Cryptogr. Commun.
DOI 10.1007/s12095-017-0245-6

Statistical integral attack on CAST-256 and IDEA

Tingting Cui[1,2] · Huaifeng Chen[1] · Long Wen[1] · Meiqin Wang[1,3]

Received: 23 November 2016 / Accepted: 11 July 2017

Abstract Integral attack, as a powerful technique in the cryptanalysis field, has been widely utilized to evaluate the security of block ciphers. Integral distinguisher is based on balanced property on output with probability one. To obtain a distinguisher covering more rounds, an attacker will usually increase the data complexity by iterating through all values of more bits of plaintexts under the firm limitation that the data complexity should be less than the whole plaintext space. In order to release the limitation and reduce the data complexity, Wang et al. proposed a statistical integral distinguisher at FSE'16. In this paper, we exploit the statistical integral distinguisher to attack the IDEA and CAST-256 block ciphers. As a result, we manage to mount a key recovery attack on 29-round CAST-256 with $2^{96.8}$ chosen plaintexts, $2^{219.4}$ encryptions and 2^{73} bytes of memory. By making a trade-off between the time complexity and data complexity, the attack can be achieved by $2^{83.9}$ chosen plaintexts, $2^{244.4}$ encryptions and 2^{66} bytes of memory. As far as we know, these are the best attacks on CAST-256 in the single-key model without weak-key assumption so far. What's more, we find an integral distinguisher of IDEA block cipher, which is the longest integral distinguisher known to now. By taking advantage of this distinguisher, we achieve a key recovery attack on 4.5-round IDEA with $2^{58.5}$ known plaintexts, $2^{120.9}$ encryptions and $2^{46.6}$ bytes of memory respectively. It is the best integral attack with respect to the number of rounds.

This article is part of the Topical Collection on *Recent Trends in Cryptography*

✉ Meiqin Wang
mqwang@sdu.edu.cn

[1] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan 250100, China

[2] Science and Technology on Communication Security Laboratory, Chengdu 610041, China

[3] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China

Published online: 04 August 2017

Keywords Statistical integral attack · IDEA · CAST-256

Mathematics Subject Classification (2010) 94-XX · 94A60

1 Introduction

Integral attack is an important cryptanalytic technique for symmetric-key ciphers, which was originally proposed by Daemen et al. as a dedicated attack against Square cipher [7], thus it is often referred as square attack. Later, Knudsen and Wagner unified it as integral attack [9]. The integral distinguisher used in such attack makes use of the *balanced property* or *zero-sum property* [3]. Explicitly, when fix the value of part of bits in the plaintext and take all possible values for the other part of bits in the plaintext, if the distribution of the value on some specific part of bits in the ciphertext is balanced, i.e. each possible partial value occurs same number of times, we say there is a balance property. If the sum (XOR) of all values of the specific part of ciphertext is zero, we say there is a zero-sum property. Integral attack has been widely applied to many block ciphers. However, its data complexity is determined by taking all values at certain input bits, which is often the bottleneck to pursue a better attack.

In order to reduce the data complexity, Wang et al. proposed a statistical integral distinguisher [17], which applied the statistical technique into the original integral distinguisher. The new distinguisher utilizes the different distributions on the special bits of output between a cipher and random permutation. Let s be the number of input bits that take all possible values at some bits of the input while the other input bits are fixed as a constant, and t be the number of the output bits of which the values are expected to be uniformly distributed. For the integral distinguis, the data complexity is $\mathcal{O}(2^s)$. But with the statistical integral distinguisher, the data complexity can be reduced to $\mathcal{O}(2^{s-t})$, which is beneficial to attack more rounds comparing with the traditional integral attack. In this paper, we exploit the statistical integral distinguisher to attack the CAST-256 and IDEA block ciphers.

1.1 Our contributions

Attacks on CAST-256 CAST-256 is a block cipher designed by Adams at SAC'97 [1], it is one of the AES candidate block ciphers. Since its establishment, many attacks have been proposed, including boomerang attack [15], linear attack [12], differential attack [13], multiple zero-correlation linear attack [16] and multidimensional zero-correlation attacks [5, 6]. The best known cryptanalytic result so far in the single-key model (without weak-key assumption) is the multidimensional zero-correlation linear attack on 29-round (out of 48) CAST-256 given by Chen et al. [6]. In this paper we manage to mount key recovery attacks on 29-round CAST-256. Although we cannot improve the number of attacked rounds, the data complexity and memory requirements are significantly reduced and the time complexity is slightly increased compared with the public attacks on 29-round CAST-256. Particularly, our attacks achieve the minimal data complexity compared with the previous attacks. The attacks on CAST-256 are summarized in Table 1.

Attacks on IDEA IDEA [10] was designed by Lai and Massey in 1991 and is widely used in several security applications, such as IPsec and PGP. Lots of cryptanalysis of IDEA have

Table 1 Summary of attacks on CAST-256

Attack	Rounds	Data	Time	Memory	Weak key rate	Ref.
Boomerang	16	$2^{49.3}$CP	–	–	1	[15]
Linear	24	$2^{124.1}$KP	$2^{156.52}$	–	1	[12]
Differential	36	2^{123}CP	2^{182}	–	2^{-35}	[13]
Multidimensional ZC	28	$2^{98.8}$DKP†	$2^{246.9}$	2^{68}bytes	1	[5]
Multiple ZC*	29	$2^{123.2}$ KP‡	$2^{218.1}$	2^{113}bytes	1	[16]
Multidimensional ZC	29	$2^{123.7}$DKP	$2^{218.1}$	2^{113}bytes	1	[6]
Statistical integral	29	$2^{83.9}$CP	$2^{244.4}$	2^{66}bytes	1	Section 4
Statistical integral	29	$2^{96.8}$CP	$2^{219.4}$	2^{73}bytes	1	Section 4

[1]CP: Chosen Plaintext; KP: Known Plaintext; DKP: Distinct Known Plaintext

[2]†: According to the analysis in [4], the known-plaintext attacks in [5, 6] are actually distinct known-plaintext attacks

[3]*: There is a weak key or independence assumption

[4]‡: The data complexity should be $2^{123.74}$

been discussed. For the integral attacks on IDEA, Nakahara et al. attacked 2.5-round IDEA in [11] and Demirci recovered the key of 4-round IDEA with 2-round integral distinguisher in [8]. In this paper, we find an integral distinguisher for 2.5 rounds of IDEA, which is the longest integral distinguisher known up to now. With this new integral distinguisher, we manage to mount a key recovery attack on 4.5-round IDEA using statistical integral technique, which is the best integral attack on IDEA with respect to the number of rounds. Note that our attack on IDEA is the distinct-known-plaintext attack instead of chosen-plaintext attack in the traditional integral attack. The integral attacks on IDEA are summarized in Table 2.

Outline The statistical integral distinguisher is recalled in Section 2. In Section 3, the specifications of CAST-256 and IDEA block ciphers are described briefly. Then the attacks on CAST-256 and IDEA are proposed in Sections 4 and 5 respectively. Finally the paper is concluded in Section 6.

2 Brief description of statistical integral distinguisher in [17]

In this section, we will recall the statistical integral distinguisher proposed by Wang et al. at FSE'16, and the notions and results about the integral distinguisher follow the description in [17].

Assume that $H : \mathbb{F}_2^n \to \mathbb{F}_2^n$ is a reduced-round block cipher (with a secret key). To be convenient and without loss of generality, one can split the inputs and outputs into two parts each.

$$H : \mathbb{F}_2^r \times \mathbb{F}_2^s \to \mathbb{F}_2^t \times \mathbb{F}_2^u, \ H(x, y) = \begin{pmatrix} H_1(x, y) \\ H_2(x, y) \end{pmatrix}.$$

Table 2 Summary of integral attacks on IDEA

Attack Type	Rounds	Data	Time	Memory	Ref.
Square	2.5	$3 \cdot 2^{16}$CP	$3 \cdot 2^{63} + 2^{48}$	–	[11]
Square	2	23CP	2^{64}	–	[8]
Square	2.5	55CP	2^{81}	–	[8]
Square	3	71CP	2^{71}	–	[8]
Square	3.5	103CP	2^{103}	–	[8]
Square	3	2^{33}CP	2^{82}	–	[8]
Square	3.5	2^{34}CP	2^{82}	–	[8]
Square	4	2^{34}CP	2^{114}	–	[8]
Statistical integral	4.5	$2^{58.5}$DKP	$2^{120.9}$	$2^{46.6}$bytes	Section 5.2

[1]CP: Chosen Plaintext, DKP: Distinct Known Plaintext, –: Not given

Then one can use T_λ to denote the function H_1 where the first r bits of its input are fixed to the value λ and only the first t bits of the output are considered:

$$T_\lambda : \mathbb{F}_2^s \to \mathbb{F}_2^t,\ T_\lambda(y) = H_1(\lambda, y).$$

For an integral distinguisher, if y in the above notation iterates all possible values of $\mathbb{F}_2^s$, then the output value $T_\lambda(y)$ is uniformly distributed. However, this uniform distribution cannot be obtained if the attacker chooses some random values ($< 2^s$) for y. The good side is that in Wang et. al.'s model when considerable quantity of different values of y are chosen, the distribution of quantity of $T_\lambda(y)$ under the right key is distinguished from that under the wrong keys in the key-recovery phase.

Assume that it needs N different values of y to distinguish the above two distributions. A t-bit value $T_\lambda(y) \in \mathbb{F}_2^t$ is computed for each y and one can allocate a counter vector $V[T_\lambda(y)]$, $T_\lambda(y) \in \mathbb{F}_2^t$ and initialize these counters to zero. These counters are used to keep track of the number of each value $T_\lambda(y)$.

Now one can construct an efficient distinguisher by investigating the distribution of the following statistic:

$$C = \sum_{T_\lambda(y)=0}^{2^t-1} \frac{(V[T_\lambda(y)] - N \cdot 2^{-t})^2}{N \cdot 2^{-t}}. \tag{1}$$

This statistic C follows different distributions determined by whether we are dealing with an actual cipher (right key guess) or a random permutation (wrong key guess).

Proposition 1 ([17]) *For sufficiently large N and t, the statistic $\frac{2^s-1}{2^s-N} C_{cipher}$ (C_{cipher} is the statistic C for cipher) follows a χ^2-distribution with degree of freedom $2^t - 1$, which means that C_{cipher} approximately follows a normal distribution with mean and variance*

$$\mu_0 = Exp(C_{cipher}) = (2^t - 1)\frac{2^s - N}{2^s - 1} \text{ and } \sigma_0^2 = Var(C_{cipher}) = 2(2^t - 1)\left(\frac{2^s - N}{2^s - 1}\right)^2.$$

The statistic C_{random} (C_{random} is the statistic C for randomly drawn permutation) follows a χ^2-distribution with degree of freedom $2^t - 1$, which means that C_{random} approximately follows a normal distribution with mean and variance

$$\mu_1 = Exp(C_{random}) = 2^t - 1 \text{ and } \sigma_1^2 = Var(C_{random}) = 2(2^t - 1).$$

To distinguish the two normal distributions with different means and variances, one can compute the data complexity required as follows when error probabilities are given.

Corollary 1 (**Data complexity** [17]) *Under the assumption of Proposition 1, for type-I error probability α_0 (the probability to wrongfully discard the cipher), type-II error probability α_1 (the probability to wrongfully accept a randomly chosen permutation as the cipher), to distinguish a cipher and a randomly chosen permutation based on t-bit outputs when fixing r-bit inputs and randomly choosing distinct values for s-bit inputs, the data complexity can be approximated by*

$$N = \frac{(2^s - 1)(q_{1-\alpha_0} + q_{1-\alpha_1})}{\sqrt{(2^t - 1)/2} + q_{1-\alpha_0}} + 1, \tag{2}$$

where $q_{1-\alpha_0}$ and $q_{1-\alpha_1}$ are the respective quantiles of the standard normal distribution.

Note that this statistic test is based on the decision threshold $\tau = \mu_0 + \sigma_0 q_{1-\alpha_0} = \mu_1 - \sigma_1 q_{1-\alpha_1}$: if $C \leq \tau$, the test outputs 'cipher'. Otherwise, if the statistic $C > \tau$, the test outputs 'random'.

Discussion In the statistical integral model proposed by Wang et al., all values of y are required to be distinct, i.e., the data samples are chosen without replacement. The similar statistical method used in the multidimensional linear attack is presented by Blondeau and Nyberg in [4]. They found that the sampling carried out with replacement or not has a great influence on the distributions for both the actual cipher and random permutations. In the case that the random sampling is without replacement, then the counters $V[j]$, for each possible t-bit value j of T_λ, are independently distributed hypergeometric variates. When the key is correct, the expected distribution of $T_\lambda(y)$ is uniform, that is, the probability for each j is equal to 2^{-t}. But if the key is wrong then, under the randomness assumption, for each wrong key the sample is drawn from a different distribution and the counters follow hypergeometric distribution with a probability which varies with the (wrong) key, but is expected to be 2^{-t}. For the case of sampling with replacement, readers can refer to the work in [4]. In this paper, we only consider to draw the sampling without replacement, i.e., the sampled data values are distinct.

3 Description of CAST-256 and IDEA ciphers

3.1 Description of CAST-256

CAST-256 was designed by Adams [2] and was one of the fifteen candidates in the first round of AES project. It belongs to the CAST family symmetric ciphers which are constructed using the CAST design procedure by Adams [1]. CAST-256 employs the generalized Feistel

structure with four 32-bit branches and uses three different round functions. The block size of CAST-256 is 128 bits and the key size could be 128, 160, 192, 224 and 256 bits. In total, there are 48 rounds consisting of six forward quad-rounds followed by six reverse quad-rounds.

Denote the three different round functions by F_1, F_2 and F_3. Exclusive-or, modulo-2^{32} addition, modulo-2^{32} subtraction and left rotation operations are used in all round functions, and they are denoted as $\oplus$, $\boxplus$, $\boxminus$ and $\lll$ respectively. Four 8×32 S-boxes are used in CAST-256's round functions, denoted as S_1, S_2, S_3 and S_4. F_1, F_2 and F_3 are defined as follows:

$$F_1 : I = (k_m \boxplus I) \lll k_r,\ O = ((S_1[I_1] \oplus S_2[I_2]) \boxminus S_3[I_3]) \boxplus S_4[I_4],$$

$$F_2 : I = (k_m \oplus I) \lll k_r,\ O = ((S_1[I_1] \boxminus S_2[I_2]) \boxplus S_3[I_3]) \oplus S_4[I_4],$$

$$F_3 : I = (k_m \boxminus I) \lll k_r,\ O = ((S_1[I_1] \boxplus S_2[I_2]) \oplus S_3[I_3]) \boxminus S_4[I_4],$$

where I is a 32-bit (4-byte) intermediate variable and O is the 32-bit return value of functions F_1, F_2 and F_3. The four bytes of I is denoted as I_1, I_2, I_3, I_4. Details of round

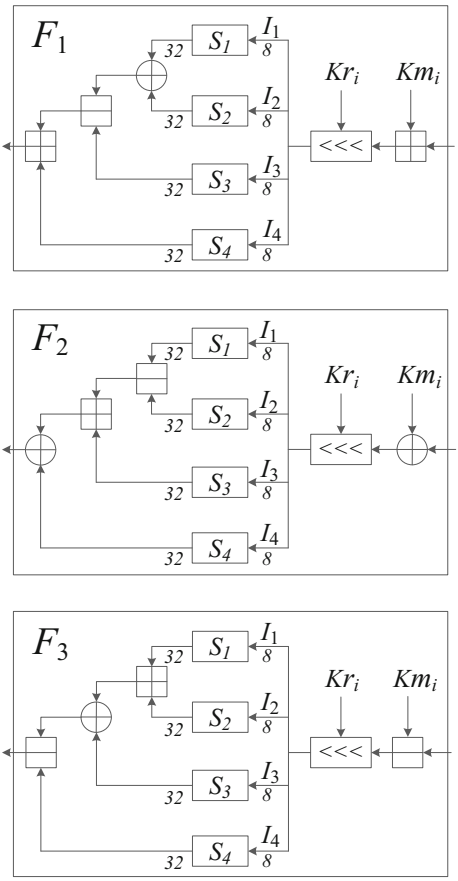

Fig. 1 Round functions of CAST-256

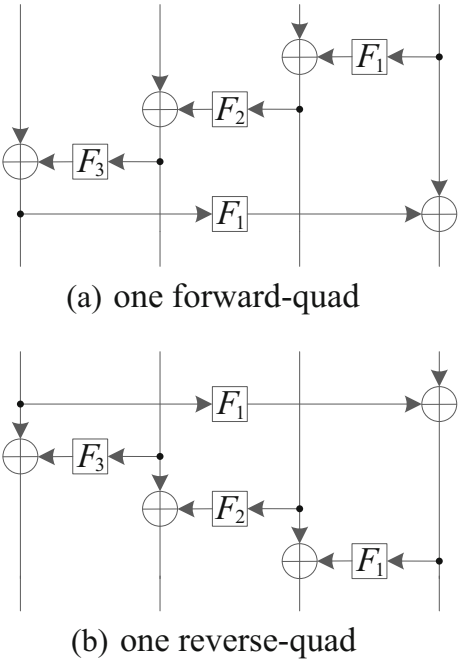

Fig. 2 Forward-quad and reverse-quad of CAST-256

functions F_1, F_2 and F_3 are shown in Fig. 1, where k_r is the 5-bit rotation subkey and k_m is the 32-bit masking subkey for each round. Forward-quad and reverse-quad are shown in Fig. 2.

3.2 Description of IDEA

IDEA is an 8.5-round block cipher with 64-bit state and 128-bit key. The round function uses interleaving operations with 16-bit quantities from different groups: bitwise Exclusive

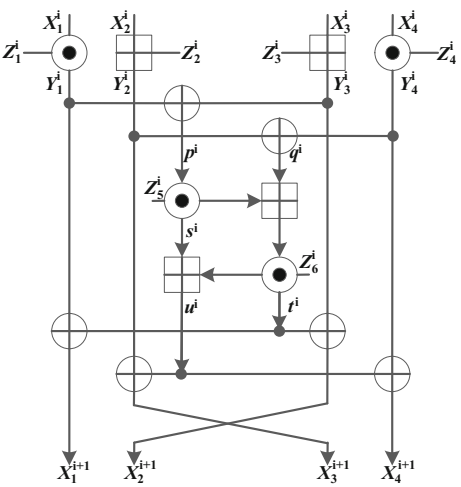

Fig. 3 Round function of IDEA

Table 3 The key schedule of IDEA

Round	Z_1^i	Z_2^i	Z_3^i	Z_4^i	Z_5^i	Z_6^i
$i = 1$	0-15	16-31	32-47	48-63	64-79	80-95
$i = 2$	96-111	112-127	25-40	41-56	57-72	73-88
$i = 3$	89-104	105-120	121-8	9-24	50-65	66-81
$i = 4$	82-97	98-113	114-1	2-17	18-33	34-49
$i = 5$	75-90	91-106	107-122	123-10	11-26	27-42
$i = 6$	43-58	59-74	100-115	116-3	4-19	20-35
$i = 7$	36-51	52-67	68-83	84-99	125-12	13-28
$i = 8$	29-44	45-60	61-76	77-92	93-108	109-124
$i = 9$	22-37	38-53	54-69	70-85		

OR (XOR), addition modulo 2^{16} and multiplication modular $2^{16} + 1$ where the all-zero word $(0x0000)$ in inputs is interpreted as 2^{16} and 2^{16} in output is interpreted as the all-zero word$(0x0000)$. We denote them as $\oplus$, $\boxplus$ and $\odot$, respectively. Each round consists of key mixing (KA) layer (two multiplications and two modular additions) and the multiplication-addition (MA) function (two multiplications, two modular additions), see Fig. 3. We denote input variables of round i by X_i, subkeys of round i by Z_i. Additional input variables are depicted in the outline of a single round in Fig. 3.

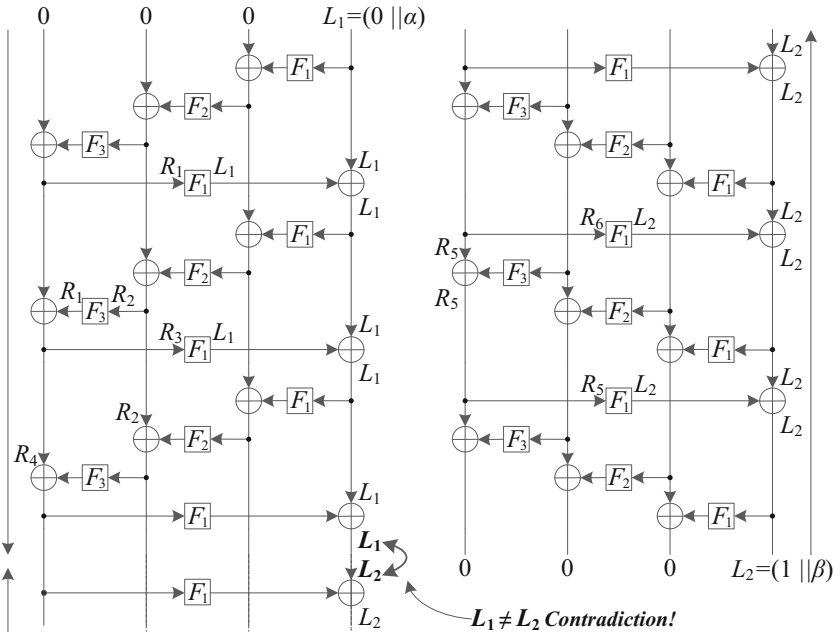

Fig. 4 ZC linear approximations over 24-round CAST-256

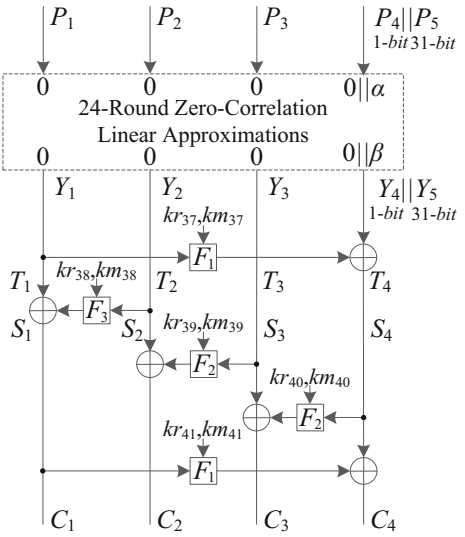

Fig. 5 Key recovery attack on 29-round CAST-256

The key schedule of IDEA is simple and the relation between the subkeys and the master key is listed in Table 3.

4 Statistical integral attack on 29-round CAST-256

4.1 Integral distinguisher for 24-round CAST-256

Bogdanov et al. showed the 24-round zero-correlation (ZC) distinguisher in [5]. If the input mask is $(0, 0, 0, L_1)$ and the output mask after three forward-quads and three reverse-quads is $(0, 0, 0, L_2)$, then the contradiction occurs at the joint of the forward-quad and reverse-quad. This contradiction makes the linear approximations $(0, 0, 0, L_1) \xrightarrow{24r} (0, 0, 0, L_2)$, $L_1 \neq 0$, $L_2 \neq 0$, $L_1 \neq L_2$ zero correlation.

In [5], a transformation method from zero correlation distinguisher to an integral distinguisher has been proposed. However, this 24-round ZC distinguisher for CAST-256 can not be transformed to an integral distinguisher directly with the method in [5] as it does not satisfy the transformation condition of independence between the input mask and the output mask. Sun et al. successfully convert the zero-correlation distinguisher to an integral distinguisher by choosing partial zero-correlation linear approximations in [14]. In order to construct the integral distinguisher, only partial zero-correlation linear approximations are used where $L_1 = 0||\alpha$ and $L_2 = 1||\beta$, $\alpha, \beta \in \mathbb{F}_2^{31}$. These ZC linear approximations can be converted to an integral distinguisher by exploiting the Lemma 1 and Corollary 4 in [14]. The details of the ZC linear approximations we used are illustrated in Fig. 4. In Fig. 4, L_1, L_2 and $R_i, 0 \leq i \leq 6$ are nonzero masks. If we denote the input value and the output value for the distinguisher by P_1, P_2, P_3, P_4, P_5 and Y_1, Y_2, Y_3, Y_4, Y_5, where P_1, P_2, P_3, Y_1, Y_2, $Y_3 \in \mathbb{F}_2^{32}$, P_4, $Y_4 \in \mathbb{F}_2$, P_5, $Y_5 \in \mathbb{F}_2^{31}$, the obtained integral distinguisher

means that if we take all 2^{97} values for P_1, P_2, P_3, P_4 and fix P_5 as a constant, then the output Y_4 and the XOR sum of each bit of Y_5 with Y_4 are balanced.

Next, we will use the integral distinguisher for 24-round CAST-256 to give a statistical integral attack on CAST-256.

4.2 Statistical integral attack on 29-round CAST-256

In this subsection, we will use the above 24-round integral distinguisher to recover the key for 29-round CAST-256 by appending additional five rounds after the distinguisher. The attack is shown in Fig. 5. We consider only seven output bits, consisting of Y_4 and the least significant six bits of Y_5, i.e. $Y_5^i, 0 \leq i \leq 5$, rather than the whole 32 bits to filter the wrong key. Note that in this subsection, superscript numbers are used to denote the bits of a variable. The whole attack is described in Algorithm 1. Here we set $\alpha_0 = 2^{-4}$ and $\alpha_1 = 2^{-39}$, so $q_{1-\alpha_0} \approx 1.5$ and $q_{1-\alpha_1} \approx 7.0$, then the data complexity is about $2^{96.8}$ chosen plaintexts.

Algorithm 1 Key recovery attack on 29-round CAST-256

```
// S_4^{0~5} denotes the least significant six bits of S_4, S_4^{31}
   denotes the most significant bit of S_4
Allocate a counter vector V_1[S_1||S_2||S_4^{31}||S_4^{0~5}].
for all 2^{111} values of kr_41, km_41, kr_40, km_40, kr_39, km_39 do
    Initialize the counter vector V_1 to zero.
    for all 2^{96.8} PC pairs do
        Decrypt three rounds to get S_1, S_2, S_3, S_4.
        Increment the corresponding counter V_1[S_1||S_2||S_4^{31}||S_4^{0~5}] by one.
    // T_4^{0~5} denotes the least significant six bits of T_4,
       T_4^{31} denotes the most significant bit of T_4
    Allocate a counter vector V_2[T_1||T_4^{31}||T_4^{0~5}].
    for all 2^{37} values of kr_38, km_38 do
        Initialize the counter vector V_2 to zero.
        for all 2^{71} values of S_1||S_2||S_4^{31}||S_4^{0~5} do
            Decrypt one round to compute T_1, T_4^{31}, T_4^{0~5}.
            Increment the counter V_2 by
            V_2[T_1||T_4^{31}||T_4^{0~5}] += V_1[S_1||S_2||S_4^{31}||S_4^{0~5}].
        // Y_5^{0~5} denotes the least significant six bits of Y_5
        Allocate a counter vector V_3[Y_4||Y_5^0 ⊕ Y_4||Y_5^1 ⊕ Y_4||···||Y_5^5 ⊕ Y_4].
        for all 2^{37} values of kr_37, km_37 do
            Initialize the counter vector V_3 to zero.
            for all 2^{39} values of T_1||T_4^{31}||T_4^{0~5} do
                Decrypt one round to compute Y_4||Y_5^{0~5}.
                Increment the corresponding V_3 by adding V_2[T_1||T_4^{31}||T_4^{0~5}] to it.
            Compute the statistic C according to (1).
            if C ≤ τ then
                Current value of kr_i, km_i, 38 ≤ i ≤ 41, is a right key candidate.
                Exhaustively search the right key.
```

Algorithm 2 Another key recovery attack on 29-round CAST-256

```
Allocate a counter vector V_1[T_1||T_4].
for all possible values of kr_41, km_41, kr_40, km_40, kr_39, km_39, kr_38, km_38 do
    Initialize the counter vector V_1[T_1||T_4] to zero.
    for all 2^83.9 PC pairs do
        Decrypt four rounds to get T_1, T_4.
        Increment the corresponding counter V_1[T_1||T_4] by one.
    Allocate a counter vector V_2[Y_4||Y_5^0 ⊕ Y4||Y_5^0 ⊕ Y4||···||Y_4 ⊕ Y_5^30].
    for all possible values of kr_37, km_37 do
        Initialize the counter vector V_2[Y_4||Y_5^0 ⊕ Y4||Y_5^0 ⊕ Y4||···||Y_4 ⊕ Y_5^30] to zero.
        for all possible values of T_1||T_4 do
            Decrypt one round to compute Y_4, Y_5.
            Increment the corresponding counter V_2 by adding V_1[T_1||T_4] to it.
        Compute the statistic C according to (1).
        if C ≤ τ then
            The current value of kr_i, km_i, 38 ≤ i ≤ 41, is a right key candidate.
            Exhaustively search the right key.
```

In Algorithm 1, the time complexity of Steps 5 and 6 is $2^{96.8} \cdot 2^{111} \cdot \frac{3}{29} \approx 2^{204.5}$ encryptions. Steps 11 and 12 will take $2^{111} \cdot 2^{37} \cdot 2^{71} \cdot \frac{1}{29} \approx 2^{214.1}$ encryptions. The time complexity of Steps 17 and 18 is $2^{111} \cdot 2^{37} \cdot 2^{37} \cdot 2^{39} \cdot \frac{1}{29} \approx 2^{219.1}$ encryptions. Step 22 takes about $2^{256-39} = 2^{217}$ encryptions. In total, the time complexity for our attack is about $2^{204.5} + 2^{214.1} + 2^{219.1} + 2^{217} \approx 2^{219.4}$ encryptions. The memory requirements are about 2^{73} bytes.

If we use 32-bit output of $Y_4||Y_5$ to sieve the wrong key, the data complexity can be reduced significantly. Here we set $\alpha_0 = 2^{-4}$ and $\alpha_1 = 2^{-14}$, then $q_{1-\alpha_0} \approx 1.5$ and $q_{1-\alpha_1} \approx 3.8$. According to (2), the data complexity is about $2^{83.9}$ chosen plaintexts. The attack process is shown in Algorithm 2. The time complexity of Algorithm 2 is about $(2^{83.9} \cdot 2^{148} \cdot 4 + 2^{64} \cdot 2^{185}) \cdot \frac{1}{29} + 2^{256-14} \approx 2^{244.4}$ 29-round encryptions. The memory requirements are about 2^{66} bytes.

5 Statistical integral attack on 4.5-round IDEA

5.1 Integral distinguisher of IDEA

We identify a 2.5-round integral property for IDEA which is described as follows.

Property 1 As the intermediate state for the i-th round $(Y_1^i, Y_2^i, Y_3^i = Y_1^i \oplus const, Y_4^i)$ takes all 2^{48} possible values, then the set of all corresponding values for the output of the $(2.5+i)$-th round $X_1^{i+3} \oplus X_2^{i+3}$ is balanced.

Proof Figure 6 depicts the integral property. At first, see left part of Fig. 6. $(A_1^i, A_2^i, A_1^i + C, A_3^i)$ for Y^i means that (Y_1^i, Y_2^i, Y_4^i) takes all 2^{48} possible values and $Y_3^i = Y_1^i \oplus const$.

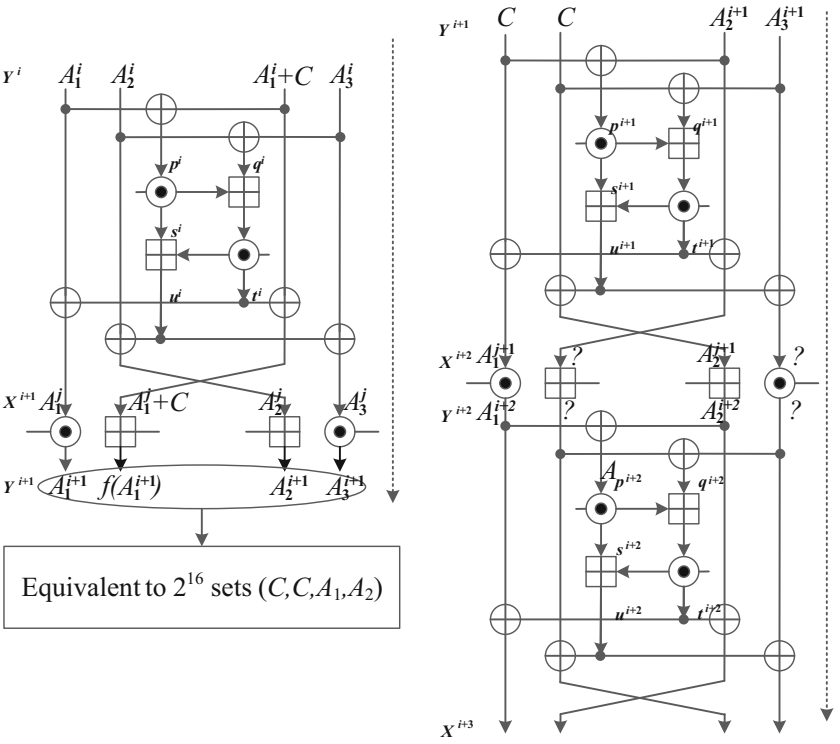

Fig. 6 Integral distinguisher of 2.5-round IDEA

After half round, there are 2^{48} distinct values for X^{i+1} with condition $X_1^{i+1} \oplus X_2^{i+1} = const$. Thus X^{i+1} takes all 2^{48} values in $(A_1^j, A_1^j + C, A_2^j, A_3^j)$. Going through the KA layer of the $(i+1)$-th round, Y_2^{i+1} would be a function of Y_1^{i+1} as follows,

$$Y_2^{i+1} = f(Y_1^{i+1}) = ((Y_1^{i+1} \odot (Z_1^{i+1})^{-1}) \oplus const) \boxplus Z_1^{i+1}.$$

In this way, Y^{i+1} takes all 2^{48} values in $(A_1^{i+1}, f(A_1^{i+1}), A_2^{i+1}, A_3^{i+1})$, which can be divided into 2^{16} sets (C, C, A_1, A_2) according to the value of Y_1^{i+1}.

For the right part of Fig. 6, as Y^{i+1} takes values in one set $(C, C, A_2^{i+1}, A_3^{i+1})$, (p^{i+1}, q^{i+1}) will take all 2^{32} values in (A_2^{i+1}, A_3^{i+1}). As MA is a permutation, the output of $(i+1)$-th MA layer (u^{i+1}, t^{i+1}) will take all 2^{32} values in (A_2^{i+1}, A_3^{i+1}). Then X^{i+2} would take values in set $(A_1^{j+1}, ?, A_2^{j+1}, ?)$, which means (X_1^{i+2}, X_3^{i+2}) takes all 2^{32} values while X_2^{i+2} and X_4^{i+2} have no particular property. After $(i+2)$-th KA layer, (Y_1^{i+2}, Y_3^{i+2}) takes all 2^{32} values in (A_1^{i+2}, A_2^{i+2}). Obviously, the value $Y_1^{i+2} \oplus Y_3^{i+2}$ is balanced. Since $Y_1^{i+2} \oplus Y_3^{i+2} = X_1^{i+3} \oplus X_2^{i+3}$, $X_1^{i+3} \oplus X_2^{i+3}$ is also balanced.

Combining the two parts in Fig. 6, we get the integral property for 2.5-round IDEA. □

5.2 Key recovery attack on 4.5-round IDEA

Using the 2.5-round integral distinguisher from $(Y_1^2, Y_2^2, Y_3^2, Y_4^2)$ to $(X_1^5, X_2^5, X_3^5, X_4^5)$ and the key schedule, we can attack 4.5-round IDEA from the input of the first MA layer $(Y_1^1, Y_2^1, Y_3^1, Y_4^1)$ to the output of round 5 $(X_1^6, X_2^6, X_3^6, X_4^6)$. As illustrated in Fig. 7, we add one round before and append one round after the distinguisher.

Here $s = 48$ and $t = 16$. Set $\alpha_0 = 2^{-2.7}$ and $\alpha_1 = 2^{-10}$, we have $q_{1-\alpha_0} \approx 1.02$ and $q_{1-\alpha_1} \approx 3.10$. Thus we need about $2^{42.5}$ distinct values of $(Y_1^2, Y_2^2, Y_3^2 = Y_1^2 \oplus const, Y_4^2)$. If we take $2^{58.5}$ distinct plaintexts $(Y_1^1, Y_2^1, Y_3^1, Y_4^1)$ randomly, after one round encryption we can obtain $2^{42.5}$ values of $(Y_1^2, Y_2^2, Y_3^2 = Y_1^2 \oplus const, Y_4^2)$, as for some fixed $const$, $Y_1^2 \oplus Y_3^2 = const$ holds with probability 2^{-16}. The key recovery procedure is shown in Algorithm 3.

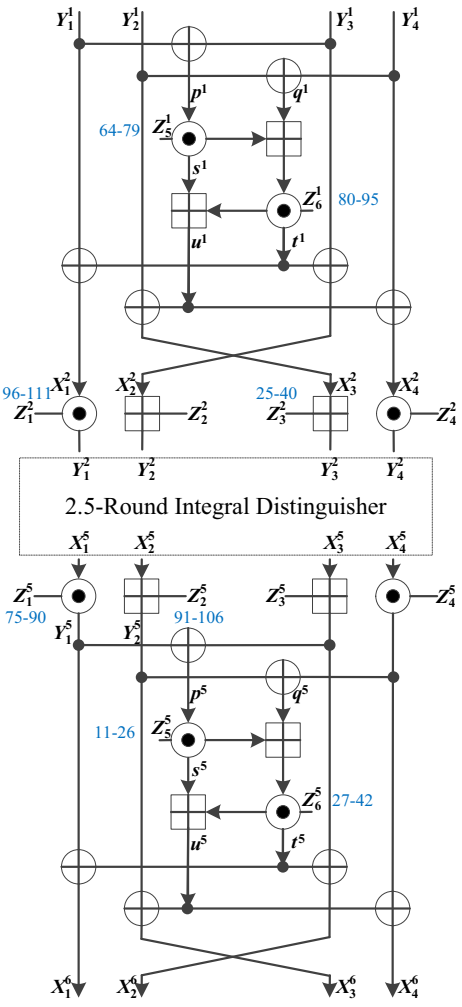

Fig. 7 Attack on 4.5-round IDEA

Algorithm 3 Key recovery attack on 4.5-round IDEA

1 **for** *all* 2^{64} *values of* Z_5^1, Z_6^1, Z_1^2 *and* Z_3^2 **do**
2 Initialize a set $\mathbb{S}$ to empty.
3 **for** *all* $2^{58.5}$ *distinct known plaintexts* $(Y_1^1, Y_2^1, Y_3^1, Y_4^1)$ **do**
4 Compute Y_1^2 and Y_3^2.
5 **if** $Y_1^2 = Y_3^2$ **then**
   `// Set` $const = 0$`. This condition holds with probability` 2^{-16}`.`
6 Store the corresponding ciphertext $(X_1^6, X_2^6, X_3^6, X_4^6)$ in the set $\mathbb{S}$.
7 Allocate a counter vector $V_1[Y_1^5||Y_2^5]$.
8 **for** *all* 2^{16} *values* Z_5^5 *and* Z_6^5 **do**
   `//` $Z_5^5||Z_6^5$ `has 32-bit, but 16 bits of them are derived by` Z_3^2`.`
9 Initialize the counter vector $V_1[Y_1^5||Y_2^5]$ to zero.
10 **for** *all values* $(X_1^6, X_2^6, X_3^6, X_4^6)$ *in* $\mathbb{S}$ **do**
11 Compute Y_1^5 and Y_2^5.
12 Increment $V_1[Y_1^5||Y_2^5]$ by one.
13 Compute Z_1^5 and Z_2^5 from Z_5^1, Z_6^1, Z_1^2.
   `//` $Z_1^5||Z_2^5$ `is a 32-bit string, but all 32 bits are determined by the guessed` Z_5^1, Z_6^1, Z_1^2 `in the upper loop`
14 Allocate and initialize the counter vector $V_2[X_1^5 \oplus X_2^5]$ to zero.
15 **for** *all* 2^{32} *values* $Y_1^5||Y_2^5$ **do**
16 Compute X_1^5 and X_2^5.
17 Increment $V_2[X_1^5 \oplus X_2^5]$ by corresponding $V_1[Y_1^5||Y_2^5]$.
18 Compute the statistic C according to (1).
19 **if** $C \leq \tau$ **then**
   `// The current key value is a right key candidate.`
20 Exhaustively search all right key candidates to find the right key.

Complexity estimation Step 4 needs about $2^{64} \cdot 2^{58.5} = 2^{122.5}$ times of partial encryptions, which is equivalent to $2^{122.5} \cdot \frac{3}{4} \cdot \frac{1}{4.5} \approx 2^{119.9}$ 4.5-round encryptions. Steps 11 and 12 take about $2^{64+16} \cdot 2^{58.5-16} = 2^{122.5}$ times of partial decryptions, which is about $2^{122.6} \cdot \frac{1}{2} \cdot \frac{1}{4.5} \approx 2^{119.3}$ encryptions. Steps 16 and 17 require about $2^{80} \cdot 2^{32} = 2^{112}$ times of partial decryptions, which is equivalent to $2^{112} \cdot \frac{1}{4} \cdot \frac{1}{4.5} \approx 2^{107.8}$ encryptions. As we set the wrong key guess filtration ratio as $\alpha_1 = 2^{-10}$, thus about $2^{128-10} = 2^{118}$ key candidates are exhausted in Step 20. To summarize, the time complexity of our attack on 4.5-round IDEA is about $2^{119.9} + 2^{119.3} + 2^{107.8} + 2^{118} \approx 2^{120.9}$ encryptions. The memory requirements of our attack is about $2^{46.6}$ bytes for the set $\mathbb{S}$ and the data complexity is $2^{58.5}$ distinct known plaintexts.

6 Conclusion

In this paper, we use the statistical integral distinguisher model to attack the block ciphers - CAST-256 and IDEA. As a result, we present the best known attacks on CAST-256 so far in the single-key model without weak-key assumption. Although the number of attacked rounds is not increased, the 29-round CAST-256 attacks are reached with significantly reduced data complexity and memory requirements. What's more, we find the longest integral distinguisher for IDEA and achieve a key recovery attack which is the best integral attack known to now, according the number of rounds.

Acknowledgements This work has been supported by 973 Program (No. 2013CB834205), NSFC Projects (No. 61133013, No. 61572293), Program for New Century Excellent Talents in University of China (NCET-13-0350), Program from Science and Technology on Communication Security Laboratory of China (No. 9140c110207150c11050).

References

1. Adams, C.M.: Constructing symmetric ciphers using the CAST design procedure. In: Kranakis, E., Oorschot, P. (eds.) SAC 1997, pp. 71–104. Springer, US (1997)
2. Adams, C.M.: The CAST-256 encryption algorithm. In: AES Proposal (1998)
3. Aumasson, J.-P., Meier, W.: Zero-sum distinguishers for reduced Keccak-f and for the core functions of Luffa and Hamsi. Presented at the rump session of cryptographic hardware and embedded systems-CHES 2009 (2009)
4. Blondeau, C., Nyberg, K.: Joint data and key distribution of simple, multiple, and multidimensional linear cryptanalysis test statistic and its impact to data complexity. Cryptology ePrint Archive: Report 2015/935. https://eprint.iacr.org/2015/935
5. Bogdanov, A., Leander, G., Nyberg, K., Wang, M.: Integral and multidimensional linear distinguishers with correlation zero. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012, LNCS, vol. 7658, pp. 244–261. Springer, Heidelberg (2012)
6. Chen, H., Cui, T., Wang, M.: Improving algorithm 2 in multidimensional (zero-correlation) linear cryptanalysis using χ^2-method Designs, Codes and Cryptography, pp. 1–18. Springer, Heidelberg (2016)
7. Daemen, J., Knudsen, L.R., Rijmen, V.: The block cipher square FSE 1997. LNCS, vol. 1267, pp. 149–165. Springer, Heidelberg (1997)
8. Demirci, H.: Square-like attacks on reduced rounds of IDEA. In: Nyberg, K., Heys, H. (eds.) SAC 2002, LNCS, vol. 2595, pp. 147–159. Springer, Heidelberg (2003)
9. Knudsen, L.R., Wagner, D.: Integral cryptanalysis. In: Daemen, J., Rijmen, V. (eds.) FSE 2002. LNCS, vol. 2365, pp. 112–127. Springer, Heidelberg (2002)
10. Lai, X., Massey, J.L.: A proposal for a new block encryption standard. In: Damgrd, I.B. (ed.) Eurocrypt 1990, LNCS, vol. 473, pp. 389–404. Springer, Heidelberg (1990)
11. Nakahara, J. Jr., Barreto, P.S.L.M., Preneel, B., Vandewalle, J., Kim, H.Y.: Square attacks against reduced-round PES and IDEA block ciphers. IACR Cryptology ePrint Archive Report 2001/068 (2001)
12. Nakahara, J. Jr., Rasmussen, M.: Linear analysis of reduced-round CAST-128 and CAST-256. SBSEG **2007**, 45–55 (2007)
13. Seki, H., Kaneko, T.: Differential cryptanalysis of CAST-256 reduced to nine quad-rounds. IEICE transactions on fundamentals of electronics communications and computer sciences, (E84-A)4, pp. 913–918 (2001)
14. Sun, B., Liu, Z., Rijmen, V., Li, R., Cheng, L., Wang, Q., Alkhzaimi, H., Li, C.: Links among Impossible Differential, Integral and Zero Correlation Linear Cryptanalysis. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015, LNCS, vol. 9215, pp. 95–115. Springer, Heidelberg (2015)
15. Wagner, D.: The boomerang attack. In: Knudsen, L. (ed.) FSE 1999, LNCS, vol. 1636, pp. 156–170. Springer, Heidelberg (1999)
16. Wen, L., Wang, M., Bogdanov, A., Chen, H.: General application of FFT in cryptanalysis and improved attack on CAST-256. In: Meier, W., Mukhopadhyay, D. (eds.) INDOCRYPT 2014, LNCS, vol. 8885, pp. 161–176. Springer, Heidelberg (2014)
17. Wang, M., Cui, T., Chen, H., Sun, L., Wen, L., Bogdanov, A.: Integrals go statistical: Improved cryptanalysis of skipjack variants. In: Peyrin, T. (ed.) FSE 2016, LNCS, vol. 9783, pp. 399–415. Springer, Heidelberg (2016)