidates. Thus the complexity of these attacks are 2^n and 3×2^n respectively, number of plaintexts to be encrypted.

When applying square attack, one adopts $\sum_x y(x) = 0$. However, in the above attack, we analysis the cipher by computing $\sum_x x^i y(x)$ for some integer i. Thus square attack can be seen as a special case of the new integral attack introduced above.

NOTE 1: In Algorithm 1, for $(x_i, y_i) = (x_i, C_R(x_i))$, $1 \le i \le m^{r-1}+1$, by using the Lagrange interpolation formula and computing the coefficient s of the second highest term, we get:

$$s = \sum_{1 \le i \le m^{r-1}+1} \frac{y_i \sum\limits_{1 \le j \le m^{r-1}+1, j \ne i} x_j}{\prod\limits_{1 \le j \le m^{r-1}+1, j \ne i} (x_i - x_j)}. \tag{3}$$

Instead of interpolation, k_1 can be computed by (3), and this can be seen as another extension of integrals.

6 Results of Attack on $\mathcal{PURE}$

$\mathcal{PURE}$ is a Feistel cipher with $2n = 64$ and $f(x) = x^3 \in \mathbb{F}_{2^{32}}[x]$. Though it has a provable security against differential and linear cryptanalysis, it is breakable by interpolation attack with up to 32 rounds in [2]. However, it is very difficult to mount a real-world attack by the method presented in [2].

6.1 Improved Attacks on $\mathcal{PURE}$

If $r \le 21$, there are two cases in consideration:

1) If 3^{r-1} is too large, it is impossible to store so many data, thus by Theorem 1, the following equation holds:

$$\sum_{x \in \mathbb{F}_{2^n}} x^{2^{32}-3^{r-1}} C_R(x) = k_1. \tag{4}$$

So, k_1 can be recovered with both data and time complexities being 2^{32} respectively by using Algorithm 3. We implemented 15-round attack by using Algorithm 3, and the round key was recovered in less than 31 hours.

2) If 3^{r-1} is not too large, then the data an interpolation needs is not too large. For this case, we use Algorithm 1 by interpolation, it only needs $3^{r-1}+1$ plaintexts to recover k_1, with some more memories to store plaintexts/ciphertexts. We implemented 10-round attack by using Algorithm 1, and the round key was recovered in less than 5 minutes.

If $r = 22$, $\mathcal{PURE}$ is breakable with 3×2^{32} encryptions by using Theorem 2, and $3 \times 3^{r-2} + 3$ by using Theorem 1:

Step 1. Encrypt $P = (P^L, P_1^R)$ where $P^L \in \mathbb{F}_{2^{32}}$ takes all values of $\mathbb{F}_{2^{32}}$ and $P_1^R \in \mathbb{F}_{2^{32}}$ is a constant;
Step 2. For the corresponding ciphertexts $C = (C_L, C_R)$, compute $s_1 = \sum_{P^L} (P^L)^{2^{32}-3^{r-2}} C_R$;
Step 3. For P_2^R and P_3^R, do step 1 and step 2, then compute the corresponding s_2 and s_3.
Step 4. Solve the following equations to get k_1 and k_2:

$$\begin{cases} s_1 = (P_1^R \oplus k_1)^3 \oplus k_2 \\ s_2 = (P_2^R \oplus k_1)^3 \oplus k_2 \\ s_3 = (P_3^R \oplus k_1)^3 \oplus k_2 \end{cases} \tag{5}$$

and the solution is

$$\begin{cases} k_1 = \dfrac{s_1\left(P_2^R \oplus P_3^R\right) \oplus s_2\left(P_3^R \oplus P_1^R\right) \oplus s_3\left(P_1^R \oplus P_2^R\right)}{\left(P_1^R \oplus P_2^R\right)\left(P_2^R \oplus P_3^R\right)\left(P_3^R \oplus P_1^R\right)} \oplus \left(P_1^R \oplus P_2^R \oplus P_3^R\right) \\ k_2 = s_1 \oplus (P_1^R \oplus k_1)^3 \end{cases}$$

6.2 Experimental Results

Table 1 shows the results of the attack on reduced-round $\mathcal{PURE}$, these results are computed by using the algebraic software Magma.

Table 1. Experimental Results of Attack on Reduced-round $\mathcal{PURE}$

Round	Algorithm	Data	Memory	Time	CPU
8	1	3^7+1	3^7+1	3.5 seconds	Pentium(R)4,3.06GHz
8	2	3^6+1	3^6+1	1 second	Pentium(R)4,3.06GHz
10	1	3^8+1	3^8+1	4.5 minutes	Pentium(R)4,3.06GHz
10	2	3^9+1	3^9+1	1.5 minutes	Pentium(R)4,3.06GHz
15	3	2^{32}	neglectable	31 hours	Pentium(R)4,3.06GHz
22	4	3×2^{32}	neglectable	148 hours	Pentium(R)4,3.06GHz

7 Conclusion

Both interpolation and integral attacks are improved in this paper. If the cipher can be described as a low degree polynomial, the new attacks can decrease the complexity of the original interpolation attack dramatically, which sometimes leads to a real-world attack. For example, 20-round $\mathcal{PURE}$ is not breakable on a personal computer if one uses the original method introduced in [2], while our method can do so. There are some interesting problems, for example, the square attack can be seen as a special case of this attack, since $\sum_x y$ is a special case of $\sum_x x^i y$. So can we use similar method to analyze AES? Another question is, how to extend this attack to the case of rational polynomials, that is, if the cipher can be described as $g_1(x)/g_2(x)$, how to apply this attack?

Acknowledgment

The authors wish to thank Ruilin Li, Shaojing Fu, Wentao Zhang and the anonymous reviewers for their useful comments.

The work in this paper is partially supported by the Natural Science Foundation of China (No: 60573028, 60803156), and the open research fund of State Key Laboratory of Information Security(No: 01-07).

References

1. Lidl, R., Niederreiter, H.: Finite Fields. Encyclopedia of Mathematics and Its Applications, vol. 20. Cambridge University Press, Cambridge (1997)
2. Jakobsen, T., Knudsen, L.R.: The Interpolation Attack on Block Cipher. In: Biham, E. (ed.) FSE 1997. LNCS, vol. 1267, pp. 28–40. Springer, Heidelberg (1997)
3. Aoki, K.: Efficient Evaluation of Security against Generalized Interpolation Attack. In: Heys, H.M., Adams, C.M. (eds.) SAC 1999. LNCS, vol. 1758, pp. 135–146. Springer, Heidelberg (1999)
4. Kurosawa, K., Iwata, T., Quang, V.D.: Root Finding Interpolation Attack. In: Stinson, D.R., Tavares, S. (eds.) SAC 2000. LNCS, vol. 2012, pp. 303–314. Springer, Heidelberg (2001)
5. Youssef, A.M., Gong, G.: On the Interpolation Attacks on Block Ciphers. In: Schneier, B. (ed.) FSE 2000. LNCS, vol. 1978, pp. 109–120. Springer, Heidelberg (2001)
6. Daemen, J., Knudsen, L.R., Rijmen, V.: The Block Cipher Square. In: Biham, E. (ed.) FSE 1997. LNCS, vol. 1267, pp. 149–165. Springer, Heidelberg (1997)
7. Knudsen, L.R., Wagner, D.: Integral Cryptanalysis. In: Daemen, J., Rijmen, V. (eds.) FSE 2002. LNCS, vol. 2365, pp. 112–127. Springer, Heidelberg (2002)
8. Hu, Y., Zhang, Y., Xiao, G.: Integral Cryptanalysis of SAFER+. Electronics Letters 35(17), 1458–1459 (1999)
9. Lucks, S.: The Saturation Attack — A Bait for Twofish. In: Matsui, M. (ed.) FSE 2001. LNCS, vol. 2355, pp. 1–15. Springer, Heidelberg (2002)
10. Nakahara Jr., J., Freitas, D., Phan, R.: New Multiset Attacks on Rijndael with Large Blocks. In: Dawson, E., Vaudenay, S. (eds.) Mycrypt 2005. LNCS, vol. 3715, pp. 277–295. Springer, Heidelberg (2005)
11. Knudsen, L.R.: Truncated and Higher Order Differentials. In: Preneel, B. (ed.) FSE 1995. LNCS, vol. 1008, pp. 196–211. Springer, Heidelberg (1995)
12. Biryukov, A., Shamir, A.: Structural Cryptanalysis of SASAS. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 394–405. Springer, Heidelberg (2001)
13. Lai, X.: Higher Order Derivations and Differential Cryptanalysis. Communications and Cryptography: Two Sides of One Tapestry, pp. 227–233. Kluwer Academic Publishers, Dordrecht (1994)
14. Z'aba, M.R., Raddum, H., Henricksen, M., Dawson, E.: Bit-Pattern Based Integral Attack. In: Nyberg, K. (ed.) FSE 2008. LNCS, vol. 5086, pp. 363–381. Springer, Heidelberg (2008)
15. Biham, E., Shamir, A.: Differential Cryptanalysis of the Data Encryption Standard. Springer, Heidelberg (1993)
16. Matsui, M.: Linear Cryptanalysis Method for DES Cipher. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 386–397. Springer, Heidelberg (1993)
17. Lee, C., Cha, Y.: The Block Cipher: SNAKE with Provable Resistance against DC and LC Attacks. In: Proceedings of 1997 Korea-Japan Joint Workshop on Information Security and Cryptology (JW–ISC 1997), pp. 3–17 (1997)

18. Morial, S., Shimoyama, T., Kaneko, T.: Interpolation Attacks of the Block Cipher: SNAKE. In: Knudsen, L.R. (ed.) FSE 1999. LNCS, vol. 1636, pp. 275–289. Springer, Heidelberg (1999)
19. Daemen, J., Rijmen, V.: The Design of Rijndael: AES — The Advanced Encryption Standard (Information Security and Cryptography). Springer, Heidelberg (2002)
20. Ferguson, N., Kelsey, J., Lucks, S., Schneier, B., Stay, M., Wagner, D., Whiting, D.: Improved Cryptanalysis of Rijndael. In: Schneier, B. (ed.) FSE 2000. LNCS, vol. 1978, pp. 213–230. Springer, Heidelberg (2001)