Revisit and Cryptanalysis of a CAST Cipher

Xiao Zhou, Jingwei Li, Xuejia Lai and Hailun Yan

ABSTRACT

CAST family of ciphers are the block ciphers created by Carlisle Adams et al. in 1996. One famous member of CAST family is CAST-128, which is a 12-round or 16-round Feistel network. In this paper, we apply the interpolation attack on both 5-round and 16-round CAST-128 Cipher to recover the last round key. We first mount the basic interpolation attack separately on it. After combining with the higher order differential cryptanalysis, our result reduces the time complexity in $2^{31.4}$ bit operation for 5-round attack. We also apply the optimized interpolation attack on the 16-round CAST-128-like structure which requires $2^{44.2}$ bit operations and $2^{31}$ chosen plaintexts.

INTRODUCTION

CAST ciphers are a family of block ciphers constructed using CAST design procedure [1] created by Adams et al. in 1996, which appear to have good performance in resisting differential attack[10], linear attack[11] and related-key attack [9]. The components in CAST ciphers include S-boxes based on bent functions, key-dependent rotations, modular addition and subtraction and XOR operations. The round function has three alternating types which are similar in structure while the only difference is in the choices of the exact operation (addition, subtraction or XOR) at various points. CAST ciphers adopt the Feistel network and allow a variety of round functions. A CAST-128-like structure is well defined in [4].

Xiao Zhou, Jingwei Li, Xuejia Lai, Hailun, Yan, School of Electronic Information and Electrical Engineering, Department of Computer Science, Shanghai Jiao Tong University, Shanghai 200240, P. R. China.
Xuejia Lai, Westone Cryptologic Research Center, Beijing 100070, China
which uses S-boxes based on bent functions proposed for CAST-128, supposing the operations XOR in the round functions.

Higher order differential cryptanalysis is an efficient algebraic method introduced into cryptography [2] in 1994. Due to significant attack results, it has extensively been used in cryptanalysis [3][14][15]. This cryptanalysis uses the properties of higher order derivatives to obtain some attack equations about the last round keys and then recovers these keys by solving the corresponding system of linear equations.

Shortly afterwards, Jakobsen and Knudsen propose the interpolation attack [5]. This attack is similar to higher order differential cryptanalysis, which is efficient against block ciphers of the low degree round function. Using this cryptanalysis strategy, the most critical step is to interpolate some intermediate bits by exploiting known or chosen plaintexts. In 2015, Dinur et al. apply this interpolation attack on LowMC [8], which is a collection of block cipher families proposed by Albrecht et al. [17]

Some attacks on CAST ciphers and CAST-128-like structure have been proposed by Shiho et al. [4], Wang et al. [12] and Sun et al. [13]. A 5-round higher order differential attack of CAST-128-like structure is shown in [4]. This attack requires $2^{17}$ chosen plaintexts and $2^{37.7}$ bits operations. However, the attacker need to evaluate the encryption function and this procedure is less efficient.

In this paper, we revisit the cryptanalysis of CAST-128-like structure and apply the basic interpolation attack on it, which needs $2^{34.6}$ bit operations using $2^{19}$ chosen plaintexts. The basic interpolation attack brings a small complexity reduction, but fails to directly recover the secret keys.

Comparing two attacks above, we realize there are some improvement for time efficiency. The observation captures our attention and prompts us to improve the attack. Consequently, we mount a more efficient attack. First, the attack equation system can be derived efficiently using the higher order differential cryptanalysis. Then we transform a carefully chosen variable subset to variables linearized monomials in the key bits as the interpolation attack. This optimized 5-round attack needs $2^{19}$ chosen plaintexts, and the time complexity of the attack is about $2^{31.4}$ bit operations. Furthermore, we also apply our optimized interpolation attack on the full round CAST-128-like structure, which requires about $2^{44.2}$ bit operations and $2^{31}$ chosen plaintexts. The summarization of our attacks and previous attack is listed in Table 1, where the data complexity is evaluated by the number of chosen plaintexts and the time complexity is evaluated by the number of bit operation needed in the attack.

The paper is organized as follows. In section II, we introduce some notions used in the attack. CAST-128-like structure is described in section III. Section IV gives the detailed attack procedure. Finally, section IV concludes the results.
TABLE I. ATTACKS ON CAST-128-LIKE STRUCTURE.

<table>
<thead>
<tr>
<th>Rounds</th>
<th>Attack</th>
<th>Data</th>
<th>Time</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Higher order differential attack</td>
<td>$2^{17}$</td>
<td>$2^{37.7}$</td>
<td>[4]</td>
</tr>
<tr>
<td>5</td>
<td>Basic interpolation attack</td>
<td>$2^{19}$</td>
<td>$2^{34.6}$</td>
<td>below</td>
</tr>
<tr>
<td>5</td>
<td>Optimized interpolation attack</td>
<td>$2^{19}$</td>
<td>$2^{31.6}$</td>
<td>below</td>
</tr>
<tr>
<td>16</td>
<td>Optimized interpolation attack</td>
<td>$2^{31}$</td>
<td>$2^{44.2}$</td>
<td>below</td>
</tr>
</tbody>
</table>

PRELIMINARIES

In this section, we describe the notions used in the rest of this paper.

Boolean Algebra

For a finite set $S$, denote its size by $|S|$. Given a vector $u = (u_1, ..., u_n) \in GF(2^n)$, let $wt(u)$ denote its Hamming weight.

Any function $F$ from $GF(2^n)$ to $GF(2)$ can be described as a polynomial in $n$ variables, which can be represented in algebraic normal form (ANF). The ANF of $F$ is unique and given as:

$$F(x_1, ..., x_n) = \sum_{u=(u_1, ..., u_n) \in GF(2^n)} \alpha_u M_u = \sum_{u=(u_1, ..., u_n) \in GF(2^n)} \alpha_u \prod_{i=1}^{n} x_i^u.$$  \hspace{1cm} (1)

where $\alpha_u \in \{0,1\}$ is the coefficient of the monomial $M_u = \prod_{i=1}^{n} x_i^u$, and the sum of $F$ is over $GF(2)$. The algebraic degree of function $F$ is defined as $\text{Deg}(F) \triangleq \max\{wt(u) | \alpha_u \neq 0\}$. A function $F$ with a degree bounded by $d \leq n$ can be described using $\sum_{i=0}^{d} \binom{n}{i}$ coefficients. Here we define $\binom{n}{i} = \sum_{j=0}^{i} \binom{n}{j}$.

We can interpolate the ANF coefficient $\alpha_u$ of $F$ by summing over $2^{wt(u)}$ evaluations of $F$: first let $S = \{x = (x_1, ..., x_n) | \bar{u} \land x = 0\}$, where $\bar{u}$ is bitwise NOT operation of $u$ and $\land$ is bitwise AND operation, and then, $\alpha_u = \sum_{(x_1, ..., x_n) \in S} F(x_1, ..., x_n)$.

Moebius Transform

If any function $f$ is a bit vector of $2^n$ entries with the given truth table, the ANF of $f$ can be represented as a bit vector of $2^n$ entries corresponding to its $2^n$ coefficients $\alpha_u$. This ANF can be obtained by Moebius transform algorithm (MTA), which requires about $n \cdot 2^n$ bit operations. For example, when $n = 1$, here are 2 entries and the ANF of $f$ is given as:

$$f(x_0, x_1) = (0,0) \oplus [f(0,0) \oplus f(1,0)]x_0 \oplus [f(0,0) \oplus f(0,1)]x_1$$

$$\oplus [f(0,0) \oplus f(1,0) \oplus f(0,1) \oplus f(1,1)]x_0 x_1.$$
The details of Moebius transform algorithm (taken from [6]) are described in Figure 1. The reason why we use Moebius transform is to reduce the time complexity of the interpolation attack, which is explained in the next section.

**Algorithm 1: Moebius transform algorithm**

| Input | Truth table $S$ of Boolean function $f$, with $2^n$ entries. $Pos$ is the small table position, $S_i$ is the small table size |
| Output | The algebraic normal form of $f$ |

1. for $i = 0$ to $n - 1$ do
2. Let $Pos ← 0$, $S_2 ← 2^i$;
3. while ($Pos < 2^n$) do
4. for $j = 0$ to $S_i - 1$ do
6. Let $Pos ← Pos + 2 · S_i$;
7. return $S$;

Figure 1. Moebius transform algorithm.

**Higher Order Differential Cryptanalysis**

Higher order differentials attack was introduced in [2]. It is very efficient to attack the block ciphers of low algebraic degree. The critical idea of higher order differential cryptanalysis is to consider some target bit $b$ and analyze its ANF representation in terms of plaintexts $P$, denoted by $F_k(P)$, where $k$ is unknown fixed secret key.

**Proposition 1.** [2] Let $L[a_1, a_2, ..., a_l]$ be the list of all $2^l$ possible linear combinations of $a_1, a_2, ..., a_l$. Then

$$
\Delta^{(l)}_{a_1, ..., a_l} F(x) = \sum_{c \in L[a_1, a_2, ..., a_l]} F(x \oplus c)
$$

**Proposition 2.**[2] For any function $F: GF(2^n) \rightarrow GF(2^m)$, the $n$-th derivative of $F$ is a constant. If $F: GF(2^n) \rightarrow GF(2^m)$ is invertible, then $(n - 1)$-th derivative of $F$ is a constant.

Given the $\deg(F_k(P)) \leq d$, the attacker can choose an arbitrary linear subspace $S$ of dimension $(d + 1)$, and computes higher order differential over $S$ for target bit $b$ (namely, over $GF(2)$). With Proposition 1 and 2, we can derive that the sum is zero. Based on the higher order derivative properties, we can get a system of attack equations and solve it to recover the key.

**Interpolation Attack**

Interpolation attack is closely similar to higher order differential cryptanalysis, which is introduced by Jakobsen and Knudsen in 1997 [5]. It is particularly efficient against block ciphers whose round function is of low algebraic degree.
In the original idea of interpolation attack over \( \text{GF}(2) \) [5], the attacker views the ANF of some intermediate encryption bit \( b \) as an initially unknown polynomial \( F_k(C) \) in the ciphertext bits \( C = (c_1, ..., c_n) \) where \( k = (k_1, ..., k_m) \) is unknown fixed secret key. Thus we can write \( F_k(C) = \sum_{u=(u_1, ..., u_n) \in \text{GF}(2^n)} \alpha_u M_u \), where \( \alpha_u \in \{0,1\} \) is the coefficient of the monomial \( M_u = \prod_{i=1}^{n} c_i^{u_i} \). The unknown coefficients \( \alpha_u \) are generally related to the secret key, and the interpolation attacker aims to recover the unknown coefficients \( \alpha_u \) of \( F_k(C) \). Furthermore, it uses various ad-hoc techniques to recover the actual secret key which is not part of the framework describe in this section.

The data and time complexities of the attack depend on the value of the degree and the number of unknown coefficient \( \alpha_u \). In order to mount efficient interpolation attacks, the attacker need to minimize these two parameters, as we demonstrate in our attacks on CAST-128-like structure.

**Model of Computation**

Since the higher order differential attack [4] and our attacks use different bitwise operations, comparing these attacks cannot be completed by simply evaluating the number of encryption round function. For consistency, we compare the complexity of attacks by counting the number of bit operations (such as XOR, AND, OR). As calculated in Sec. 3, a straight-line implementation of one round function evaluation of CAST-128-like structure requires about \( 2^{11.7} \) bit operations.

**DESCRIPTION OF CAST-128-LIKE STRUCTURE**

Moriai et al. [4] presents a 5-round CAST-128-like structure, which are specified with both structure and round function in Figure 2. CAST ciphers are based on the Feistel network. The structure splits the 64-bit plaintext into left and right 32-bit halves, namely, \( L_0 \) and \( R_0 \). During the \( i \)-th round, the output of \( (i-1) \)-th round’s right part, the 32-bit data \( R_{i-1} \) is the input to the round function \( F \) along with corresponding round key \( k^i \). These two parties are combined with operation XOR, and then the 32-bit result is split into four 8-bit pieces. Each piece is the input to a different \( 8 \times 32 \) S-box \( (S_1, S_2, S_3, S_4) \) proposed for CAST-128. We have the following formulas for each round of CAST-128-like structure:

\[
L_i = R_{i-1},
\]

\[
R_i = F(R_{i-1}, k^i) \oplus L_{i-1},
\]

where \( L_i \) and \( R_i \) represent the left and right half of output of \( i \)-th round, respectively. For the last round, the output of the left part and right part exchange with each other, which forms the ciphertext.
Figure 3 shows the pseudo-code of the encryption function for N-round CAST-128 (taken from [15]).

Figure 2. The CAST-128-like structure in N-round.
INPUT: plaintext $m_1\ldots m_{64}$; key $K = k_1\ldots k_{128}$.
OUTPUT: ciphertext $c_1\ldots c_{64}$.

1. Key schedule: Compute $N$ pairs of subkeys from $K$.
2. Split plaintext into left and right 32-bit halves:
   \[ L_0 = m_1\ldots m_{32} \text{ and } R_0 = m_{33}\ldots m_{64}. \]
3. $N$ rounds: For $i$ from 1 to $N$, compute $L_i$ and $R_i$ as follows:
   \[ L_i = R_{i-1}; \]
   \[ R_i = L_{i-1} \oplus f(R_{i-1}, x_i), \] where $f$ is round function.
4. Exchange final blocks $L_N$, $R_N$ and form the ciphertext:
   \[ c_1\ldots c_{64} \leftarrow (R_N, L_N). \]

Figure 3. Pseudo-code of the encryption function for $N$-round CAST-128.

Note that for the last round, the ciphertext $C = (R_N, L_N)$, which means $L_N$ and $R_N$ exchange with each other twice in total.

CAST-128-like structure uses S-boxes, which is proposed for CAST-128. In [4]’s appendix A, it describes the obtained Boolean polynomials of some S-boxes. The S-boxes in CAST-128 have the property of bent functions, that is the degree of a bent function: $GF(2^{2n}) \rightarrow GF(2)$ is at most $n$. Thus, we claim that all the degrees of all S-boxes are at most 8 since the number of input bits is 8. Due to the existences of XOR in a round function $F$, the degrees of all output of $F$ are at most 4.

Moreover, we can compute that an S-box requires about 815 multiplication bit operations. Thus, a straight-line implementation of $F$ evaluation requires $4 \times 815 + 4 \times 32 \approx 2^{11.7}$ bit operations.

ATTACK OF CAST-128-LIKE STRUCTURE

In this section, we review a higher order differential attack of CAST-128-like structure and apply the basic and optimized interpolation attack on it.

Higher Order Differential Attack on CAST-128-like Structure

In 5-round CAST-128-like structure, if the right half of plaintext is fixed value, the degree of the output of right half of the 4-th round as well as the input of the left part of the 5-th round, $L_4(L_0)$, is at most 16 [4]. So the 16-th order differential of $L_4(L_0)$ is independent of the key and becomes constant.

Therefore, we can obtain a system of attack equations given as:

\[ \sum_{L_0 \in V(16)+a} L_4(L_0) = \sum_{L_0 \in V(16)+a} F(R_5(L_0), k^5) \oplus \sum_{L_0 \in V(16)+a} L_5(L_0), \quad (3) \]

where $L_4, L_5, R_5 \in GF(2)^{32}$, $F: GF(2)^{32} \rightarrow GF(2)^{32}$ and $L_0, k^5 \in GF(2)^{32}$, $a \in GF(2)^{32}$ is a constant value. $V(16)$ is a 16-dimensional subspace of $GF(2)^{32}$. 

522
which is the set of all $2^{16}$ possible linear combinations of $(a_1, \ldots, a_{16})$, where each $a_i$ is in $GF(2)^{32}$ and linearly independent. Equation (3) has degree 3 with respect to $\{k_0^{(5)}, k_1^{(5)}, \ldots, k_{31}^{(5)}\}$. $k_{31}^{(5)}$ means the MSB of $k_5$. Then attacker transforms it to a system of linear equations with $M_0$ unknown, where $M_0 = (4 \times C_8^1) + (4 \times C_8^2) + (4 \times C_8^3) = 368$. The system is described as follows:

\[
\begin{bmatrix}
a_{0,0} & a_{0,1} & \cdots & a_{0,M_0-1} \\
a_{1,0} & a_{1,1} & \cdots & a_{1,M_0-1} \\
\vdots & \vdots & \ddots & \vdots \\
a_{31,0} & a_{31,1} & \cdots & a_{31,M_0-1}
\end{bmatrix}
\begin{bmatrix}
k_0^{(5)} \\
k_1^{(5)} \\
\vdots \\
k_{31}^{(5)}
\end{bmatrix}
= \begin{bmatrix}
b_0 \\
b_1 \\
\vdots \\
b_{31}
\end{bmatrix}
\]

Solving above system, we can recover the unknown secret keys with $2^{37.7}$ bit operations and $2^{17}$ chosen plaintexts. For more details about this attack, refer to [4].

**The Basic Interpolation Attack**

In 5-round CAST-128-like structure, if the right half of plaintext is fixed value, we can mount a basic interpolation attack on CAST-128-like structure. First, we re-write equation (3) as:

\[
\sum_{L_0 \in V^{(16)} + a} F(R_5(L_0), k^5) = \sum_{L_0 \in V^{(16)} + a} (L_4(L_0) \oplus L_5(L_0))
\]

Then, we denote $F(R_5(L_0), k^5)$ by $\alpha_u M_u = \alpha_u \prod_{i=1}^n R_i^{w_i}$. This attack equation implies that we can select some 16-dimensional subspaces, then sum over a target bit of $L_4(L_0) \oplus L_5(L_0)$ to compute the constants which are independent of the round key $k^5$. Every such constant can give one information bit, then we will exploit it as the constant value of an equation in the basic interpolation attack.
Note that the equation (4) has degree 4 at most with respect to the right part of the ciphertext \( R_5 = \{c_0, ..., c_{31}\} \), which refers to the algebraic degree of the inverse S-boxes of CAST-128-like structure. Considering the round function \( F \), the input variables of four S-boxes \( S_1, S_2, S_3, S_4 \) are disjointed: i.e., the set of input variables of \( S_1 \) is \( \{c_0, ..., c_7\} \), that of \( S_2 \) is \( \{c_8, ..., c_{15}\} \), that of \( S_3 \) is \( \{c_{16}, ..., c_{23}\} \), and that of \( S_4 \) is \( \{c_{24}, ..., c_{31}\} \). So all the monomials included in equation (4) are products of variables from one of all above sets. We denote \( U \) as the set of coefficient \( \alpha_u \), and \(|U|\) is the number of coefficients \( \alpha_u \). Therefore, the number of unknown variables in the system of linear equations is \(|U| = 4 \times \left( C_8^1 + C_8^2 + C_8^3 + C_8^4 \right) = 648 (\approx 2^{9.3}) \).

In order to obtain enough equations to solve the system of attack equations, we need at least \(|U|\) 16-dimensional subspaces. As usually, we can select about \( e = 2^{9.3} \) arbitrary linearly independent 16-dimensional subspaces, which results in an attack with the data complexity of \( 2^{9.3} \times 2^{16} = 2^{25.3} \). To reduce data complexity, we select a 19-dimensional subspace \( S \), containing \( \binom{19}{16} > 2^{9.9} \) linear independently 16-dimensional subspaces which are sufficient for the attack. The main steps of the basic interpolation attack are described below.

**Offline:**
1. Compute \( \sum_{L_0 \in V^{(16)} + \alpha}(L_4(L_0) \oplus L_5(L_0)) \). We can set the key as zero and consider the term \( \sum_{L_0 \in V^{(16)} + \alpha}(L_4(L_0) \oplus L_5(L_0)) \) as free coefficients of the equations system. By summing on \( L_4(L_0) \oplus L_5(L_0) \) for the \( e \) 16-dimensional subspaces \( S_1, ..., S_e \) in \( S \), we can obtain the free coefficients and store them as an \( e \)-bit array \( B \).
2. Record the \(|U|\) vectors \( \{u|\alpha_u \in U\} \), which are defined by 648 monomials \( M_u \) with respect to \( \{c_0, ..., c_{31}\} \).

**Online:**
1. Store the corresponding right part of the ciphertext \( R_5(L_0) \) in a table after encrypting the \( 2^{19} \) plaintexts in \( S \).
2. For each \( R_5(L_0) \), calculate the value of \( F(R_5(L_0), k^5) \). For each vector \( u \) in \( \{u|\alpha_u \in U\} \), calculate \( M_u \) corresponding to \( R_5 \). Then allocate a \(|U|\)-bit array to store the results. The results of all ciphertexts form a \( 2^{19} \times |U| \) matrix \( A \).
3. For each 16-dimensional subspace \( S_j \) in \( S \), sum the corresponding \( 2^{16} \) rows over \( GF(2) \) to obtain a new row. All rows can be stored in \( ae \times |U| \) matrix \( E \).
4. After obtaining the attack equation system \( Ex = B \), we can solve it using Gauss elimination method. Here \( x \) is the unknown coefficients \( \alpha_u \) of the monomial \( M_u \).
The data complexity of the basic interpolation attack is $2^{19}$ chosen plaintexts. And the time complexity of the attack is dominated by online Step 3. For each 16-dimensional subspace, we sum over $2^{16}$ bits of $|U|$, which requires about $2^{16} \cdot |U| \approx 2^{25.3}$ bit operations. Therefore, the total time complexity is about $e \cdot 2^{16} \cdot |U| \approx 2^{34.6}$ bit operations. We note that this algorithm only recovers the $|U|$ variables $\alpha_u$. The final round key can be further recovered which is not part of our paper.

The Optimized Interpolation Attack

The key idea of an optimized interpolation attack on CAST-128-like is that monomials of higher degree can be transformed into variables of smaller degree [8]. We will describe more details in the following.

Assuming each round key is independent, we can apply a variable transformation technique [8]. We can consider the target bit $b$ as a Boolean polynomial in terms of both key and ciphertext. Considering that $\deg(\alpha_u) + \deg(M_u) \leq \deg(F)$, we can select an index $sp$ which splits the variables $\alpha_u$ into two parts, one part containing variables with less than $sp$ algebraic degree while the other part containing variables with more than or equal to $sp$ algebraic degree. The other part of $\alpha_u$ can be expressed as

$$\alpha_u = \sum_{v=(v_1,\ldots,v_k)|wt(v)\leq d-sp} \beta_v M_v = \sum_{v=(v_1,\ldots,v_k)|wt(v)\leq d-sp} \beta_v \prod_{j=1}^{k} v_j$$

where $d = 4$ is the degree of the round function $F$ of CAST-128-like structure. In order to minimize the variables in terms of key and ciphertext, we set the splitting index $sp$ as 3. Consequently, we obtain that:

$$f(x_1,\ldots,x_n) = \sum_{u=(u_1,\ldots,u_n)|wt(u)<sp} \alpha_u \prod_{i=1}^{n} x_i^{u_i} + \sum_{u=(u_1,\ldots,u_n)|wt(u)\geq sp} \sum_{v=(v_1,\ldots,v_k)|wt(v)\leq d-sp} \beta_v \prod_{j=1}^{k} v_j \prod_{i=1}^{n} x_i^{u_i}$$

In order to calculate the coefficient $\beta_v$ of $M_v$, we evaluate coefficients $\alpha_u$ of monomials of $3 \leq d \leq 4$ and store them in a bit array $a_1$. Then, for each $M_v$, the coefficient $\beta_v$ of $M_v$ in $\alpha_u$ can be computed as follows: Select the subset of keys $\{k|\overline{v} \land k = 0\}$, sum the $2^{wt(v)}$ values in $a_1$ and store the corresponding results in another bit array $a_2$.

Due to the variable transformation, we can reduce the number of unknown variables. Specifically, the monomials $M_u$ having degree 3 or 4 with respect to $\{c_0,\ldots,c_{31}\}$, have been transformed to the monomials $M_v$ of degree 1 or 2. Thus, the number of unknown variables $M_2$ in this system are $M_2 = 4 \times (C_1^1 + C_2^2 + C_3^1 + C_4^2) = 288(\approx 2^{8.2})$. Note that the round key $\{k_0^5,\ldots,k_{31}^5\}$ is XORed with the right part of the ciphertext and are split into four inputs to S-boxes. In analogy
with Step 2 of the basic interpolation attack, we also need a 19-dimensional subspace $S$. The main steps of optimized interpolation attack are given as follows.

**Offline:**
1. Evaluate the target bits of $L_4(L_0) \oplus L_5(L_0)$ corresponding to $|S|$ input plaintexts in subspace $S$, and sum over the 16-dimensional subspaces $S_1, ..., S_e$ by Moebius transform. We can obtain the values of the free coefficients and store them in an $e$-bit array $B$.
2. Record the $|U|$ vectors $\{u|\alpha_u \in U\}$, which are defined by 288 monomials $M_u$ and $M_v$ with respect to $\{c_0, ..., c_{31}\}$ and $\{k_0^5, ..., k_{31}^5\}$ separately.

**Online:**
1. Encrypt the plaintexts in $S$ and store the corresponding $R_5(L_0)$ in a table.
2. Allocate a $|S|$-bit array $a_l$ that stores the $l$-th column of matrix $A$ in the basic interpolation attack. Then allocate an $e \times M_2$ matrix $E$ to represent the equation system. $E$ can be vertically divided into two matrices $E_1$ and $E_2$.
3. For each $R_5(L_0)$, calculate the monomial $M_u$ in $R_5(L_0)$ for each $\alpha_u$, and set the corresponding result in $a_l$.
   a) If $1 \leq \deg(M_u) \leq 2$, we allocate an $e \times M_2/2$ matrix $E_1$, and populate the element $E_1[j][l]$ on the $l$-th column by summing over the $2^{16}a_l$ corresponding to each 16-dimensional subspace $S_j$ by Moebius transform.
   b) If $3 \leq \deg(M_u) \leq 4$, we allocate an $e \times M_2/2$ matrix $E_2$. First we sum over the $2^{16}a_l$ corresponding to each 16-dimensional subspace $S_j$. Then, if the sum is 1, we can populate the element $E_2[j][l]$ by adding the coefficients $\beta_v$ (The method of calculating $\beta_v$ are described above).
4. Solve the Equation System. Two smaller matrices $E_1$ and $E_2$ can be vertically composed a large matrix $E$ of size $e \times M_2$. Later, we obtain the equation system $Ex = B$, where $x$ is the unknown variables $\alpha_u$ or $M_v$. We can solve this system by using Gauss elimination algorithm.
5. Deduce the Secret Key. The monomials $M_v$ of degree 1 are expected as keys.
The optimized attack needs $2^{19}$ chosen plaintexts. Time complexity of the attack is dominated by calculating $M_u$. For each $R_5(L_0)$, calculate the monomial $M_u$ in $R_5(L_0)$ for each $\alpha_u$, which requires about $\log(|S|) \cdot |S| \cdot M_2 \approx 2^{4.2} \cdot 2^{19} \cdot 2^{8.2} = 2^{31.4}$ bit operations. Therefore, the total time complexity of the attack is about $2^{31.6}$ bit operations.

Utilizing the interpolation attack in this section, we can analyze 16-round CAST-128-like structure as well. Fixing the right part of the input, we know the degree of $L_{15}$, the output of the right half of the 15-th round as well as the input of the left part of the 16-th round, is $4^{14} = 2^{28}$. Thus, we can obtain the attack equation as follows:

$$\sum_{L_0 \in V^{(29)}} F(R_{16}(L_0), k^{15}) = \sum_{L_0 \in V^{(29)}} (L_{15}(L_0) \oplus L_{16}(L_0))$$

(7)

Similar with the attack on 5-round structure, the unknown coefficient $M_2 = 288(\approx 2^{8.2})$. So we can select a 31-dimensional subspace S, containing $\binom{31}{29} = 465 \approx 2^{8.8} > 2^{8.2}$ linear independently 29-dimensional subspaces which are sufficient for the attack. Therefore, we can apply the optimized interpolation attack on 16-round CAST-128-like structure and recover the last round keys, which requires $2^{31}$ chosen plaintexts and $\log(|S|) \cdot |S| \cdot M_2 \approx 2^5 \cdot 2^{31} \cdot 2^{8.2} = 2^{44.2}$ bit operations.

CONCLUSIONS

In this paper, we revisited the higher order differential attack of the CAST-128-like structure and found some existing disadvantages on it. Based on the observation, we applied the basic interpolation attack and optimized interpolation attack to the structure respectively. Without increasing the data complexity heavily, our attack reduced time complexity from $2^{37.7}$ bit operations to $2^{31.4}$ bit operations on 5-round CAST-128-like structure. Similarly, we can apply our attack on 16-round structure with $2^{31}$ chosen plaintexts and $2^{44.2}$ bit operations.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (61702331, 61472251, U1536101), National Cryptography Development Fund MMJJ20170105 and Science and Technology on Communication Security Laboratory. The authors would like to thank the editor and the anonymous referees for their helpful comments and suggestions.
REFERENCES