Improved Linear Attacks on the Chinese Block Cipher Standard

Mingjie Liu[1]* and Jiazhe Chen[2]

[1] Beijing International Center for Mathematical Research, Peking University, Beijing 100871, China
liumj9705@pku.edu.cn
[2] CNITSEC, China
jiazhechen@gmail.com

Abstract. The block cipher used in the Chinese Wireless LAN Standard (WAPI), SMS4, was recently renamed as SM4, and became the block cipher standard issued by the Chinese government[3]. This paper improves the previous linear cryptanalysis of SMS4 by giving the first 19-round one-dimensional approximations. The 19-round approximations hold with bias $2^{-62.27}$; we use one of them to leverage a linear attack on 23-round SMS4. Our attack improves the previous 23-round attacks by reducing the time complexity. Furthermore, the data complexity of our attack is further improved by the multidimensional linear approach.

Key words: Block Cipher, SMS4, Linear Cryptanalysis, Multidimensional Linear Cryptanalysis

1 Introduction

SMS4 was issued in 2006 by the Chinese government as the block cipher used in the wireless LAN products [26], the English translation of the specification can be found in [9]. Recently, in 2012, SMS4 was announced as the Chinese commercial block cipher standard with the name SM4 [25], which implies that this cipher will be more widely used.

Since SMS4 was known to the public, quite a few cryptanalytic results were proposed to evaluate its security. Liu et al. proposed an integral attack on 13-round SMS4 in [18]; Ji and Hu gave an algebraic cryptanalysis by analysing the structure of the cipher [14]. In [20], Lu presented a 14-round rectangle attack and a 16-round impossible differential attack. Later, Toz and Dunkelman claimed that the complexities in [20] are underestimated; they made a more comprehensive analysis of the attacks in [20] and further improved the results [30]. Zhang et al. gave a rectangle attack and a differential attack on SMS4 reduced to 16 rounds and 18 rounds, respectively [33]. Several attacks were proposed in [17] by Kim et al., which were rectangle and boomerang attacks on 18-round SMS4, as well as linear and differential attacks on the 22-round version. Etrog and Robshaw also presented a linear attack on 22-round SMS4 and discussed the possibility of extending the attack to 23 rounds by using multiple linear attack [10]. Zhang et al. gave an improved 22-round differential attack in [34]; Liu et al. gave a multiple linear attack on SMS4 reduced to 22 rounds [19]. Su et al. further improved the result in [34] by proposing a differential attack on the 23-round version [29]; the attack is the best previous attack in terms of attacked number of rounds and complexity. Cho and Nyberg responsed to the question in [10] and proposed a multidimensional linear attack on 23-round SMS4 [5]. In addition, Zhang and Jin gave the lower bound of the number of linear active S-boxes for SMS4-like ciphers in [32].

Linear cryptanalysis was proposed by Matsui [21]. The attack first finds the linear approximation between the plaintexts, ciphertexts and the key bits with the highest bias, then recovers one bit of the key information; this method is called Algorithm 1. Matsui also gave another algorithm named Algorithm 2 which is more efficient. Algorithm 2 adds an additional round to the bottom of the linear approximation; the attacker recovers a part of the key of the last round by guessing the partial key and ranking them by the number of plaintext-ciphertext pairs that satisfy the linear approximation. In 2007, Collard et al [6] used Fast Fourier Transform to reduce the off-line time complexity of Matsui's algorithm 2 from $O(2^{2k})$ to $O(k2^k)$ where k is the number of bits in key guessing.

Kaliski and Robshaw proposed multiple approximations linear cryptanalysis that the same key bit information is involved in different approximations [15,16]. Later, Biryukov et al. removed the restriction and presented a framework of linear attack using multiple linear approximations [3]. The attacks in

* This author is supported by China's 973 Program Grants No. 2013CB834201 and China Postdoctoral Science Foundation No. 2013M540786

[3] In this paper, we will keep using the old name SMS4 as it is more familiar to the cryptographic community.

[3,15,16] are all based on the assumption that the linear approximations are statistically independent. Using the multidimensional probability distribution and standard statistical methods, Hermelin et al. proposed the frameworks of multidimensional linear attacks without the assumption of statistical independence for Algorithm 1 and Algorithm 2 in [11] and [12], respectively. Later, Hermelin and Nyberg gave another statistical method to reduce the off-line time complexity of Algorithm 1 [13]. Nguyen et al. improved the on-line complexity of Algorithm 2 in [23].

Our contributions. The linear attacks on SMS4 in [5,10,17,19] are all based on 18-round linear approximations whose fundamental parts are 5-round iterative linear approximations. The first contribution of this paper is proposing a new family of 19-round linear approximations with a different approach. We give a two-step method for finding the linear approximations. In the first step, the MILP manner proposed in [22] by Mouha et al. is adopted to obtain the framework of a 19-round linear approximation with the smallest number of active S-boxes. Since the result found by MILP is a lower bound that usually cannot be achieved, we add one more active S-box to each active round. Then in the second step, we give an algorithm to search for the linear approximation of this form. A useful observation is given as the starting point of the algorithm. In order to reduce the time complexity caused by the large search space, a time-memory tradeoff method is also used. Our algorithm returns eight 16-round linear approximations, each of which can be extended to 25 19-round linear approximations with the same bias. Our 19-round approximations have the same number of active S-boxes as the previous 18-round ones. While we cannot ensure that each active S-box has the highest bias, our 19-round approximations hold with bias $2^{-62.27}$. The second contribution of the paper is improving the previous linear attacks on SMS4. With one of these 19-round linear approximations, we propose a linear attack on 23-round SMS4; the data complexity is $2^{126.54}$, the time complexity is about 2^{122} 23-round encryptions and the memory complexity is about 2^{116} bytes. By using more linear approximations and applying the multidimensional linear attack, the data complexity of our attack is improved to $2^{122.6}$, while the time and memory complexities are increased to $2^{122.7}$ encryptions and $2^{120.6}$ bytes, respectively.

The rest of the paper is organized as follows. Section 2 briefly describes the SMS4 block cipher and (multidimensional) linear attack, as well as some notations and definitions. Improved linear attacks on SMS4 are illustrated in Sect. 3. Finally, Sect. 4 concludes the paper.

2 Preliminaries

This section first denotes some notations and definitions used throughout the paper, then gives a brief description of SMS4 as well as the method of (multidimensional) linear cryptanalysis.

2.1 Notations and Definitions

- $\oplus$ bit-wise OR (XOR)
- $||$ cascade of two words
- $\lll n$ rotation to the left for n bits
- V_n the space of n-dimensional binary vectors
- Linear mask: for $\mathbf{x} = (x^0, \cdots, x^{n-1})$, $\mathbf{y} = (y^0, \cdots, y^{n-1}) \in V_n$, $\mathbf{x}$ is called a linear mask of $\mathbf{y}$ if $x \cdot y = x^0y^0 \oplus \cdots \oplus x^{n-1}y^{n-1}$, where $\cdot$ is the bit-wise inner product.

Definition 1. *Given a linear mask Γ, a characteristic function χ is defined as:*

$$\chi(\Gamma) = \begin{cases} 1 & \Gamma \neq 0 \\ 0 & \Gamma = 0, \end{cases}$$

Definition 2. *Branch Number [8]. If we denote the byte weight of a vector $\boldsymbol{v}$ as $HW(\boldsymbol{v})$, the linear branch number of a linear transformation L is*

$$min_{\boldsymbol{b} \neq \mathbf{0}}(HW(\boldsymbol{b}) + HW(L^t(\boldsymbol{b}))),$$

where $\boldsymbol{b}$ is a vector of linear masks, L^t is the transpose of L.

A function f: $V_n \to V_1$ is called a Boolean function. A function $\boldsymbol{f}$: $V_n \to V_m$ with $\boldsymbol{f} = (f_0, \cdots, f_{m-1})$, where f_i $(i = 0, \cdots, m-1)$ are Boolean functions, is named a vector Boolean function of dimension m.

The correlation of a Boolean function g: $V_n \to V_1$ is defined as:

$$c = 2^{-n}(\#\{\xi | g(\xi) = 0\} - \#\{\xi | g(\xi) = 1\}), \quad \xi \in V_n.$$

The bias of g is $\epsilon = c/2$. Throughout the paper, when we refer to the values of correlations and biases, we mean the absolute values of them.

2.2 Brief Description of SMS4

SMS4 is a block cipher with unbalanced generalized Feistel structure. The block size and key size of the cipher are both 128 bits; the number of rounds is 32. Denote the main key of SMS4 as MK =(MK_0,MK_1,MK_2, MK_3), the round subkeys $(rk_0, rk_1, \cdots, rk_{31})$ are deduced from MK by the key schedule algorithm, where MK_0,MK_1,$\cdots$,MK_3 and rk_i $(0 \le i \le 31)$ are 32-bit words.

Let the plaintext be $(P_0, P_1, P_2, P_3) = (X_0, X_1, X_2, X_3) \in (Z_2^{32})^4$, the encryption procedure can be described as:

$$\begin{aligned} X_{i+4} &= F(X_i, X_{i+1}, X_{i+2}, X_{i+3}) = X_i \oplus T(X_{i+1} \oplus X_{i+2} \oplus X_{i+3} \oplus rk_i) \\ &= X_i \oplus L \circ S(X_{i+1} \oplus X_{i+2} \oplus X_{i+3} \oplus rk_i), for\ i = 0, \cdots, 31, \end{aligned}$$

and the ciphertext $(C_0, C_1, C_2, C_3) = R(X_{32}, X_{33}, X_{34}, X_{35}) = (X_{35}, X_{34}, X_{33}, X_{32})$, where R is the switch transformation[4]. The decryption procedure is the same as the encryption procedure except that the subkeys are intervened in the reverse order.

The schematic description of one round of SMS4 is given in Fig. 1. One can know from the figure that the round function F is composed of subkey addition and the function T, where there are two layers in T, which are the non-linear layer S and the linear transformation L. In layer S, an 8×8 S-box is used four times in parallel. The specification of the S-box could be found in [9]. L transforms a 32-bit word B to a 32-bit word D:

$$D = L(B) = B \oplus (B \lll 2) \oplus (B \lll 10) \oplus (B \lll 18) \oplus (B \lll 24).$$

One can easily verify that the linear branch number of the linear transformation L is 5.

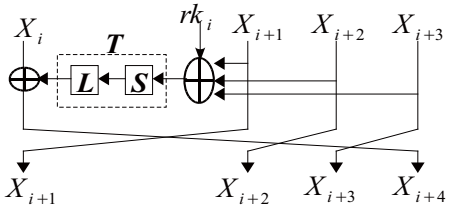

Fig. 1. One Round of SMS4

In the key schedule algorithm, first denote $K_i \in Z_2^{32}$ $(i = 0, \cdots, 35)$ and $(K_0, K_1, K_2, K_3) = (MK_0 \oplus FK_0, MK_1 \oplus FK_1, MK_2 \oplus FK_2, MK_3 \oplus FK_3)$, where FK_i $(i = 0, 1, \cdots, 3)$ are constants. Then the round subkeys are generated by:

$$rk_i = K_{i+4} = K_i \oplus T'(K_{i+1} \oplus K_{i+2} \oplus K_{i+3} \oplus CK_i),$$

where the function T' is the same as T, except the linear transformation L is replaced by L':

$$L'(B) = B \oplus (B \lll 13) \oplus (B \lll 23).$$

CK_j $(j = 0, 1, \cdots, 31)$ are constants, we refer to [9] for the details of them.

Note that, from the key schedule we deduce that if the subkeys of four successive rounds are known, then the main key is uniquely determined.

[4] To be simplified, we will omit the switch transformation R when we attack the reduced versions of SMS4.

2.3 Linear Attack

Linear cryptanalysis aims to find the linear approximation of a cipher:

$$\Gamma_P \cdot P \oplus \Gamma_C \cdot C = \Gamma_K \cdot K, \tag{1}$$

where Γ_P, Γ_C and Γ_K are called the linear masks of P, C and K, respectively. Denote $\Gamma_K \cdot K = \kappa$, if Eq. (1) holds with probability $1/2 \pm \epsilon$, Matsui showed that with about $|\epsilon|^{-2}$ plaintexts, the one bit information κ of the key can be recovered. The method is so called Algorithm 1 in [21]. Matsui also provided a more efficient manner (Algorithm 2) by adding an additional round after the linear approximation; the last round key is guessed and ranked. We restate the framework of linear cryptanalysis which follows the principle of Biryukov et al.'s approach [3]:

- `Distillation Phase.` This phase can be regarded as a phase that preprocesses the plaintexts/ciphertexts. Each plaintext-ciphertext pair is evaluated by the parity of a part of the linear approximation that is related to the plaintext/ciphertext; counters indexed by the key-relevant parts of the plaintext and ciphertext are incremented or decremented accordingly.
- `Analysis Phase.` This phase guesses a part of the (equivalent) key which is necessary to compute the other part of the linear approximation, evaluates the parity of this part of linear approximation and sets up a counter for each guessed (equivalent) key by utilising the values in the counters of the distillation phase. If we guess k-bit (equivalent) key and would like to get a-bit advantage[5], the top 2^{k-a} (equivalent) keys with the highest absolute values are the (equivalent) keys we will keep.
- `Search Phase.` For each of the kept (equivalent) key, guess the remaining key information and recover the correct key by trial encryption.

A formal way to calculate the success rate P_S and the data complexity can be found in [28].

2.4 Multidimensional Linear Attack

Unlike [15] and [16], Biryukov et al. proposed an approach that could use multiple linear approximations with different key bits involved [3]. However, these methods assume that the linear approximations are statistically independent.

Biryukov et al. proved that, if the attacker had m' linear approximations, c_i $(i = 0, \cdots, m'-1)$ are the theoretical correlations, $\hat{c}_i$ $(i = 0, \cdots, m'-1)$ are the empirical correlations, and $\kappa = (\kappa_0, \cdots, \kappa_{m'-1})$ are the key bits involved in the linear approximations, then the correct key is most likely implied when the minimal value of the following distance happens:

$$|\hat{\boldsymbol{c}} - \boldsymbol{c}_{\boldsymbol{\kappa}}|^2 = \sum_{i=0}^{m'-1} (\hat{c}_i - (-1)^{\kappa_i} c_i)^2,$$

where $\hat{\boldsymbol{c}} = (\hat{c}_0, \cdots, \hat{c}_{m'-1})$, $\boldsymbol{c}_{\boldsymbol{\kappa}} = ((-1)^{\kappa_0} c_0, \cdots, (-1)^{\kappa_{m'-1}} c_{m'-1})$. They also gave a heuristic enhancement to this method by adding more approximations that are dependent.

In [11], Hermelin et al. constructed the multidimensional probability distribution; based on this, they proposed the statistical methods for multidimensional linear versions of Algorithm 1 [11] and Algorithm 2 [12]. We simply recall these here.

Let $f: V_l \to V_n$ be a vector Boolean function, binary vectors $u_i \in V_l$ and $w_i \in V_n$ $(i = 0, \cdots, m-1)$ be linear masks such that (u_i, w_i) are linearly independent. Denote the functions g_i as:

$$g_i(\xi) := w_i \cdot f(\xi) \oplus u_i \cdot \xi,$$

the correlation of g_i is c_i, $i = 0, \cdots, m-1$.

Lemma 1. *(from [24]) Let $g = (g_0, \cdots, g_{m-1}): V_l \to V_m$ be a vector-valued Boolean function and $p = (p_0, \cdots, p_{2^m-1})$ its probability distribution. Then*

$$2^l p_\eta = 2^{-m} \sum_{b \in V_m} \sum_{\xi \in V_l} (-1)^{b \cdot (g(\xi) \oplus \eta)}, \qquad \eta \in V_m.$$

[5] a-bit advantage means that with probability P_S the right key is ranked among the top 2^{k-a} (equivalent) keys out of all 2^k key (equivalent) candidates.

For all $b \in V_m$, the combined approximations $b \cdot g$ has the correlation $c(b)$.

Corollary 1. *(from [11]) Let $g : V_l \to V_m$ be a Boolean function with probability distribution p and correlations $c(b)$ of $b \cdot g$. Then*

$$p_\eta = 2^{-m} \sum_{b \in V_m} (-1)^{b \cdot \eta} c(b), \qquad \eta \in V_m.$$

Let $q = (q_0, \cdots, q_{2^m-1})$ be the empirical distribution of p; statistical methods like χ^2-statistic and LLR-statistic could be used to distinguish the correct key from the wrong ones [12].

Hermelin and Nyberg pointed out that Biryukov et al.'s enhancement is equivalent to the convolution method [13], that is, for Biryukov et al.'s enhancement,

$$B(\kappa) = \sum_{b \in V_m} ((-1)^{b \cdot \kappa} c(b) - \hat{c}(b))^2, \quad \kappa \in V_m.$$

The key that minimises $B(\kappa)$ is suggested as the right key.

Moreover,

$$B(\kappa) = -2 \sum_{b \in V_m} (-1)^{b \cdot \kappa} c(b) \hat{c}(b) + \sum_{b \in V_m} (c(b)^2 + \hat{c}(b)^2).$$

Then the key that maximum

$$G(\kappa) = \sum_{b \in V_m} (-1)^{b \cdot \kappa} c(b) \hat{c}(b) = 2^m \sum_{\eta \in V_m} q_\eta p_{\eta \oplus \kappa}$$

is supposed to be the right key. Hermelin and Nyberg also deduced that the LLR-method has the smallest data complexity given the success rate P_S, and the data required by the convolution method had the same order of magnitude as the LLR-method in practice.

Before we present our attack, we recall two important tools to compute matrix multiplication. Given a k-dimensional vector $\mathbf{e}$ and a matrix $\mathbf{F}$ of size $k \times k$, the algorithm to obtain $\mathbf{Fe}$ can be optimized when every entry of the matrix $\mathbf{F}$ satisfies $F(i,j) = (-1)^{ij}$ or $F(i,j) = e^{2\pi\sqrt{-1}ij/k}$. The former one is called a Hadamard matrix and the later one is a Fourier matrix. For either Hadamard matrix or Fourier matrix, the vector $\mathbf{Fe}$ can be obtained with complexity $O(k \log k)$ by Fast Walsh Hadamard Transform [31] or Fast Fourier Transform[7]. In [6], Collard et al. proposed that when $F(i,j)$ can be denoted as a function of $i \oplus j$, the computation of $\mathbf{Fe}$ can be achieved by three products between a Fourier matrix and a vector. That implies $\mathbf{Fe}$ can be computed in $O(3k \log k)$ operations when $F(i,j) = f(i \oplus j)$, here f is a known function.

3 Improved Linear Attacks on SMS4

In this section, we first introduce how to construct our new linear approximations which are quite different from the previous ones with a semi-automatic method. Based on one of these linear approximations, we then give a 23-round attack on SMS4. Finally, we improve the data complexity of the attack by multidimensional linear extension.

3.1 A Novel Way to Find the Linear Approximations of SMS4

Biham pointed out that similar to differential cryptanalysis [2], characteristics can be defined in linear cryptanalysis [1]. Consequently, one can find the linear approximations of a cipher by concatenating characteristics of each round. However, there are important differences for the concatenation rule:

- For the XOR operation: If $x = y \oplus z$, Γ_x, Γ_y and Γ_z are the masks of x, y and z, respectively. Then $\Gamma_x = \Gamma_y = \Gamma_z$.
- For the branching operation: If $x = y = z$, Γ_x, Γ_y and Γ_z are the masks of x, y and z, respectively. Then $\Gamma_x = \Gamma_y \oplus \Gamma_z$.
- For the linear layer L: If $y = L(x)$, Γ_x and Γ_y are the masks of x and y, respectively. Then $\Gamma_x = L^t(\Gamma_y)$, where L^t is the transpose of L.

With these rules, one can find a linear approximation just as finding a differential trail. The rest of the task is to design a manner to find a linear approximation that is as long as possible. In this subsection, we propose a new method to search for the linear approximations of SMS4. The purpose of our method is to seek for the linear approximations with as few active S-boxes as possible, which results in non-iterative ones that are different from those in [5,10,17,19]. We use a two-step procedure to achieve this goal.

In the **first step**, we would like to determine the lower bound of the number of active S-boxes of the linear approximation, as well as the positions of the active rounds. As proposed by Mouha et al. in [22], this can be done using Mixed-Integer Liner Programming (MILP). For linear cryptanalysis, Mouha et al. first constructed the equations with extra binary dummy variables for all the branching operations and the linear transformations in the cipher, then they put the equations into a MILP solver for the answer. For example, if the masks of a branching operation are Γ_x, Γ_y, Γ_z and the binary dummy variable is d_1, then the equations for this branching operation are:

$$\begin{aligned} \chi(\Gamma_x) + \chi(\Gamma_y) + \chi(\Gamma_z) &\geq 2d_1 \ , \\ d_1 &\geq \chi(\Gamma_x) \ , \\ d_1 &\geq \chi(\Gamma_y) \ , \\ d_1 &\geq \chi(\Gamma_z) \ . \end{aligned}$$

Similarly, if the input and output byte-masks of a linear transformation are Γ_{in_1}, Γ_{in_2}, Γ_{in_3}, Γ_{in_4}, Γ_{out_1}, Γ_{out_2}, Γ_{out_3}, Γ_{out_4} and the dummy variable is d_2, then the equations are:

$$\begin{aligned} \chi(\Gamma_{in_1}) + \chi(\Gamma_{in_2}) + \chi(\Gamma_{in_3}) + \chi(\Gamma_{in_4}) + \chi(\Gamma_{out_1}) + \chi(\Gamma_{out_2}) + \chi(\Gamma_{out_3}) + \chi(\Gamma_{out_4}) &\geq \mathbf{b}d_2 \ , \\ d_2 &\geq \chi(\Gamma_{in_1}) \ , \\ d_2 &\geq \chi(\Gamma_{in_2}) \ , \\ d_2 &\geq \chi(\Gamma_{in_3}) \ , \\ d_2 &\geq \chi(\Gamma_{in_4}) \ , \\ d_2 &\geq \chi(\Gamma_{out_1}) \ , \\ d_2 &\geq \chi(\Gamma_{out_2}) \ , \\ d_2 &\geq \chi(\Gamma_{out_3}) \ , \\ d_2 &\geq \chi(\Gamma_{out_4}) \ . \end{aligned}$$

Where $\mathbf{b}$ is the linear branch number, in the case of SMS4, $\mathbf{b} = 5$. Following this method, we generate the MILP for SMS4 and put it to the solver implemented in SAGE [27]. Since the best previous linear approximation is 18 rounds, we try to find a linear approximation with 19 rounds. The solver gives us a 19-round linear approximation with one active S-box in the 1st, 4th, 5th, 8th, 9th, 12th, 13th, 16th and 17th rounds, respectively. However, what we have found is only a lower bound for the number of S-boxes, and one can never find such a linear approximation due to the limitation of degrees of freedom. Our solution is fixing the positions of the active rounds and increasing the number of active S-boxes until we find a valid linear approximation. We find that a linear approximation is probably valid when the number of active S-boxes in each active round is two, i.e., the linear approximation we try to find is with the form:

$$2-0-0-2-2-0-0-2-2-0-0-2-2-0-0-2-2-0-0.$$

In the **second step**, we give an algorithm to search for linear approximations with the above form. An observation that is the design criteria of the algorithm will be given first.

Observation 1 *In order to form the above 19-round linear approximation, the input masks of the T functions of the two consecutive active rounds should be the same.*

Proof. Denote Γ_{in}^i and Γ_{out}^i $(i = 1, \cdots, 6)$ as the input and output masks of T functions of the six-round linear approximation with the form $0-0-2-2-0-0$. Then $\Gamma_{in}^3 \oplus \Gamma_{out}^1 \oplus \Gamma_{in}^2 = \Gamma_{in}^4 \oplus \Gamma_{out}^5$, $\Gamma_{in}^4 \oplus \Gamma_{out}^6 \oplus \Gamma_{in}^5 = \Gamma_{in}^3 \oplus \Gamma_{out}^2$. Since we have $\Gamma_{in}^j = \Gamma_{out}^j = 0$ for $j = 1, 2, 5, 6$, consequently, $\Gamma_{in}^3 = \Gamma_{in}^4$. □

With Observation 1, we are now ready to introduce our Algorithm 1 which actually searches for the suitable masks of the last 16 rounds in the above 19-round linear approximation. Now let Γ_{in}^i and Γ_{out}^i $(i = 1, \cdots, 19)$ denote the input and output masks of T functions of rounds 1-19, respectively. Further

let function $\mathfrak{B}(\Gamma_{in}^i, L^t(\Gamma_{out}^i))$ be the bias of the function T [6], where L^t is the transpose of the linear transformation L of SMS4. Denote $\#(\Gamma)$ as the number of non-zero bytes in mask Γ.

A time-memory tradeoff is used in Algorithm 1: the algorithm first searches the linear masks of rounds 4-11 and save them in a list $\mathcal{L}_1$; then the algorithm searches the linear masks of rounds 8-15, looking for the compatible masks of rounds 8-11 in $\mathcal{L}_1$ and save the compatible ones in $\mathcal{L}_2$. Finally, we search the linear masks of rounds 16-19 for each mask in $\mathcal{L}_2$.

The Γ_{in}^j, Γ_{out}^j $(j = 4, 5, 8, 9, 12, 13, 16, 17)$ output by Algorithm 1 will form a 16-round linear approximation. By computing $\Gamma_{out}^1 := \Gamma_{out}^5 \oplus \Gamma_{in}^4$ and finding Γ_{in}^1 which make the bias of round 1 to be highest, one can trivially extend the 16-round linear approximation to a 19-round one. By choosing $B_1 = 0.0011$ and $B_2 = 2^{-15}$ we found eight 16-round linear approximations with reasonable bias that can be extended to useful 19 round approximations. The input and output masks (in hexadecimal) of the active T functions of these approximations are given in Table 1. Our program, which is implemented in C++, runs less than 2 days with a laptop. Note that one can further reduce the running time: Instead of steps 32 and 33, we put the resulting $(\Gamma_{in}^9, \Gamma_{out}^8, \Gamma_{out}^9, \Gamma_{in}^{12})$ in a list $\mathcal{L}_3$ and parallelize the modified steps 17-33 with steps 1-16. Then find the matches of $(\Gamma_{in}^9, \Gamma_{out}^8, \Gamma_{out}^9)$ in $\mathcal{L}_1$ and $\mathcal{L}_3$ with birthday paradox.

When extending backward to 19 rounds, we known that the number of active S-boxes in the first round is 2. Since for each S-box, there are 5 linear masks that lead to the highest bias, a total number of 200 19-round linear approximations with the same bias can be found. One of these linear approximations can be found in Table 2. In Table 2, the fourth and the fifth columns stand for the output and input masks of the S-box layer, respectively; the sixth column indicates the bias of the round, the rest of the columns give the masks of the intermediate values. We learn from [10] that the piling-up lemma [21] works quite well for SMS4, so the bias of the linear approximation in Table 2 is about $2^{-62.27}$.

3.2 Linear Attacks on 23-round SMS4

We add four additional rounds to the bottom of the 19-round linear approximation in Table 2 and give a linear attack on 23-round SMS4 (Fig. 2) using Matsui's Algorithm 2, as well as the technique in [6].

Denote $\alpha =$ 0xae007d6b, $\beta =$ 0x00233300, $\gamma =$ 0xd3f00289, $\delta =$ 0x6bd3d389 and $\zeta =$ 0x16009f6b.

Since from the linear approximation, we have $\alpha \cdot P_0 \oplus \beta \cdot P_1 \oplus \beta \cdot P_2 \oplus \gamma \cdot P_3 \oplus \delta \cdot X_{19} \oplus \zeta \cdot X_{20} = \kappa$, we need to guess the subkeys of the last four rounds and rank them.

We have $X_{22} = T(C_0 \oplus C_1 \oplus C_2 \oplus rk_{22}) \oplus C_3$,

$X_{21} = T(X_{22} \oplus C_0 \oplus C_1 \oplus rk_{21}) \oplus C_2 = T(T(C_0 \oplus C_1 \oplus C_2 \oplus rk_{22}) \oplus C_0 \oplus C_1 \oplus C_3 \oplus rk_{21}) \oplus C_2$,

$X_{20} = T(X_{21} \oplus X_{22} \oplus C_0 \oplus rk_{20}) \oplus C_1 = T(T(T(C_0 \oplus C_1 \oplus C_2 \oplus rk_{22}) \oplus C_3 \oplus C_0 \oplus C_1 \oplus rk_{21}) \oplus T(C_0 \oplus C_1 \oplus C_2 \oplus rk_{22}) \oplus C_0 \oplus C_2 \oplus C_3 \oplus rk_{20}) \oplus C_1$,

then $\delta \cdot X_{19} = \delta \cdot T(\mathfrak{m}_2(X_{20} \oplus X_{21} \oplus X_{22} \oplus rk_{19})) \oplus \delta \cdot C_0 = \delta \cdot T(\mathfrak{m}_2(T(T(T(C_0 \oplus C_1 \oplus C_2 \oplus rk_{22}) \oplus C_3 \oplus C_0 \oplus C_1 \oplus rk_{21}) \oplus T(C_0 \oplus C_1 \oplus C_2 \oplus rk_{22}) \oplus C_0 \oplus C_2 \oplus C_3 \oplus rk_{20})) \oplus \mathfrak{m}_2(T(T(C_0 \oplus C_1 \oplus C_2 \oplus rk_{22}) \oplus C_0 \oplus C_1 \oplus C_3 \oplus rk_{21})) \oplus \mathfrak{m}_2(T(C_0 \oplus C_1 \oplus C_2 \oplus rk_{22})) \oplus \mathfrak{m}_2(C_1 \oplus C_2 \oplus C_3 \oplus rk_{19})) \oplus \delta \cdot C_0$,

where $\mathfrak{m}_2(x) =$ 0xffff0000&x.

Further denote $\mathfrak{C}_0 = C_0 \oplus C_1 \oplus C_2$, $\mathfrak{C}_1 = C_0 \oplus C_1 \oplus C_3$, $\mathfrak{C}_2 = C_0 \oplus C_2 \oplus C_3$, $\mathfrak{C}_3 = C_1 \oplus C_2 \oplus C_3$.

In our attack, we aim to obtain a 8-bit advantage from the subkey bits we guessed. The reason is that we guess the subkeys of four successive rounds, and we can deduce the main key once these subkeys are known. The procedure of the attack is demonstrated as follows:

Distillation Phase.

1. Collect $N = 4|\epsilon|^{-2} = 2^{126.54}$ plaintext/ciphertext pairs.
2. Initialize 2^{112} counters $t[0] \cdots t[2^{112} - 1]$ to zero and regard them as a column vector $\mathfrak{t}$.
3. For each plaintext/ciphertext pair, calculate $b = \alpha \cdot P_0 \oplus \beta \cdot P_1 \oplus \beta \cdot P_2 \oplus \gamma \cdot P_3 \oplus \delta \cdot C_0 \oplus \zeta \cdot C_1$, increase the counter $t[\mathfrak{C}_0||\mathfrak{C}_1||\mathfrak{C}_2||\mathfrak{m}_2(\mathfrak{C}_3)]$ by one if $b = 0$; otherwise, decrease it by one.

Analysis Phase.

4. Utilising the technique in [6], we define a conceptual $2^{112} \times 2^{112}$ matrix M. The rows of M are indexed by $rk_{22}||rk_{21}||rk_{20}||\mathfrak{m}_2(rk_{19})$; the columns of M are indexed by $\mathfrak{C}_0||\mathfrak{C}_1||\mathfrak{C}_2||\mathfrak{m}_2(\mathfrak{C}_3)$. From [6], we know that each row or column of M defines the complete matrix. As a result, only the first column of M would be stored.

[6] The bias of function T can be calculate by the pilling lemma [21].

Algorithm 1: The Second Step for Finding the Linear Approximations

Input: Bound B_1, B_2, function $\mathfrak{B}$

Output: Γ_{in}^j, Γ_{out}^j $(j = 4, 5, 8, 9, 12, 13, 16, 17)$

for *all* Γ_{in}^9 *with* $\#(\Gamma_{in}^9) = 2$ **do**
  $\Gamma_{in}^8 := \Gamma_{in}^9$
  for *all* Γ_{out}^9 *with* $\#(L^t(\Gamma_{out}^9)) = 2$ **do**
    if $\mathfrak{B}(\Gamma_{in}^9, L^t(\Gamma_{out}^9)) \geq B_1$ **then**
      $\Gamma_{out}^5 := \Gamma_{out}^9 \oplus \Gamma_{in}^9$
      if $\#(L^t(\Gamma_{out}^5)) = 2$ **then**
        for *all* Γ_{out}^8 *with* $\#(L^t(\Gamma_{out}^8)) = 2$ **do**
          if $\mathfrak{B}(\Gamma_{in}^8, L^t(\Gamma_{out}^8)) \geq B_1$ **then**
            $\Gamma_{out}^{12} := \Gamma_{out}^8 \oplus \Gamma_{in}^9$
            if $\#(L^t(\Gamma_{out}^{12})) = 2$ **then**
              for *all* Γ_{in}^5 *with* $\#(\Gamma_{in}^5) = 2$ **do**
                $\Gamma_{in}^4 := \Gamma_{in}^5$
                if $\mathfrak{B}(\Gamma_{in}^5, L^t(\Gamma_{out}^5)) \geq B_1$ **then**
                  $\Gamma_{out}^4 := \Gamma_{out}^8 \oplus \Gamma_{in}^5$
                  if $\#(L^t(\Gamma_{out}^4)) = 2$ *and* $\mathfrak{B}(\Gamma_{in}^4, L^t(\Gamma_{out}^4)) \geq B_1$ **then**
                    push the quadruple $(\Gamma_{in}^9, \Gamma_{out}^8, \Gamma_{out}^9, \Gamma_{in}^4)$ into a list $\mathcal{L}_1$

for *all* Γ_{in}^8 *with* $\#(\Gamma_{in}^8) = 2$ **do**
  $\Gamma_{in}^9 := \Gamma_{in}^8$
  for *all* Γ_{out}^8 *with* $\#(L^t(\Gamma_{out}^8)) = 2$ **do**
    if $\mathfrak{B}(\Gamma_{in}^8, L^t(\Gamma_{out}^8)) \geq B_1$ **then**
      $\Gamma_{out}^{12} := \Gamma_{out}^8 \oplus \Gamma_{in}^9$
      if $\#(L^t(\Gamma_{out}^{12})) = 2$ **then**
        for *all* Γ_{out}^9 *with* $\#(L^t(\Gamma_{out}^9)) = 2$ **do**
          if $\mathfrak{B}(\Gamma_{in}^9, L^t(\Gamma_{out}^9)) \geq B_1$ **then**
            $\Gamma_{out}^5 := \Gamma_{out}^9 \oplus \Gamma_{in}^9$
            if $\#(L^t(\Gamma_{out}^5)) = 2$ **then**
              for *all* Γ_{in}^{12} *with* $\#(\Gamma_{in}^{12}) = 2$ **do**
                $\Gamma_{in}^{13} := \Gamma_{in}^{12}$
                if $\mathfrak{B}(\Gamma_{in}^{12}, L^t(\Gamma_{out}^{12})) \geq B_1$ **then**
                  $\Gamma_{out}^{13} := \Gamma_{out}^9 \oplus \Gamma_{in}^{12}$
                  if $\#(L^t(\Gamma_{out}^{13})) = 2$ *and* $\mathfrak{B}(\Gamma_{in}^{13}, L^t(\Gamma_{out}^{13})) \geq B_1$ **then**
                    if $(\Gamma_{in}^9, \Gamma_{out}^8, \Gamma_{out}^9)$ *is in* $\mathcal{L}_1$ **then**
                      push the quintuple $(\Gamma_{in}^9, \Gamma_{out}^8, \Gamma_{out}^9, \Gamma_{in}^4, \Gamma_{in}^{12})$ into a list $\mathcal{L}_2$

for *all the the quintuple in* $\mathcal{L}_2$ **do**
  $\Gamma_{out}^{16} := \Gamma_{out}^{12} \oplus \Gamma_{in}^{13}$
  if $\#(L^t(\Gamma_{out}^{16})) = 2$ **then**
    for *all* Γ_{in}^{16} *with* $\#(\Gamma_{in}^{16}) = 2$ **do**
      $\Gamma_{in}^{17} := \Gamma_{in}^{16}$
      $\Gamma_{out}^{17} := \Gamma_{out}^{13} \oplus \Gamma_{in}^{16}$
      if $(\#(L^t(\Gamma_{out}^{17})) = 2)$ *and* $(2 \times \mathfrak{B}(\Gamma_{in}^{16}, L^t(\Gamma_{out}^{16})) \times \mathfrak{B}(\Gamma_{in}^{17}, L^t(\Gamma_{out}^{17})) \geq B_2)$ **then**
        output Γ_{in}^j, Γ_{out}^j $(j = 4, 5, 8, 9, 12, 13, 16, 17)$

Table 1. Eight 16-round Linear Approximations

Round	4	5	8	9	12	13	16	17
input	b800e200	b800e200	b8e20000	b8e20000	00e2e200	00e2e200	b800e200	b800e200
output	d3d33189	16009f6b	6bd3d389	aee29f6b	d331d389	ae007d6b	d3d33189	16009f6b
input	00e200b8	00e200b8	e2e20000	e2e20000	e20000b8	e20000b8	00e200b8	00e200b8
output	009f6b16	d33189d3	007d6bae	31d389d3	e29f6bae	d3d3896b	009f6b16	d33189d3
input	b800e200	b800e200	00e2e200	00e2e200	b8e20000	b8e20000	b800e200	b800e200
output	16009f6b	d3d33189	ae007d6b	d331d389	aee29f6b	6bd3d389	16009f6b	d3d33189
input	00b800e2	00b800e2	00b8e200	00b8e200	0000e2e2	0000e2e2	00b800e2	00b800e2
output	89d3d331	6b16009f	896bd3d3	6baee29f	89d331d3	6bae007d	89d3d331	6b16009f
input	e200b800	e200b800	e20000e2	e20000e2	0000b8e2	0000b8e2	e200b800	e200b800
output	9f6b1600	3189d3d3	7d6bae00	d389d331	9f6baee2	d3896bd3	9f6b1600	3189d3d3
input	00e200b8	00e200b8	e20000b8	e20000b8	e2e20000	e2e20000	00e200b8	00e200b8
output	d33189d3	009f6b16	d3d3896b	e29f6bae	31d389d3	007d6bae	d33189d3	009f6b16
input	e200b800	e200b800	0000b8e2	0000b8e2	e20000e2	e20000e2	e200b800	e200b800
output	3189d3d3	9f6b1600	d3896bd3	9f6baee2	d389d331	7d6bae00	3189d3d3	9f6b1600
input	00b800e2	00b800e2	0000e2e2	0000e2e2	00b8e200	00b8e200	00b800e2	00b800e2
output	6b16009f	89d3d331	6bae007d	89d331d3	6baee29f	896bd3d3	6b16009f	89d3d331

Table 2. One of the 19-round Linear Approximations

Round	i	X_i	S out	S in	bias	X_{i+1}	X_{i+2}	X_{i+3}
1	0	0xae007d6b	0x004c6200	0x00233300	2^{-7}	0x00233300	0x00233300	0xd3f00289
2	1	0	0	0		0	0xd3d33189	0xae007d6b
3	2	0	0	0		0xd3d33189	0xae007d6b	0
4	3	0xd3d33189	0xda00f400	0xb800e200	$2^{-8.093}$	0xae007d6b	0	0
5	4	0x16009f6b	0x2e009600	0xb800e200	$2^{-7.193}$	0xb800e200	0xb800e200	0xd3d33189
6	5	0	0	0		0	0x6bd3d389	0x16009f6b
7	6	0	0	0		0x6bd3d389	0x16009f6b	0
8	7	0x6bd3d389	0xf44c0000	0xb8e20000	$2^{-8.093}$	0x16009f6b	0	0
9	8	0xaee29f6b	0xdaf40000	0xb8e20000	$2^{-8.093}$	0xb8e20000	0xb8e20000	0x6bd3d389
10	9	0	0	0		0	0xd331d389	0xaee29f6b
11	10	0	0	0		0xd331d389	0xaee29f6b	0
12	11	0xd331d389	0x00b89600	0x00e2e200	$2^{-8.42}$	0xaee29f6b	0	0
13	12	0xae007d6b	0x004c6200	0x00e2e200	$2^{-8.093}$	0x00e2e200	0x00e2e200	0xd331d389
14	13	0	0	0		0	0xd3d33189	0xae007d6b
15	14	0	0	0		0xd3d33189	0xae007d6b	0
16	15	0xd3d33189	0xda00f400	0xb800e200	$2^{-8.093}$	0xae007d6b	0	0
17	16	0x16009f6b	0x2e009600	0xb800e200	$2^{-7.193}$	0xb800e200	0xb800e200	0xd3d33189
18	17	0	0	0		0	0x6bd3d389	0x16009f6b
19	18	0	0	0		0x6bd3d389	0x16009f6b	0
20	19	0x6bd3d389	(0xf44c0000)	*		0x16009f6b	0	0

* We do not concern about this mask, as it is not within the linear approximation.

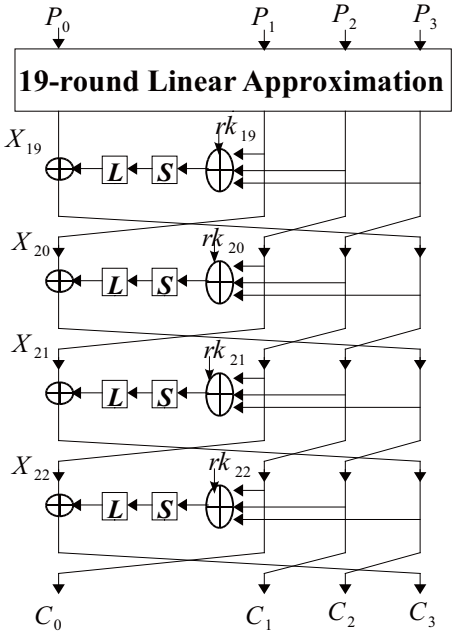

Fig. 2. Linear Attack on 23-round SMS4

5. Compute $g = \delta \cdot T(\mathfrak{m}_2(T(T(T(\mathfrak{C}_0 \oplus rk_{22}) \oplus \mathfrak{C}_1 \oplus rk_{21}) \oplus T(\mathfrak{C}_0 \oplus rk_{22}) \oplus \mathfrak{C}_2 \oplus rk_{20})) \oplus \mathfrak{m}_2(T(T(\mathfrak{C}_0 \oplus rk_{22}) \oplus \mathfrak{C}_1 \oplus rk_{21})) \oplus \mathfrak{m}_2(T(\mathfrak{C}_0 \oplus rk_{22})) \oplus \mathfrak{m}_2(\mathfrak{C}_3 \oplus rk_{19})) \oplus \zeta \cdot T(T(T(\mathfrak{C}_0 \oplus rk_{22}) \oplus \mathfrak{C}_1 \oplus rk_{21}) \oplus T(\mathfrak{C}_0 \oplus rk_{22}) \oplus \mathfrak{C}_2 \oplus rk_{20})$ and insert $(-1)^g$ to the corresponding row of the first column of M.
6. Using the technique in [6], efficiently compute $M\mathfrak{t} = \mathfrak{e}$ with Fast Fourier Transform [7].

Search Phase.

7. Since we aim to get a 8-bit advantage, for each of the top 2^{104} absolute values in $\mathfrak{e}$, guess the remaining 16 bits of rk_{19}, then we can get the main key by the key schedule and test the key by trial encryption.

The time complexity of Step 3 is about $2^{126.54}$ one-round encryptions, equivalent to 2^{122} 23-round encryptions, which is also the dominated complexity of the attack. The time complexity of Step 5 is about 2^{112} 4-round encryptions. In Step 6, $\mathfrak{e}$ is calculated by the technique in [6], which needs to carry out 3 Fast Fourier Transformations; the complexity is $3 \times 112 \times 2^{112} \approx 2^{120.4}$ arithmetic operations. The time complexity of Step 7 is 2^{120} encryptions. The memory complexity is about $(126.54 \times 2^{112} + 2^{112})/8 \approx 2^{116}$ bytes, which is to store $\mathfrak{t}$ and the first column of M.

Success Rate. By [28], the success rate $P_S = \Phi(2\sqrt{N}|\epsilon| - \Phi^{-1}(1 - 2^{-a-1})) = 86.74\%$. By the novel result in [4], $P_S = \Phi(2\sqrt{N}|\epsilon| - \sqrt{1 + \frac{N}{2^n}}\Phi^{-1}(1 - 2^{-a-1})) = 73.58\%$.

As shown in [4], the most deviations between the two models occur when adversary seeks a particularly big advantage or the data complexity gets close to the whole codebook. The advantage a in our attack is only 8, thus we try to reduce the data complexity in the next subsection.

3.3 Reducing the Data Complexity of the 23-round Attack by Multidimensional Extension

As mentioned in Section 3.1, for each of the 16-round linear approximations in Table 1, we have 5 input masks with the highest bias for each of the 2 active S-boxes in the first round when extending backward to 19 rounds. For the 16-round linear approximation we used in the previous subsection, the input masks of the 2 active S-boxes in the first round with the highest bias are given in Table 3.

In this subsection, we will use these 25 linear approximations simultaneously and mount an multidimensional linear attack on SMS4. By studying the 25 linear approximations, we found that they could

S-box1	S-box2
0x1f	0x33
0x23	0x4e
0x3c	0x8c
0x95	0xbf
0xa9	0xf1

Table 3. The Input Masks of the Two Active S-boxes in the First Round of the 19-round Approximation with Highest Bias

be spanned by $m = 6$ linearly independent linear approximations, where m is the number of base approximations. Then we can apply the theory of multidimensional linear attack in [11,12,13] to attack SMS4.

Denote binary vector $\mathrm{k} = (\kappa^0, \cdots, \kappa^5)$, where κ^i $(i = 0, \cdots, 5)$ are the one-bit key information involved in the 6 base linear approximations. Since the bias $|\epsilon|$ of each of the 19-round linear approximations is $2^{-62.27}$, the correlation is $c = 2|\epsilon| = 2^{-61.27}$. The capacity of the 25 linear approximations is $C_p = 25 \times c^2 = 2^{-117.9}$.

Here we use the same notations as the previous subsection except that β and γ are not a single value any more, but have different values corresponding to the 25 linear approximations.

Note that in this attack, we also aim to get a 8-bit advantage, the attack procedure is as follows:

Distillation Phase.

1. Collect $N = 2^{122.6}$ plaintext-ciphertext pairs.
2. Initialize 25×2^{112} counters $t[i][0] \cdots t[i][2^{112}-1]$ to zero $(i = 0, \cdots, 24)$, regard $t[i]$ as a column vector $\mathfrak{t}^i$.
3. For each plaintext-ciphertext pair and each of the 25 linear approximations, calculate $b = \alpha \cdot P_0 \oplus \beta \cdot P_1 \oplus \beta \cdot P_2 \oplus \gamma \cdot P_3 \oplus \delta \cdot C_0 \oplus \zeta \cdot C_1$, increment the counter $t[i][\mathfrak{C}_0||\mathfrak{C}_1||\mathfrak{C}_2||\mathfrak{m}_2(\mathfrak{C}_3)]$ if $b = 0$; otherwise, decrement it. For $i = 0, \cdots, 24$.

Analysis Phase.

4. Define conceptual $2^{112} \times 2^{112}$ matrices M^i. The rows of M^i are indexed by $rk_{22}||rk_{21}||rk_{20}||\mathfrak{m}_2(rk_{19})$; the columns of M^i are indexed by $\mathfrak{C}_0||\mathfrak{C}_1||\mathfrak{C}_2||\mathfrak{m}_2(\mathfrak{C}_3)$. Only the first columns of M^i would be stored.
5. For all the linear approximations, compute $g^i = \delta \cdot T(\mathfrak{m}_2(T(T(T(\mathfrak{C}_0 \oplus rk_{22}) \oplus \mathfrak{C}_1 \oplus rk_{21}) \oplus T(\mathfrak{C}_0 \oplus rk_{22}) \oplus \mathfrak{C}_2 \oplus rk_{20})) \oplus \mathfrak{m}_2(T(T(\mathfrak{C}_0 \oplus rk_{22}) \oplus \mathfrak{C}_1 \oplus rk_{21})) \oplus \mathfrak{m}_2(T(\mathfrak{C}_0 \oplus rk_{22})) \oplus \mathfrak{m}_2(\mathfrak{C}_3 \oplus rk_{19})) \oplus \zeta \cdot T(T(T(\mathfrak{C}_0 \oplus rk_{22}) \oplus \mathfrak{C}_1 \oplus rk_{21}) \oplus T(\mathfrak{C}_0 \oplus rk_{22}) \oplus \mathfrak{C}_2 \oplus rk_{20})$ and insert $(-1)^{g^i}$ to the corresponding row of the first column of M^i.
6. Efficiently compute $M^i\mathfrak{t}^i = \mathfrak{e}^i$ with Fast Fourier Transform.
7. Denote the set of 2^6 linear approximations spanned by the 6 base approximations as I, a binary vector $l \in V_6$ indicates the combination of the base approximations for each approximation in I. Consequently, each of the 25 approximations corresponds to a specific value l. Index the $\mathfrak{e}^i$ $(i = 0, \cdots, 24)$ obtained in the previous step by $\mathfrak{e}^l$, for an l that indicates an approximation that is out of the 25 ones we used, $\mathfrak{e}^l = \mathbf{0}$. Denote the j-th coordinate in $\mathfrak{e}^l$ as e^l_j, where $j = 0, \cdots, 2^{112} - 1$. For each j and each k, compute $G(j, \mathrm{k}) = \sum_{l \in V_6} (-1)^{l \cdot \mathrm{k}} (\frac{e^l_j}{N}) \times c$. This step can be calculated using Fast Walsh-Hadamard Transform [31]. Define $G(j) = max_k G(j, k)$.

Search Phase.

8. For the j that result the top 2^{104} $G(j)$, guess the rest 16 bits of rk_{19}, then we can get the main key by the key schedule and test the key by trial encryption.

The time complexity of Step 3 is about $25 \times 2^{122.6}/23 \approx 2^{122.7}$ 23-round encryptions. The time complexity of Step 5 is about 25×2^{112} 4-round encryptions. The complexity of Step 6 is $25 \times 3 \times 112 \times 2^{112} \approx 2^{125}$ arithmetic operations, which is about $2^{120.5}$ encryptions. Step 7 needs about $2^{112} \times 6 \times 2^6 = 2^{120.6}$ arithmetic operations. The time complexity of Step 8 is 2^{120} encryptions. The memory complexity is about $25 \times (122.6 \times 2^{112} + 2^{112})/8 \approx 2^{120.6}$ bytes.

Since the advantage $a = 8$, according to [12], the success rate $P_S = \Phi(\sqrt{NC_p} - \Phi^{-1}(1 - 2^{-m-a})) = 89.55\%$; if we evaluate the success rate by means of [4], it will be 88.71%.

4 Conclusion

This paper improved the best previous linear cryptanalysis of block cipher SMS4, a summary of some primary cryptanalytic results, as well as our results on SMS4 is given in Table 4. Note that, since the attacks are neither applied to the full cipher nor practical, they do not harm the security of the full SMS4.

Table 4. Summary of the Attacks on SMS4

#Rounds	Attack Type	Data	Time	Source
13	Integral	2^{16}	2^{114}	[18]
14	Rectangle	$2^{107.89}$	$2^{87.69}$	[30]
16	Rectangle	2^{125}	2^{116}	[33]
16	Impossible Differential	$2^{117.06}$	$2^{96.07}$	[30]
18	Rectangle	2^{124}	$2^{112.83}$	[17]
18	Boomerang	2^{120}	$2^{116.83}$	[17]
21	Differential	2^{118}	$2^{126.6}$	[33]
22	Linear	$2^{118.4}$	2^{117}	[10]
22	Linear	2^{117}	$2^{109.86}$	[17]
22	Multiple Linear	2^{112}	$2^{119.75}$†	[19]
22	Differential	2^{118}	$2^{125.71}$	[17]
22	Differential	2^{117}	$2^{112.3}$	[34]
23	Multidimensional Linear	$2^{126.6}$	$2^{127.4}$	[5]
23	Differential	2^{118}	$2^{126.7}$	[29]
23	Linear	$2^{126.54}$	2^{122}	This paper
23	Multidimensional Linear	$2^{122.6}$	$2^{122.7}$	This paper

† The complexity is $2^{124.21}$ arithmetic operations in the original paper, we convert it to the number of 22-round encryptions by assuming that one arithmetic operation is equivalent to one-round SMS4 encryption, which is quite overestimated. Hence it is reasonable.

Acknowledgement

The authors would like to thank the anonymous reviewers of IET Information Security for their valuable comments. The work is supported by the National Natural Science Foundation of China (Grant No.61202493).

References

1. Biham, E.: On Matsui's Linear Cryptanalysis. In: Santis, A.D. (ed.) Advances in Cryptology - EUROCRYPT '94. Lecture Notes in Computer Science, vol. 950, pp. 341–355. Springer (1995)
2. Biham, E., Shamir, A.: Differential Cryptanalysis of DES-like Cryptosystems. In: Menezes, A., Vanstone, S.A. (eds.) Advances in Cryptology - CRYPTO '90. Lecture Notes in Computer Science, vol. 537, pp. 2–21. Springer (1991)
3. Biryukov, A., Cannière, C.D., Quisquater, M.: On Multiple Linear Approximations. In: Franklin, M.K. (ed.) Advances in Cryptology - CRYPTO 2004. Lecture Notes in Computer Science, vol. 3152, pp. 1–22. Springer (2004)
4. Bogdanov, A., Tischhauser, E.: On the Wrong Key Randomisation and Key Equivalence Hypotheses in Matsui's Algorithm 2. In: FSE 2013 (2013)
5. Cho, J.Y., Nyberg, K.: Improved Linear Cryptanalysis of SMS4 Block Cipher. In: Symmetric Key Encryption Workshop 2011 (SKEW 2011) Lyngby, Denmark, 16-17 February 2011 (2011), `http://skew2011.mat.dtu.dk/proceedings/Improved%20Linear%20Cryptanalysis%20of%20SMS4%20Block%20Cipher.pdf`
6. Collard, B., Standaert, F.X., Quisquater, J.J.: Improving the Time Complexity of Matsui's Linear Cryptanalysis. In: Nam, K.H., Rhee, G. (eds.) ICISC 2007. Lecture Notes in Computer Science, vol. 4817, pp. 77–88. Springer (2007)
7. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, Third Edition. The MIT Press, 3rd edn. (2009)

8. Daemen, J., Rijmen, V.: The Design of Rijndael. Springer-Verlag New York, Inc., Secaucus, NJ, USA (2002)
9. Diffie, W., Ledin, G. (translators): SMS4 Encryption Algorithm for Wireless Networks. Cryptology ePrint Archive, Report 2008/329 (2008), http://eprint.iacr.org/
10. Etrog, J., Robshaw, M.J.B.: The Cryptanalysis of Reduced-Round SMS4. In: Avanzi, R.M., Keliher, L., Sica, F. (eds.) Selected Areas in Cryptography 2008. Lecture Notes in Computer Science, vol. 5381, pp. 51–65. Springer (2009)
11. Hermelin, M., Cho, J.Y., Nyberg, K.: Multidimensional Linear Cryptanalysis of Reduced Round Serpent. In: Mu, Y., Susilo, W., Seberry, J. (eds.) ACISP 2008. Lecture Notes in Computer Science, vol. 5107, pp. 203–215. Springer (2008)
12. Hermelin, M., Cho, J.Y., Nyberg, K.: Multidimensional Extension of Matsui's Algorithm 2. In: Dunkelman, O. (ed.) FSE 2009. Lecture Notes in Computer Science, vol. 5665, pp. 209–227. Springer (2009)
13. Hermelin, M., Nyberg, K.: Dependent Linear Approximations: The Algorithm of Biryukov and Others Revisited. In: Pieprzyk, J. (ed.) Topics in Cryptology - CT-RSA 2010. Lecture Notes in Computer Science, vol. 5985, pp. 318–333. Springer (2010)
14. Ji, W., Hu, L.: New Description of SMS4 by an Embedding over GF(2^8). In: Srinathan, K., Rangan, C.P., Yung, M. (eds.) INDOCRYPT 2007. Lecture Notes in Computer Science, vol. 4859, pp. 238–251. Springer (2007)
15. Kaliski, B.S., Robshaw, M.J.B.: Linear Cryptanalysis Using Multiple Approximations. In: Desmedt, Y. (ed.) Advances in Cryptology - CRYPTO '94. Lecture Notes in Computer Science, vol. 839, pp. 26–39. Springer (1994)
16. Kaliski, B.S., Robshaw, M.J.B.: Linear Cryptanalysis Using Multiple Approximations and FEAL. In: Preneel, B. (ed.) FSE 1994. Lecture Notes in Computer Science, vol. 1008, pp. 249–264. Springer (1995)
17. Kim, T., Kim, J., Hong, S., Sung, J.: Linear and Differential Cryptanalysis of Reduced SMS4 Block Cipher. Cryptology ePrint Archive, Report 2008/281 (2008), http://eprint.iacr.org/
18. Liu, F., Ji, W., Hu, L., Ding, J., Lv, S., Pyshkin, A., Weinmann, R.P.: Analysis of the SMS4 Block Cipher. In: Pieprzyk, J., Ghodosi, H., Dawson, E. (eds.) ACISP 2007. Lecture Notes in Computer Science, vol. 4586, pp. 158–170. Springer (2007)
19. Liu, Z., Gu, D., Zhang, J.: Multiple Linear Cryptanalysis of Reduced-Round SMS4 Block Cipher. Chinese Journal of Electronics 19(3), 389–393 (2010)
20. Lu, J.: Attacking Reduced-Round Versions of the SMS4 Block Cipher in the Chinese WAPI Standard. In: Qing, S., Imai, H., Wang, G. (eds.) ICICS 2007. Lecture Notes in Computer Science, vol. 4861, pp. 306–318. Springer (2008)
21. Matsui, M.: Linear Cryptoanalysis Method for DES Cipher. In: Helleseth, T. (ed.) Advances in Cryptology - EUROCRYPT '93. Lecture Notes in Computer Science, vol. 765, pp. 386–397. Springer (1994)
22. Mouha, N., Wang, Q., Gu, D., Preneel, B.: Differential and linear cryptanalysis using mixed-integer linear programming. In: Wu, C., Yung, M., Lin, D. (eds.) Inscrypt 2011. Lecture Notes in Computer Science, vol. 7537, pp. 57–76. Springer (2012)
23. Nguyen, P.H., Wu, H., Wang, H.: Improving the Algorithm 2 in Multidimensional Linear Cryptanalysis. In: Parampalli, U., Hawkes, P. (eds.) ACISP 2011. Lecture Notes in Computer Science, vol. 6812, pp. 61–74. Springer (2011)
24. Nyberg, K., Hermelin, M.: Multidimensional Walsh Transform and a Characterization of Bent Functions. In: Information Theory for Wireless Networks, 2007 IEEE Information Theory Workshop on. pp. 1–4 (2007)
25. Office of State Commercial Cryptography Administration: Announcement of 6 Cryptographic Standards. (in Chinese), http://www.oscca.gov.cn/News/201204/News_1228.htm
26. Office of State Commercial Cryptography Administration: Specification of SMS4, block cipher for WLAN products-SMS4. (in Chinese), http://www.oscca.gov.cn/UpFile/200621016423197990.pdf
27. SAGE: http://www.sagemath.org/.
28. Selçuk, A.A.: On Probability of Success in Linear and Differential Cryptanalysis. J. Cryptology 21(1), 131–147 (2008)
29. Su, B., Wu, W., Zhang, W.: Security of the SMS4 Block Cipher Against Differential Cryptanalysis. J. Comput. Sci. & Technol. 26(1), 130–138 (2011)
30. Toz, D., Dunkelman, O.: Analysis of Two Attacks on Reduced-Round Versions of the SMS4. In: Chen, L., Ryan, M.D., Wang, G. (eds.) ICICS 2008. Lecture Notes in Computer Science, vol. 5308, pp. 141–156. Springer (2008)
31. Yarlagadda, R., Hershey, J.: Hadamard Matrix Analysis and Synthesis: with Applications to Communications and Signal/Image Processing. Kluwer international series in engineering and computer science, Kluwer Academic Publishers (1997)
32. Zhang, B., Jin, C.: Practical Security Against Linear Cryptanalysis for SMS4-like Ciphers with SP Round Function. Sci China Inf Sci 55(9), 2161–2170 (2012)
33. Zhang, L., Zhang, W., Wu, W.: Cryptanalysis of Reduced-Round SMS4 Block Cipher. In: Mu, Y., Susilo, W., Seberry, J. (eds.) ACISP 2008. Lecture Notes in Computer Science, vol. 5107, pp. 216–229. Springer (2008)

34. Zhang, W., Wu, W., Feng, D., Su, B.: Some New Observations on the SMS4 Block Cipher in the Chinese WAPI Standard. In: Bao, F., Li, H., Wang, G. (eds.) ISPEC 2009. Lecture Notes in Computer Science, vol. 5451, pp. 324–335. Springer (2009)