Improved Automatic Search Algorithm for Differential and Linear Cryptanalysis on SIMECK and the Applications

Mingjiang Huang[1,2](✉), Liming Wang[1], and Yan Zhang[1,2]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{huangmingjiang,wangliming,zhangyan}@iie.ac.cn

[2] University of Chinese Academy of Sciences, Beijing, China

Abstract. In CHES'15, Yang et al. proposed a family of lightweight block cipher SIMECK which combines the good designs of SIMON and SPECK. In this paper, we analysis the properties of the round function of SIMECK, and eliminate the repeated use of rotational independence judgment condition in Liu's algorithm that proposed in FSE'17, constructing the partial difference distribution table with limited Hamming weight of input difference to improve the search results. We get new differentials of 14/21/27 rounds for SIMECK32/48/64 which can provide higher probability than previous results, and find a new 28 rounds differential for SIMECK64. We also get new 13/21/27 rounds linear hulls with higher square correlation for SIMECK32/48/64, and we find new 14/22/28 rounds linear hulls for SIMECK32/48/64, which are the best linear hulls of SIMECK as far as we know. With the application of the new distinguishers and combination with the dynamic key-guessing techniques, we mount key recovery attacks on SIMECK variants, which can reduce the computational complexity and/or data complexity.

Keywords: SIMECK · Differential · Linear hull · Cryptanalysis Block cipher

1 Introduction

With the development of the Internet of Things, the security issues in the IoT application systems are getting more and more attention. The cryptographic primitives are the basic components of a security application. The research of lightweight cryptographic algorithms aims at protecting the application security for these terminal devices with limited resources. In recent years, various lightweight block ciphers have been proposed, such as: PRINCE [10], PRESENT [9], TWINE [21], SIMON [5], SPECK [5], SIMECK [23], RECTANGLE [25], GIFT [4], etc.

In 2017, authors of SIMON cited the latest researches of cryptanalysis and gave the explanations on the security of SIMON, but still did not give their own results of SIMON's security analysis [6]. At SAC'17, AlTawy et al. proposed a set of permutation algorithm sLiSCP that based on SIMECK for lightweight sponge cryptographic primitive, used in the sponge framework to construct authenticated encryption, stream cipher, MAC and hash function [2].

At present, there are three mainstream methods to search for the differential/linear distinguishers of SIMON/SIMECK, which are the mixed integer linear programming (MILP) based technique adopted by Sun et al. [20], the SAT/SMT solver method for satisfaction problem solving adopted by Kölbl et al. [12,13], and the Matsui's branch and bound automated search algorithm adopted by Abed et al. [1], Biryukov et al. [8], and Liu et al. [14]. In the MILP method of Sun et al., the non-independece of the input differences are not considered yet. At CRYPTO'15, Kölbl et al. derived the differential propagation relationship for SIMON-like round function, but it takes much time for the SAT/SMT solver to find the optimal differential trails in the large block size variants of SIMON. At FSE'14, Abed et al. firstly searched for the possible output difference corresponding to the input difference in the SIMON-like round function, and took into account the dependence of the input differences on the bitwise AND operation, but they did not find the optimal differential trails. Biryukov et al. introduced the concept of pDDT, and applied the Matsui's approach to the ARX cipher by using the threshold search method, but by the limitation of the heuristic search methods they used, the optimal differential trails are may not obtained. At FSE'17, Liu et al. separately considered the independence and dependence of the inputs of the bitwise AND operation, and they introduced the concept of small block size DDT of bitwise AND operation with independenct inputs, constructing the possible output difference space corresponding to an fixed input difference with a large block size. But in the search algorithm, they reused the independence condition, which will lead more computational efforts. Differential and linear analysis for SIMECK and SIMON are similar. Based on the explicit formula for the differential and linear propagation probability of SIMON-like round function, the optimal differential and linear trails to achieve the security bounds of each variants of SIMON and SIMECK were searched out in [14,15]. For SIMECK32/48/64, the optimal differential trails cover 13/19/25 rounds, and the optimal linear trails also cover 13/19/25 rounds, respectively. In the previous works, the differentials and linear hulls obtained are based on the optimal trails that do not exceed the security boundary, but the differentials and linear hulls based on lower potential trails are not considered. For SIMECK, the probability of the 14/21/27 rounds differentials and the linear square correlation of the 13/21/27 rounds linear hulls are not tight enough.

For the key recovery attacks, 19/26/33 rounds on SIMECK32/48/64 were attacked by differential cryptanalysis in [13], and 22/28/35 rounds were attacked by dynamic key-guessing technique in [17]. However, the differentials that used to attack SIMECK in [13] and [17] are both 13/20/26 rounds, respectively. Intuitively, with the application of differentials with longer rounds and higher probability, the key recovery attack on SIMECK may need less complexity.

In [18], 13/20/26 rounds linear hulls were used to mount the 23/30/37 rounds key recovery attack on SIMECK32/48/64. However, the squared correlation of the linear hulls they used in the attack are derived from the previous differentials, which are estimated values and not tight enough yet. There are also some other analysis results on SIMECK worth attention under different cryptanalysis model, such as the linear cryptanalysis [3], zero correlation linear cryptanalysis [24], and distinguishing attack [19].

Our Contributions. In this paper, we further investigate the differential and linear propagation properties of the non-linear bitwise AND operation of the round function of SIMECK. We propose improved efficient algorithms to find the possible output differences of nonzero probabilities with a fixed input difference, eliminating the repeated use of the independence judgment condition in Liu's algorithm, and constructing a partial difference distribution table with Hamming weight less than a set threshold. We get 14/21/27 rounds differentials for SIMECK with higher probability than the previous works, and find a new 28 rounds differential for SIMECK64. Simultaneously, we get 13/21/27 rounds linear hulls with higher square correlation for SIMECK32/48/64 respectively. And the 14/22/28 rounds linear hulls for SIMECK32/48/64 we find are the best linear hulls so far. We improve the key recovery attack on SIMECK and reduce the computational complexity and/or data complexity. The 29-round differential attack on SIMECK48 is the longest so far.

Outline. This paper is organized as follows. In Sect. 2, we give the notations used in this paper. In Sect. 3, we give improved efficient algorithms to search for the possible nonzero probability output differences with fixed input difference. And we construct all the possible valid input mask space by using the base vectors of input mask introduced in [15]. In Sect. 4, we apply the obtained differentials to key recovery attacks on all variants of SIMECK for reducing the computational complexity and/or data complexity, as listed in Table 5. Conclusions are given in Sect. 5.

2 Preliminaries

2.1 Notations

The main notations used in this paper are shown in Table 1.

Let $P(\alpha, \beta)$ be the probability of a given input difference α propagate to a given output difference β, which is defined as

$$P(\alpha, \beta) = 2^{-n} \cdot \#\{x : f(x) \oplus f(x \oplus \alpha) = \beta\}.$$

Let $f(x) : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ be a vectorial boolean function on n bits with the input mask α and output mask β, we denote by $g(\alpha, \beta) = \sum_{x \in \mathbb{F}_2^n} (-1)^{\alpha \cdot x \bigoplus \beta \cdot f(x)}$, and

Table 1. The notations used in the following paper.

Notation	Description of the notation
$\oplus$	bitwise XOR
$\wedge$	bitwise AND
$\vee$	bitwise OR
$x \lll \mathrm{i}$	rotate x to the left by i bits
(x_i, x_{i-1})	the 2n-bit input state of round i, for $1 \leq i \leq r$
x^j	the j_{th} bit of x, $0 \leq j \leq n-1$
Δx	the difference of $x \oplus x'$
rk_{i-1}	the n bits subkey of round i, for $1 \leq i \leq r$

$g(\alpha, \beta) = \varepsilon \cdot 2^n$ for all $x \in \mathbb{F}_2^n$. Hence, the square correlation of α propagate to β, we denote by

$$C^2(\alpha \to \beta) = \varepsilon^2 = \left(\frac{g(\alpha, \beta)}{2^n}\right)^2.$$

Under the Markov's assumption, the probability of a differential (or linear) trail is the product of the probability of each round. Let α be the input difference, and β is the given output difference after r rounds, then the probability of the r-round differential is the sum of all r-round differential trails with the same input and output difference. Similarly, the square correlation of the r-round linear hull is the sum of all r-round linear trails with the same input and output mask.

2.2 Description of SIMECK

The SIMECK family has 3 variants: SIMECK32/64 (32 rounds), SIMECK48/96 (36 rounds) and SIMECK64/128 (44 rounds). They share the same rotational constant set $(a, b, c) = (0, 5, 1)$, with which SIMECK32/48/64 to achieve full diffusion needs 8/9/11 rounds respectively as investigated in [13]. In this paper, we denote the input states of round i by (x_{i+1}, x_i). The state transformation function of SIMECK can be presented as $x_{i+1} = (x_i \lll a) \wedge (x_i \lll b) \oplus (x_i \lll c) \oplus x_{i-1} \oplus rk_{i-1}$. Let $x_{i+1} = f(x_i) \oplus x_{i-1} \oplus rk_{i-1}$, we generally call $f(x)$ the round function of SIMECK. The differential and linear propagation in the round function of SIMECK is shown in Fig. 1.

3 Automatic Search Algorithm for Differentials and Linear Hulls of SIMECK

3.1 The Properties of SIMECK Round Function

The bitwise AND operation is the only non-linear component in the SIMECK round function, we first study its differential and linear propagation properties.

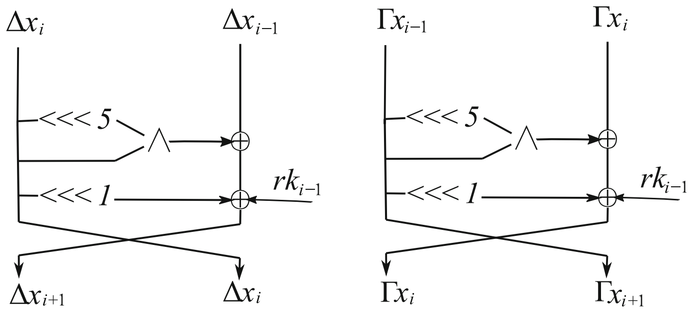

Fig. 1. The differential and linear propagation of the round function of SIMECK.

Table 2. The probability propagation relationship of bitwise AND operation.

α	β	γ	$P(\gamma = 0)$	$P(\gamma = 1)$
0	0	0	1	0
0	1	x	1/2	1/2
1	0	y	1/2	1/2
1	1	$x \oplus y \oplus 1$	1/2	1/2

Property 1. Let x, x', y, y', α, β, $\gamma \in \{0,1\}$, $f(x,y) = x \wedge y$, and $x = x' \oplus \alpha$, $y = y' \oplus \beta$, $\gamma = f(x,y) \oplus f(x',y')$, so α, β, γ have the probability propagation relationship in Table 2. Hence, when $P\{(\alpha,\beta) \rightarrow \gamma\} \neq 0$ is satisfied, if and only if $\overline{\alpha} \wedge \overline{\beta} \wedge \gamma = 0$ is satisfied [1].

According to the definition of differential probability, let α, β, γ be the n bits XOR difference, and x, x', y, $y' \in \mathbb{F}_2^n$, $f(x,y) = x \wedge y$, and $x = x' \oplus \alpha$, $y = y' \oplus \beta$, $\gamma = f(x,y) \oplus f(x',y')$, the probability of two n bits inputs lead to one n bits output is the product of the n probabilities of bitwise operation. As $P\{(\alpha,\beta) \rightarrow \gamma\} = 2^{-2n} \cdot \#\{(x,y)|f(x,y) \oplus f(x \oplus \alpha, y \oplus \beta) = \gamma\}$, which implies the following lemma.

Lemma 1. *Let α, β, γ be the n-bit XOR difference, and x, x', y, $y' \in \mathbb{F}_2^n$, $f(x,y) = x \wedge y$, and $x = x' \oplus \alpha$, $y = y' \oplus \beta$, $\gamma = f(x,y) \oplus f(x',y')$, then*

$$P\{(\alpha,\beta) \rightarrow \gamma\} = \begin{cases} 2^{-wt(\overline{\alpha} \wedge \overline{\beta})}, if\ \overline{\alpha} \wedge \overline{\beta} \wedge \gamma = \mathbf{0}; \\ 0, else. \end{cases}$$

Where $wt(\alpha)$ denotes the Hamming weight of the vector $\alpha \in \mathbb{F}_2^n$, the $\mathbf{0}$ represents an all-zero vector of n bits, and correspondingly $\mathbf{1}$ represents an all-one vector of n bits. When one of the inputs of the bitwise AND is rotated r bits to the left(or right), i.e. $f(x,y) = x \wedge (y \lll r)$, it is easy to get

$$P\{(\alpha,\beta) \rightarrow \gamma\} = \begin{cases} 2^{-wt(\bar{\alpha} \wedge \overline{\beta \lll r})}, if\ \bar{\alpha} \wedge \overline{(\beta \lll r)} \wedge \gamma = \mathbf{0}; \\ 0, else. \end{cases}$$

If the two input values of bitwise AND are mutual rotational dependent of each other, i.e. let x, x', α, $\beta \in \mathbb{F}_2^n$, $x = x' \oplus \alpha$, $f(x) = x \wedge (x \lll r)$, and $\beta = f(x) \oplus f(x \oplus \alpha)$, then the differential probability is

$$P\{\alpha \to \beta\} = 2^{-n} \cdot \#\{x | f(x) \oplus f(x \oplus \alpha) = \beta\},$$

then

$$P\{\alpha \to \beta\} = 2^{-n} \cdot \#\{x | x \wedge (\alpha \lll r) \oplus \alpha \wedge (x \lll r) \oplus \alpha \wedge (\alpha \lll r) \oplus \beta = \mathbf{0}\},$$

and let

$$L_{\alpha,\beta}(x) = \{x \wedge (\alpha \lll r) \oplus \alpha \wedge (x \lll r) \oplus \alpha \wedge (\alpha \lll r) \oplus \beta = \mathbf{0}\}$$

be a set of equations with variable x, where $\alpha, \beta \in \mathbb{F}_2^n$ as parameters. By solving the number of solutions of x, considering these 2^n equations with the relationship between (α, β), Kölbl et al. derived the explicit formula of the differential probability propagation relationship between input and output difference [12]. Therefore, there is an equivalent relationship as shown in Theorem 1. Similarly, the linear square correlation can be defined in Theorem 2 [12].

Theorem 1. *Let α, β be fixed n-bit XOR differences, $x \in \mathbb{F}_2^n$, $x' = x \oplus \alpha$, and $f(x) = x \wedge (x \lll r)$, $\beta = f(x) \oplus f(x')$, define variables $u = (\alpha \lll r) \vee \alpha$, and $v = \alpha \wedge \overline{(\alpha \lll r)} \wedge (\alpha \lll 2r)$, so the differential propagation probability of the bitwise AND with two inputs mutual rotational dependent is*

$$P\{\alpha \to \beta\} = \begin{cases} 2^{-n+1}, if\ \ \alpha = \mathbf{1}, and\ wt(\beta) \equiv 0 \bmod 2; \\ 2^{-\mathrm{wt}(v \oplus u)}, if\ \ \alpha \neq \mathbf{1}, and\ \beta \wedge \overline{v} = \mathbf{0}, and\ (\beta \oplus (\beta \lll r)) \wedge u = \mathbf{0}; \\ 0, else. \end{cases}$$

Theorem 2. *Let α, β be an input and an output mask, $f(x) = x \wedge (x \lll r)$, where $x \in \mathbb{F}_2^n$, and the set U_β is defined by $U_\beta = \{x | (\beta \wedge (x \lll r)) \oplus ((\beta \wedge x) \ggg r) = \mathbf{0}\}$, the dimension of U_β is $d = dim\ U_\beta$, and $U_\beta^\perp$ is the orthogonal complement space of U_β. Hence, the squared correlation calculation of $f(x)$ that α propagate to β can be denoted by*

$$C^2(\alpha, \beta) = \begin{cases} 2^{-n+2}, if\ \ \beta = \mathbf{1}, and\ \alpha \in U_\beta^\perp; \\ 2^{-n+d}, if\ \ \beta \neq \mathbf{1}, and\ \alpha \in U_\beta^\perp; \\ 0, else. \end{cases}$$

Corollary 1. *Let α, β be fixed n-bit XOR differences, x, $x' \in \mathbb{F}_2^n$, $x' = x \oplus \alpha$, $f(x) = x \wedge (x \lll r)$, and $\beta = f(x) \oplus f(x')$, so the probability $P(\alpha \to \beta)$ is only related to the value of the input difference α, and the differential probability corresponding to each valid output difference β is equivalent. And the number of valid output differences is equal to $^1/_{P\{\alpha \to \beta\}}$, where $P\{\alpha \to \beta\} \neq 0$.*

Corollary 2. *Let $\{\Delta x_1, \Delta x_0\}$ be the 2n-bit input XOR difference, $\{\Delta x_{i+1}, \Delta x_i\}$ is the 2n-bit output XOR difference of i round differential of SIMECK2n, and the probability is P. There exist other $n-1$ differentials with similar probability, the input difference and output difference can be constructed as $\{\Delta x_1 \lll j, \Delta x_0 \lll j\}$ and $\{\Delta x_{i+1} \lll j, \Delta x_i \lll j\}$ respectively, for $j \in [1, n-1]$.*

3.2 Improved the Differentials of SIMECK Variants

When the inputs of the bitwise AND are independent of each other, i.e. let $f(x, y) = x \wedge y$, the DDT can be constructed directly according to Lemma 1, as shown in Algorithm 1. If the inputs of the bitwise AND are mutually rotational dependent, i.e. let $f(x) = x \wedge (x \lll r)$, the difference distribution table of bitwise AND can be deduced by Theorem 1. Algorithm 2 constructs the nonzero probability difference distribution table of the bitwise AND with inputs rotational dependent. When the block size n is not too large, generally when $n \leq 16$, the difference distribution table (DDT) generated by it can be storaged feasibly. The storage size of the DDT of the bitwise AND operation with input rotational dependent can be reduced from 2^{2n} to $2^n \cdot \mathcal{O}$, when n=16, $\mathcal{O} \approx 2^{8.62}$.

Algorithm 1. Given two m-bit input difference α, β, constructing the nonzero probability difference distribution table $DDT_m(\alpha, \beta)$.

```
1: for each $\alpha, \beta = 0$ to $2^m - 1$ do
2:     cnt = 0;
3:     for $\gamma = 0$ to $2^m - 1$ do
4:         if $\gamma \wedge (\overline{\alpha} \wedge \overline{\beta}) = \mathbf{0}$ then
5:             $beta[\alpha||\beta][cnt++] = \gamma$;
6:         end if
7:     end for
8:     $p[\alpha||\beta] = 1/cnt$;
9: end for
```

When the inputs of the bitwise AND are mutually rotational dependent and the block length is larger than 16 bits, it will be difficult to store the large DDT produced by Algorithm 2 directly. We are inspired by Liu et al., in order to reduce the storage complexity, consider the independent and dependent conditions of the inputs separately. Firstly, assuming that the inputs are independent of each other, referring to Algorithm 1, a large block of blocksize n can be splited into several small blocks with m bits length each, and the possible output differences corresponding to the fixed input difference are reconstructed by the output of small blocksize DDT whose inputs are mutually independent. For example, the blocksize of a cipher is $2n = 64$, let $m = 8$ in Algorithm 1, the input of round function with $n = 32$ bits length can be splited into four 8-bit blocks, then for each block just lookup the DDT produced by Algorithm 1, and then recombine the 4 sets of 8-bit possible output difference which construct the 32 bits length possible output differences. Secondly, verifing whether the recombined output differences are valid possible output differences with nonzero probability or not through the judgment condition in Theorem 1, and considering the constraints of output difference and input inter dependence. Here, for filtering out the possible output differences of nonzero probability, we eliminate the repeated use of input independent condition (i.e. $\beta \wedge \overline{(\alpha \vee \alpha \lll r)} = \mathbf{0}$) in Liu's algorithm. Thirdly, considering the increasement in the Hamming weight of the input difference α,

Algorithm 2. Computing the difference distribution table $DDT_n(\alpha)$ when the inputs are mutually dependent.

```
$p[2^n] = \{0\}$, $\beta\_value[2^n][\,] = \{0\}$, $\beta\_num[2^n] = \{0\}$;
for $\alpha = 0$ to $2^n - 2$ do
    $u = \alpha \vee \alpha \lll 5$;
    $v = \alpha \wedge \overline{\alpha \lll 5} \wedge \alpha \lll 10$;
    for $\beta = 0$ to $2^n - 1$ do
        if $(\bar{u} \wedge \beta) \vee (v \wedge (\beta \oplus \beta \lll 5)) = \mathbf{0}$ then
            $\beta\_value[\alpha][\beta\_num[\alpha]] = \beta$;
            $\beta\_num[\alpha] + +$;
        end if
    end for
    $p[\alpha] = 1/\beta\_num[\alpha]$;
end for
let $\alpha = 2^n - 1$;
for $\beta = 0$ to $2^n - 1$ do
    if $wt(\beta) \equiv 0 \bmod 2$ then
        $\beta\_value[2^n - 1][\beta\_num[2^n - 1]] = \beta$;
        $\beta\_num[2^n - 1] + +$;
    end if
end for
$p[2^n - 1] = 1/\beta\_num[2^n - 1]$;
return $p[\,]$, $\beta\_value[\,]$;
```

the probability of the corresponding output difference decrease, shown in [7,14]. When the Hamming weight of the input difference increases, the upper bound of the probability of the round function will also decrease, which lead to the Matsui's pruning condition in the search procedure will not be satisfied mostly. The output differences and probabilities corresponding the low Hamming weight input difference are used frequently. Inspired by the concept of partial difference distribution table introduced by Biryukov et al. [8], we can just precompute and store the difference distribution table corresponding to the input difference with low Hamming weight, where the Hamming weight is less than a certain threshold. When the input difference Hamming weight is smaller than the set threshold, just look up the precomputed table, otherwise, calculating the possible output difference from $POD_n(\alpha)$ in Algorithm 3.

To search for the differentials of SIMECK, we firstly employ the Matsui's branch-bound search approach similar to [14] for finding the optimal differential characteristics(trails). Try to search for longer rounds of differential trails that exceed security bound with limitting the Hamming weight of the input difference according to the differential probability upper bound of the input Hamming weight [7,14]. Afterward, we fix the input difference α and output difference β to find enough amount of trails, and statistic the probability of all trails. In order to effectively search for trails as much as possible within a feasible time, we limit the search range of probability weights$(-\log_2(p))$ from $wt_{\min}$ to $wt_{\max}$. And $wt_{\min}$ is the probability weight of the differential characteristic, $wt_{\max}$ is the probability

Algorithm 3. Given a n-bit input difference α, computing the set $POD_n(\alpha)$ of possible output differences β with nonzero probability.

```
Input: $\alpha \in \mathbb{F}_2^n, n = mt$;
1: p = 0, $\beta\_value[\,] = \{0\}, \beta\_num = 0$;
2: if $\alpha \neq 2^n - 1$ then
3:     let $X=\alpha, Y=\alpha \lll 5$;
4:     divide $X = \{x_{t-1}, ...x_0\}, Y = \{y_{t-1}, ...y_0\}, x_i, y_i \in \mathbb{F}_2^m$;
5:     for $i = 0$ to t - 1 do
6:         lookup $DDT_m(x_i, y_i)$; //Computed by Algorithm 1.
7:         $\beta^i[cnt^i] = beta[x_i, y_i][cnt^i] \in \{0,1\}^m$
8:     end for
9:     for each $\beta := \{\beta^{t-1}[cnt^{t-1}]||...\beta^0[cnt^0]\}$ do
10:        if $(\beta \oplus \beta \lll 5) \wedge (\alpha \wedge \overline{\alpha \lll 5} \wedge \alpha \lll 10) = \mathbf{0}$ then
11:            $\beta\_value[\beta\_num++] = \beta$;
12:        end if
13:    end for
14: else
15:    for $\beta = 0$ to $2^n - 1$ do
16:        if $wt(\beta) \equiv 0 \bmod 2$ then
17:            $\beta\_value[\beta\_num++] = \beta$;
18:        end if
19:    end for
20: end if
21: p = $1/\beta\_num$;
22: return p, $\beta\_value[\,]$;
```

weight of the minimal probability that be limited. In the first two rounds of the search process, using the left and right part of the input difference as the inputs of round function. During the branch search process of differential trails of 3 to $r-1$ rounds, searching the possible output differences corresponding to the input difference by lookup table generated by Algorithm 4, when the Hamming weight of input difference is less than H. Otherwise utilizing the Algorithm 3 to generate the possible output differences. In the process of the middle rounds, we use Matsui's branch pruning condition $wt(p_1)+wt(p_1)+...+wt(p_i)+wt(p_{r-i}) \leq wt_{\max}$ as the stop condition, when the probability of searched truncated path is larger than the set value, remove it in advance. Even so, there are still some trails in the front $r-1$ round maybe satisfying the brach pruning condition, but when multiplied by the differential probability of the last round, the probability weight will larger than the limited $wt_{\max}$, while they are also propagate to the same output difference of the differential. In our statistical approach, these part trails with lower probability are also counted. Probability weight marked by * in Table 3 means there are some trails with lower probability than the set probability weight $wt_{\max}$ be counted. The differential probability is calculated by $P = \sum_{w=wt_{\min}}^{wt_{\max}} \#trails[w] \times 2^{-w}$.

Algorithm 4. Pre-calculating and store the partial difference distribution table $PDDT_{n,H}(\alpha)$ with Hamming weight less than H.

```
1: for $wt(\alpha) = 0$ to $H$ do
2:     $POD_n(\alpha)$;
3:     $\beta[\alpha][\,] \leftarrow \beta\_value[\,]$;
4:     $p[\alpha] = p$;
5: end for
```

Table 3. The Differentials of SIMECK variants.

Cipher	Rd	Δin	Δout	$wt_{\min}$	$wt_{\max}$	Prob	#Trails	Ref
SIMECK32	13	0,2	2,0	38	52	$2^{-28.91}$	1846518	[18]
	13	8000,4011	4000,0	36	49	$2^{-27.28}$	/	[13]
	14	1,800A	4,8002	/	/	$2^{-31.64}$	/	[14]
	14	2,15	8,5	36	57	$2^{-31.63}$	8065284	This paper
	14	0,2	4,2	40	53	$2^{-30.90}$	1678405	This paper
SIMCEK48	21	20000,470000	50000,20000	/	100	$2^{-45.65}$	/	[13]
	21	1,800002	800002,1	/	/	$2^{-45.28}$	/	[14]
	21	2,5	5,2	52	73	$2^{-45.18}$	34899905	This paper
SIMECK64	26	0,4400000	8800000,400000	/	121	$2^{-60.02}$	/	[13]
	27	0,10	5,2	/	/	$2^{-61.49}$	/	[14]
	27	0,11	5,2	70	89*	$2^{-60.75}$	32649265	This paper
	28	0,11	A8,5	74	93*	$2^{-63.91}$	617703755	This paper

For a differential of i rounds, take the output difference of the i_{th} round as the input difference of $(\mathrm{i}+1)_{th}$ round, according to Algorithm 3, one more round can be extended by check the number of output difference. For every valid possible output difference of $(\mathrm{i}+1)_{th}$ round, probability of the new $i+1$ round differential can be calculated by Corollary 1. The obtained differentials for SIMECK is shown in Table 3, in which the number of possible output difference corresponding to the output difference of the 13-round differential $(0x0, 0x2 \rightarrow 0x2, 0x0)$ used in [17] of SIMECK32 is 4. Hence, there exists 14 rounds differential of probability at least $\frac{1}{4} \times 2^{-28.91}$, the input difference of it is $(0x0, 0x2)$ and the output difference is one of $\{(0x4, 0x2), (0x6, 0x2), (0x44, 0x2), (0x46, 0x2)\}$, and the searched experimental result of the probability is $2^{-30.90}$. In addition, the result of 27/28 round differential of SIMECK64 that also confirmed the guesswork of Corollary 1. Additionally, we also searched out some differentials that follow Corollary 2, such as the 14-round differential $(2, 15) \rightarrow (8, 5)$, which is the rotational pair of $(1, 800A) \rightarrow (4, 8002)$ that rotated 1-bit to the left for each half state[1]. More differentials can be constructed from Table 3 by the Corollary 2, for example, the 14-round differential of SIMECK32 implies $((0, 2) \rightarrow (4, 2)) \Rightarrow ((0 \lll j, 2 \lll j) \rightarrow (4 \lll j, 2 \lll j))$, $j \in [1, 15]$ with the similar probability that larger than $2^{-30.90}$. And the derivation process for SIMECK48/64 is similar. The obtained 14/21/27 rounds differential of SIMECK32/48/64 are the best so far, meanwhile, the 28-round differential of SIMECK64 is the longest so far.

[1] The reason why the experimental result of the probability is higher than that of [14] is because of more trails are counted.

Algorithm 5. Using the non-zero base vectors of α to construct the space of all vaild possible α.

```
Input: Nonzero vectors of $y[i]$ construct the bases of $U_\beta^\perp$, $i \in [0, n-1]$.
 1: $\alpha_{num} = 0, \alpha[\,] = \{0\}, z[\,] = \{0\}, j = 0$;
 2: for $i = 0$ to $n-1$ do
 3:     if $y[i] \neq \mathbf{0}$ then
 4:         $z[j] = y[i]$;
 5:         $j++$;   //Record the number of non-zero base vectors.
 6:     end if
 7: end for
 8: $\alpha_{num} = 2^j$;
 9: for $k = 0$ to $\alpha_{num} - 1$ do   //Construct each valid $\alpha$.
10:     for $t = 0$ to $j-1$ do
11:         if $(k \wedge (1 \ll t)) \neq \mathbf{0}$ then
12:             $\alpha[k] = \alpha[k] \oplus z[t]$;
13:         end if
14:     end for
15: end for
16: return $\alpha_{num}, \alpha[\,]$;
```

3.3 Improved the Linear Hulls of SIMECK Variants

In [15], an automatic search algorithm to search for the optimal linear trails of SIMON and SIMECK are proposed. They gave an algorithm to find the base of all possible input masks $\alpha \in U_\beta^\perp$ in Theorem 2. By using the base vectors of α, we can construct all the possible valid input masks by Algorithm 5. Then, we use the method in [15] and search for longer round linear trails with the square correlation exceed the security boundary. Take the input and output mask of the trails as that of the linear hull, and we limit the search scope by the lower bound of the square correlation $(-log_2 C^2(\alpha, \beta))$. Also, setting the branch pruning condition $wt(p_1) + wt(p_1) + ... + wt(p_i) + wt(p_{r-i}) \leq wt_{\max}$ as the stop condition. By using the longer round linear trails with square correlation exceed the security boundary, the new longer round linear hulls are found in Table 4. The 14/22/28 rounds linear hulls of SIMECK32/48/64 are the longest linear hulls so far[2].

4 Key Recovery Attack on Round Reduced SIMECK

In 2014, Wang et al. proposed dynamic key-guessing techniques in differential attack on SIMON [22], and then these techniques had also been used to achieve linear hull attack on SIMON at FSE'16 [11]. The differential and linear hull attack on SIMECK with dynamic key-guessing techniques are the most efficient method until now [17,18]. Hence, by applying the new differentials obtained, we

[2] All experiments code are runned on a PC with Intel® Core^{TM} i7-2600 CPU@3.40GHz $\times$ 8.

Table 4. The Linear hulls of SIMECK variants.

Cipher	Rd	Γin	Γout	$wt_{\min}$	$wt_{\max}$	Potential	#Trails	Ref
SIMECK32	13	2,0	0,2	40	/	$2^{-30.91}$	1846518	[18]
	13	11,0	2,15	/	/	$2^{-29.43}$	/	[15]
	13	22,0	4,2A	32	38*	$2^{-28.11}$	876	This paper
	14	800A,1	8008,4004	36	42*	$2^{-31.90}$	1456	This paper
SIMECK48	20	280000,100000	200000,100000	/	/	$2^{-45.66}$	/	[18]
	21	800002,1	1,800002	/	/	$2^{-46.30}$	/	[15]
	21	880002,1	1,800002	52	64*	$2^{-44.48}$	352999	This paper
	22	800000,1	200000,500001	56	68*	$2^{-47.68}$	1326121	This paper
SIMECK64	26	440000,0	400000,200000	/	/	$2^{-62.09}$	/	[18]
	27	11,0	8,14	/	/	$2^{-61.14}$	/	[15]
	27	22,0	10,28	70	86*	$2^{-59.79}$	27489363	This paper
	28	80000001,5	0,40000004	74	88	$2^{-63.67}$	9103911	This paper

try to get some new results for the key recovery attacks on SIMECK variants. And for the attack on SIMECK with the new linear hulls obtained, better results with less complexity may also occur combining with dynamic key-guessing techniques in linear hulls attack. Even so, we leave it as subsequent researchs because the attack process is too cumbersome, it is recommended to refer to [11] for the process details of using linear hulls for key recovery.

4.1 Dynamic Key-Guessing in Differential Attack

In [17,22], the dynamic key guessing technique was introduced to the differential attack on SIMON and SIMECK. By observing the round function of SIMECK, the bit-transformation relationship of it is denoted by follows. Let $x_i = \{x_i^{n-1} x_i^{n-2} \cdots x_i^j \cdots x_i^0\}$, $x_i^j \in \{0,1\}$, there have,

$$x_{i+1}^j = x_i^j \wedge x_i^{j-5} \oplus x_i^{j-1} \oplus x_{i-1}^j \oplus rk_{i-1}^j.$$

The bits relationship of the differential propagation of the SIMECK round function is expressed as follows:

$$\Delta x_{i+1}^j = \Delta x_i^j \wedge x_i^{j-5} \oplus x_i^j \wedge \Delta x_i^{j-5} \oplus \Delta x_i^j \wedge \Delta x_i^{j-5} \oplus \Delta x_i^{j-2} \oplus \Delta x_{i-1}^j$$

and the plaintext bits x_i^j and x_i^{j-5} involved secret subkey bits for $i \geq 2$.

$$x_i^j = x_{i-1}^j \wedge x_{i-1}^{j-5} \oplus x_{i-1}^{j-1} \oplus x_{i-2}^j \oplus rk_{i-1}^j$$

$$x_i^{j-5} = x_{i-1}^{j-5} \wedge x_{i-1}^{j-10} \oplus x_{i-1}^{j-6} \oplus x_{i-2}^{j-5} \oplus rk_{i-1}^{j-5}$$

As the choosen plaintexts are known, considering Δx_i^j and Δx_i^{j-5} as parameters under different values, and discussing the number of subkey bits satisfy the equations. When some subkey bits are determined, which can reduce the number of the remaining subkey bits that need to be exhaustive searched.

If $(\Delta x_i^j, \Delta x_i^{j-5}) = (0,0)$, $\Delta x_{i+1}^j = b$, $b \in \{0,1\}$, when $\Delta x_i^{j-2} \oplus \Delta x_{i-1}^j \neq b$, the differential propagation relationship of round function is not satisfied, this condition can be used to filter out the wrong plaintext pairs in the first and last round. When $\Delta x_i^{j-2} \oplus \Delta x_{i-1}^j = b$, the bit differential propagation relationship can not be used for determining subkey bits, so $(rk_{i-1}^j, rk_{i-1}^{j-5}) \in \mathbb{F}_2^2$ with 4 solutions. And if $(\Delta x_i^j, \Delta x_i^{j-5}) = (0,1), (1,0)$ *or* $(1,1)$, then $(rk_{i-1}^j, rk_{i-1}^{j-5})$ have only 2 solutions. After appending r_0 rounds forward and r_1 rounds backward, generate the extended path with sufficient bit conditions, from which the related subkey bits for each sufficient bit condition can be solved by whether to satisfy the equations.

4.2 Applying the Differentials to Key Recovery Attack on SIMECK Variants

In [17,22], to extend the differential path according to the rules that the output differences of AND operation is 0, if and only if its input differences are (0,0), otherwise set the output difference of AND operation to * as uncertain bit. However, the rotational dependence of the input differences of AND operation is not considered in these previous works. To extend the differential path with adding r_0 round forward and r_1 round backward, we use the nonzero probability outputs produce by Algorithm 3. In the data collection phase, there are Q_{r_0} possible plaintext differences in the set $\{\Delta_P\}$, which will lead to at least 1 input difference Δ_{in} of the r-round truncated differential that be appended r_0 round forward. There are Q_{r_1} possible ciphertext output differences in the set $\{\Delta_C\}$, which can be deduced from 1 output difference Δ_{out} of the r-round truncated differential that be appended r_1 round backward. The number of possible plaintext difference Q_{r_0} is less than the plaintext pairs in each data structure constructed in [22]. Selecting one plaintext X, then the plaintexts set $\{X \oplus \Delta_P\}$ obtained by Q_{r_0} XOR additions which can yield Q_{r_0} plaintext pairs which lead to at least one input difference Δ_{in}. Choosing 2^t arbitrary plaintext X, for example $X \in \{0,1\}^t$, then we can get $2^t \times Q_{r_0}$ plaintext pairs that lead to 2^t paris with intermediate state in $(r_1)_{th}$ round with Δ_{in}. Since the set of ciphertext differences resulting from the output difference of choosen differential, there exists conditions that some bit positions of the ciphertext difference are fixed, which can be used to filter out the invalid plaintext differece in $\{\Delta_P\}$, so does the invalid plaintexts, and then check the remaining ciphertext belong to $\{\Delta_C\}$ or not, which can reduce the storage complexity.

For SIMECK32/64, similarly to the attack in [17], we use the 14-round differential $\Delta in : 0x0000\ 0002 \rightarrow \Delta out : 0x0004\ 0002$ with the probability of $2^{-30.90}$ in Table 3, and extended 4 rounds forward and 4 rounds backward of it to mount the key recovery attack against the 22 rounds of SIMECK32/64. The extended differential path with sufficient bit conditions listed in Table 6.

In the data collection phase, we choose 2^{32} plaintexts, with $2^{32} \times Q_{r_0} \approx 2^{32+21.3} = 2^{53.3}$ times XOR addition, $2^{53.3}$ plaintext pairs can be get, which will lead to $2^{32} \times p = 2^{32-30.90} \approx 2.14$ right pairs occured in average. For the choosen

Table 5. Comparison of Differential Cryptanalysis on SIMECK.

Cipher	Total Rounds	Diff. Rounds	Diff. Prob	Attacked Rounds	Time	Data	Ref
SIMECK32	32	13	$2^{-27.28}$	19	2^{40}	2^{31}	[13]
		13	$2^{-29.64}$	22	$2^{57.9}$	2^{32}	[17]
		14	$2^{-30.90}$	22	$2^{54.3}A^{a} + 2^{56}E^{b}$	2^{32}	This paper
SIMECK48	36	20	$2^{-43.65}$	26	2^{62}	2^{47}	[13]
		20	$2^{-43.65}$	28	$2^{68.3}$	2^{46}	[17]
		21	$2^{-45.18}$	29	$2^{77.04}A + 2^{83.14}E$	2^{47}	This paper
SIMECK64	44	26	$2^{-60.02}$	33	2^{115}	2^{63}	[13]
		26	$2^{-60.02}$	35	$2^{116.3}$	2^{63}	[17]
		27	$2^{-60.75}$	35	$2^{90.6}A + 2^{105.5}E$	2^{62}	This paper

[a] ‘A’ represents a XOR addition operation.
[b] ‘E’ represents an encryption of attacked rounds

Table 6. Extended Differential Path of 22-round SIMECK32 with sufficient conditions.

Rd	Input difference of each round represented in bits	Q_{r_0}
0	0,0,0,*,*,0,0,*,*,*,0,1,*,*,*,*,0,0,*,*,*,0,*,*,*,*,*,*,*,*,*,*	2589180
1	0,0,**0**,**0**,*,0,**0**,**0**,*,*,**0**,**0**,**1**,*,*,**0**,0,0,0,*,*,0,0,*,*,*,0,1,*,*,*,*	5638
2	0,0,0,**0**,**0**,0,0,**0**,**0**,*,0,0,**0**,**1**,*,**0**,0,0,0,0,*,0,0,0,*,*,0,0,1,*,*,0	68
3	0,0,0,0,**0**,0,0,0,**0**,**0**,0,0,0,**0**,**1**,0,0,0,0,0,0,0,0,0,0,*,0,0,0,1,*,0	4
4	0,0,0,0,0,0,0,0,0,**0**,0,0,0,0,**0**,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 1,0	1
4-18	$\Delta in : 0x0000\ 0002 \rightarrow \Delta out : 0x0004\ 0002$	Q_{r_1}
18	0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,**0**,0,0,0,0,**0**,**1**,0	1
19	0,0,0,0,0,0,0,0,*,0,0,0,1,*,*,0,0,0,0,**0**,0,0,0,**0**,**0**,**0**,0,0,**0**,**1**,**0**,0	4
20	0,0,0,*,0,0,0,*,*,*,0,1,*,*,*,0,0,0,**0**,**0**,**0**,0,**0**,**0**,*,0,**0**,**0**,**1**,*,*,0	176
21	0,0,*,*,*,0,*,*,*,*,1,*,*,*,*,0,0,**0**,**0**,*,**0**,**0**,**0**,*,*,*,**0**,**1**,*,*,*,**0**	34336
22	0,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,0,0,*,*,*,0,*,*,*,*,1,*,*,*,*,0	30906788

plaintext pairs, after $2^{32} \times (Q_{r_0}+1) \approx 2^{53.3}$ times (r_0+r+r_1) rounds encryption, using the fixed 6 bits of the ciphertext difference as conditions to filter out the wrong pairs with $2^{47.3}$ pairs left in approximately. Then use the $\{\Delta_C\}$ as the filter oracle, $2^{47.3}/Q_{r_1} \approx 2^{47.3-24.85} = 2^{22.45}$ pairs remained and should be stored in table T.

Considering the recovery of 4 consecutive rounds of subkeys, by partial decrypt the last four round, there are totally 35 bits of $rk_{21}^{10,15}$, $rk_{20}^{0,5,9-11,14,15}$, $rk_{19}^{10,15}$, $rk_{18}^{0,4-5,8-11,13-15}$ involved in the 18 bit conditions in the 18 to 22 rounds. Create a list of counters for the 35 subkey bits, and solve the 18 bit condition equations with each pairs remained in T. If the candidate subkey bits matches the equations then increment the counter. The counter associated with the candidate subkey bits has the highest count value.

For the data complexity, there requires 2^{32} plaintexts, and 3 tables of $\{\Delta_P\}$, $\{\Delta_C\}$ and T with $2^{21.3} + 2^{24.85} + 2^{22.45} \approx 2^{26.3}$ storage size. The computa-

Table 7. Extended Differential Path of 29-round SIMECK48 with sufficient conditions.

Rd	Input difference of each round represented in bits	Q_{r_0}
0	00000000*0************** 0000000*****************	$\approx 2^{29.41}$
1	00000000000*0*0****1**** 00000000*0**************	950080
2	0000000000000000*0*01*0* 00000000000*0*0****1****	1156
3	000000000000000000000101 0000000000000000*0*01*0*	16
4	000000000000000000000010 000000000000000000000101	1
4-25	$\Delta in: 0x000002\ 000005 \rightarrow \Delta out: 0x000005\ 000002$	Q_{r_1}
25	000000000000000000000101 000000000000000000000010	1
26	0000000000000000*0*01*0* 000000000000000000000101	16
27	00000000000*0*0****1**** 0000000000000000*0*01*0*	1156
28	00000000*0************** 00000000000*0*0****1****	950080
29	0000000***************** 00000000*0**************	$\approx 2^{29.41}$

tional effort contains the XOR addition, encryptions, filtering phase, subkey bits guessing, and the brute-force search phase. The computational time complexity $C_t = (2^{32} \times 2^{21.3})A + (2^{32} \times (2^{21.3}+1))E + ((2^{53.3})A + (2^{47.3})A) + (2 \cdot 2^{22.45} \cdot 2^{35} \cdot \frac{4}{22})E + 2^{64-35}E \approx 2^{54.3}A + 2^{56}E$. Here, 'A' represents a XOR addition operation and 'E' represents the encryption operation of attacked rounds. For the calculation of the success probability, we refer to the theory in [22], the success probability equals to $1 - \mathit{Poisscdf}(s, \lambda_r)$, where $s = \lfloor \lambda_r \rfloor$ is the number of hits that no more than right pairs λ_r, and $\mathit{Poisscdf}(s, \lambda_r)$ is the probability density function of Poisson distribution. Hence, we can get the success probability is 73% for the attack on SIMECK32.

For SIMECK48/96, we use a 21-round differential $\Delta in : 0x0002\ 0005 \rightarrow \Delta out : 0x0005\ 0002$ with the probability of $2^{-45.18}$ in Table 3, and add 4 rounds forward and 4 rounds backward of it to mount the key recovery attack against the 29 rounds of SIMECK48/96. The extended differential path with sufficient bit conditions listed in Table 7.

We choose 2^{47} plaintexts, with $2^{47} \times Q_{r_0} \approx 2^{47+29.41} = 2^{76.41}$ times XOR addition, $2^{76.41}$ plaintext pairs can be get, which will lead to $2^{47-45.18} \approx 3.53$ right pairs occured in average. After $2^{47} \times (Q_{r_0}+1) \approx 2^{76.41}$ times encryption for all pairs, we use the fixed 16 difference bits in the last round to filter out most part of wrong pairs with $2^{60.41}$ pairs left. Then use the $\{\Delta_C\}$ as the filter oracle, $2^{60.41-29.41} = 2^{31}$ pairs remained and should be stored in table T. Create counters for the candidate 54 subkey bits $rk_{27}^{9,11-23}, rk_{26}^{4,6,8-23}, rk_{25}^{1,3-23}$ involved in the last 4 rounds, and solve the 32 bit condition equations with each pairs remained in T. For the data complexity, there requires 2^{47} choosen plaintexts, and $2^{29.41} + 2^{29.41} + 2^{31} \approx 2^{31.5}$ storage size for $\{\Delta_P\}$, $\{\Delta_C\}$ and table T is needed. And for the computational effort, there needs $C_t = (2^{76.41})A + (2^{76.41})E + ((2 \cdot$

Table 8. Extended Differential Path of 35-round SIMECK64 with sufficient conditions.

Rd	Input difference of each round represented in bits
0	0000000000000000**00***0****1*** 0000000000000000***0************
1	00000000000000000*000**00***01** 0000000000000000**00***0****1***
2	0000000000000000000000*000**001* 00000000000000000*000**00***01**
3	00000000000000000000000000010001 0000000000000000000000*000**001*
4	00000000000000000000000000000000 00000000000000000000000000010001
4-31	$\Delta in: 0x00000000\ 00000011 \rightarrow \Delta out: 0x00000005\ 00000002$
31	00000000000000000000000000000101 00000000000000000000000000000010
32	000000000000000000000000*0*01*0* 00000000000000000000000000000101
33	0000000000000000000*0*0****1**** 000000000000000000000000*0*01*0*
34	0000000000000000*0************** 0000000000000000000*0*0****1****
35	000000000000000***************** 0000000000000000*0**************
Q_{r_0}	$\approx 2^{28.6}$,363076,1156,16,1 for Rd.= 0,1,2,3,4,respectively
Q_{r_1}	1,16,1156,950080,$\approx 2^{29.41}$ for Rd.= 31,32,33,34,35,respectively

$2^{76.41})A + (2 \cdot 2^{60.41})A) + (2 \cdot 2^{31} \cdot 2^{54} \cdot \frac{4}{29})E + 2^{96-54}E \approx 2^{77.04}A + 2^{83.14}E$, and the success probability is 78%.

For SIMECK64/128, we use the 27-round differential Δin:0x00000000 00000011 $\rightarrow$ Δout:0x00000005 00000002 with the probability of $2^{-60.75}$ in Table 3, by appending 4 rounds on the top and 4 rounds at the bottom, and extend it to mount the key recovery attack against the 35 rounds of SIMECK64/128. The extended dfferential path of the 35 rounds SIMECK64/128 is listed in Table 8. Choosing 2^{62} plaintexts, we can get $2^{90.6}$ plaintext pairs with $2^{90.6}$ XOR additions, and $2^{62-60.75} \approx 2.13$ right pairs occured in average. For all choosen plaintext pairs, there need $2^{90.6}$ encryptions of 35 rounds which lead to $2^{58.6}$ pairs left, and then use the $\{\Delta_C\}$ as the filter oracle, $2^{29.19}$ pairs stored in table T. There are 78 bits of $rk_{34}^{9,11-31}$, $rk_{33}^{4,6,8-31}$, $rk_{32}^{1,3-31}$ can be guessed. There requires 2^{62} choosen plaintexts for data complexity, and storage complexity is $2^{28.6}+2^{29.41}+2^{29.19} \approx 2^{30.7}$ for $\{\Delta_P\}$, $\{\Delta_C\}$ and table T. For the computational complexity, there needs $C_t = (2^{90.6})A + (2^{90.6})E + ((2^{58.6})A + (2^{29.19})A) + (2 \cdot 2^{29.19} \cdot 2^{78} \cdot \frac{4}{35})E + 2^{128-78}E \approx 2^{90.6}A + 2^{105.5}E$, and the success probability is 73%.

5 Conclusions

In this paper, we analyzed the differential and linear propagation properties of the bitwise AND operation with inputs mutually independent or dependent, and improved efficient algorithms by constructing the partial difference distribution table for bitwise AND operation. We searched out new differentials and linear hulls for SIMECK, which are the best results as far as we know. We applied our differentials to the key recovery attack on SIMECK and less complexity required while compared to previous differential attack.

Acknowledgements. The authors are very grateful to the anonymous reviewers for their valuable comments. This work was supported by the National Key Research and Development Program of China (No. 2017YFB0801900).

References

1. Abed, F., List, E., Lucks, S., Wenzel, J.: Differential cryptanalysis of round-reduced Simon and Speck. In: Cid, C., Rechberger, C. (eds.) FSE 2014. LNCS, vol. 8540, pp. 525–545. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46706-0_27
2. AlTawy, R., Rohit, R., He, M., Mandal, K., Yang, G., Gong, G.: sLiSCP: Simeck-based permutations for lightweight sponge cryptographic primitives. In: Selected Areas in Cryptography - SAC 2017–24th International Conference, Ottawa, ON, Canada, 16–18 August 2017, Revised Selected Papers, pp. 129–150 (2017)
3. Bagheri, N.: Linear cryptanalysis of reduced-round SIMECK variants. In: Biryukov, A., Goyal, V. (eds.) INDOCRYPT 2015. LNCS, vol. 9462, pp. 140–152. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26617-6_8
4. Banik, S., Pandey, S.K., Peyrin, T., Sasaki, Y., Sim, S.M., Todo, Y.: GIFT: a small present. In: Fischer, W., Homma, N. (eds.) CHES 2017. LNCS, vol. 10529, pp. 321–345. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66787-4_16
5. Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., Wingers, L.: The SIMON and SPECK families of lightweight block ciphers. Cryptology ePrint Archive, Report 2013/404 (2013)
6. Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., Wingers, L.: Notes on the design and analysis of SIMON and SPECK. IACR Cryptology ePrint Archive 2017/560 (2017)
7. Beierle, C.: Pen and paper arguments for SIMON and SIMON-like designs. In: Zikas, V., De Prisco, R. (eds.) SCN 2016. LNCS, vol. 9841, pp. 431–446. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44618-9_23
8. Biryukov, A., Roy, A., Velichkov, V.: Differential analysis of block ciphers SIMON and SPECK. In: Cid, C., Rechberger, C. (eds.) FSE 2014. LNCS, vol. 8540, pp. 546–570. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46706-0_28
9. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: an ultra-lightweight block cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74735-2_31
10. Borghoff, J., et al.: Prince - a low-latency block cipher for pervasive computing applications (full version). In: Proceedings of Advances in Cryptology - ASIACRYPT 2012–18th International Conference on the Theory and Application of Cryptology and Information Security (2012)
11. Chen, H., Wang, X.: Improved linear hull attack on round-reduced Simon with dynamic key-guessing techniques. In: Peyrin, T. (ed.) FSE 2016. LNCS, vol. 9783, pp. 428–449. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-52993-5_22

12. Kölbl, S., Leander, G., Tiessen, T.: Observations on the SIMON block cipher family. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9215, pp. 161–185. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-47989-6_8
13. Kölbl, S., Roy, A.: A brief comparison of Simon and Simeck. In: Bogdanov, A. (ed.) LightSec 2016. LNCS, vol. 10098, pp. 69–88. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-55714-4_6
14. Liu, Z., Li, Y., Wang, M.: Optimal differential trails in simon-like ciphers. IACR Trans. Symmetric Cryptol. **2017**(1), 358–379 (2017)
15. Liu, Z., Li, Y., Wang, M.: The security of simon-like ciphers against linear cryptanalysis. IACR Cryptology ePrint Archive 2017/576 (2017)
16. Matsui, M.: On correlation between the order of S-boxes and the strength of DES. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 366–375. Springer, Heidelberg (1995). https://doi.org/10.1007/BFb0053451
17. Qiao, K., Hu, L., Sun, S.: Differential analysis on simeck and SIMON with dynamic key-guessing techniques. In: Camp, O., Furnell, S., Mori, P. (eds.) ICISSP 2016. CCIS, vol. 691, pp. 64–85. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-54433-5_5
18. Qin, L., Chen, H., Wang, X.: Linear hull attack on round-reduced simeck with dynamic key-guessing techniques. In: Liu, J.K., Steinfeld, R. (eds.) ACISP 2016. LNCS, vol. 9723, pp. 409–424. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-40367-0_26
19. Ryabko, B., Soskov, A.: The distinguishing attack on speck, simon, simeck, HIGHT and LEA. Cryptology ePrint Archive, Report 2018/047 (2018)
20. Sun, S., et al.: Towards finding the best characteristics of some bit-oriented block ciphers and automatic enumeration of (related-key) differential and linear characteristics with predefined properties. Cryptology ePrint Archive, Report 2014/747 (2014)
21. Suzaki, T., Minematsu, K., Morioka, S., Kobayashi, E.: *TWINE*: a lightweight block cipher for multiple platforms. In: Knudsen, L.R., Wu, H. (eds.) SAC 2012. LNCS, vol. 7707, pp. 339–354. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-35999-6_22
22. Wang, N., Wang, X., Jia, K., Zhao, J.: Differential attacks on reduced SIMON versions with dynamic key-guessing techniques. Cryptology ePrint Archive, Report 2014/448 (2014)
23. Yang, G., Zhu, B., Suder, V., Aagaard, M.D., Gong, G.: The Simeck family of lightweight block ciphers. In: Güneysu, T., Handschuh, H. (eds.) CHES 2015. LNCS, vol. 9293, pp. 307–329. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48324-4_16
24. Zhang, K., Guan, J., Hu, B., Lin, D.: Security evaluation on simeck against zero-correlation linear cryptanalysis. IET Inf. Secur. **12**(1), 87–93 (2018)
25. Zhang, W., Bao, Z., Lin, D., Rijmen, V., Yang, B., Verbauwhede, I.: RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms. SCIENCE CHINA Inf. Sci. **58**(12), 1–15 (2015)