Lv Ji-Qiang. Differential attack on five rounds of the SC2000 block cipher. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 26(4): 722–731 July 2011. DOI 10.1007/s11390-011-1171-2

Differential Attack on Five Rounds of the SC2000 Block Cipher*

Ji-Qiang Lv (吕继强)

Department of Computer Science, Ecole Normale Superieure, 45 Rue d'Ulm, Paris 75005, France

E-mail: lvjiqiang@hotmail.com

Received November 25, 2010; revised May 3, 2011.

Abstract The SC2000 block cipher has a 128-bit block size and a user key of 128, 192 or 256 bits, which employs a total of 6.5 rounds if a 128-bit user key is used. It is a CRYPTREC recommended e-government cipher in Japan. In this paper we address how to recover the user key from a few subkey bits of SC2000, and describe two 4.75-round differential characteristics with probability 2^{-126} of SC2000 and seventy-six 4.75-round differential characteristics with probability 2^{-127}. Finally, we present a differential cryptanalysis attack on a 5-round reduced version of SC2000 when used with a 128-bit key; the attack requires $2^{125.68}$ chosen plaintexts and has a time complexity of $2^{125.75}$ 5-round SC2000 encryptions. The attack does not threat the security of the full SC2000 cipher, but it suggests for the first time that the safety margin of SC2000 with a 128-bit key decreases below one and a half rounds.

Keywords cryptology, block cipher, SC2000, differential cryptanalysis

1 Introduction

SC2000[2] is a 128-bit block cipher with a user key of 128, 192 or 256 bits, which employs a total of 6.5 rounds for a 128-bit user key, and a total of 7.5 rounds for a 192 or 256-bit key. It was designed to "have high performance on a wide range of platforms from the low-end processors used in smart cards and mobile phones to the high-end ones that will be available in the near future by suitably implementing it in each platform, and also to have high security"[3]. In 2002, SC2000 became a CRYPTREC recommended e-government cipher in Japan[4], after a thorough analysis of its security and performance. Below we consider the version of SC2000 that uses 128 key bits. In the field of block cipher cryptanalysis, an exhaustive key search (i.e., brute force search) attack is usually assumed to be the best generic attack, and a cryptanalytic attack is commonly regarded as effective if it is faster (i.e., it has lower time complexity) than exhaustive key search.

The SC2000 designers first analysed the security of SC2000 against differential cryptanalysis[5] as well as certain other cryptanalytic techniques. In 2001, Raddum and Knudsen[6] presented a differential attack on 4.5-round SC2000, which is based on two 3.5-round differential characteristics with probabilities 2^{-106} and 2^{-107}, respectively. In 2002, by exploiting a few short differentials with large probabilities, Biham *et al.*[7] presented boomerang[8-9] and rectangle[10] attacks on 3.5-round SC2000, following the work described in [11]. In the same year, Yanami *et al.*[12] described a 2-round iterative differential characteristic with probability 2^{-58}, and obtained a 3.5-round differential characteristic with probability 2^{-101} by concatenating the 2-round differential twice and then removing the first half round; finally they presented a differential attack on 4.5-round SC2000 with a time complexity smaller than that of the attack of Raddum and Knudsen. Yanami *et al.* also presented linear[13] attacks on 4.5-round SC2000. The attacks on 4.5-round SC2000 are the best previously published cryptanalytic results on SC2000 in terms of the numbers of attacked rounds.

We note that these published cryptanalytic attacks on SC2000 retrieved only a few subkey bits of SC2000, and they did not address how to recover the user key. As SC2000 uses a very complicated key schedule algorithm, it seems tough to recover the user key from a few subkey bits. However, in this paper we find that there is an efficient way to do so in certain circumstances; more importantly, we describe two 4.75-round differential characteristics with probability 2^{-126} and seventy-six 4.75-round differential characteristics with probability 2^{-127}, building on the two-round iterative differential characteristic with probability 2^{-58} of Yanami *et al.* Finally, using some of these 4.75-round differential characteristics we present a differential cryptanalysis

Regular Paper

This work as well as the author was supported by the French ANR Project SAPHIR II.

*A preliminary version appeared in post-proceedings of INSCRYPT 2009[1].

attack on 5-round SC2000, faster than an exhaustive key search. The attack is the first published attack on 5-round SC2000. Table 1 summarises both the previous and our new cryptanalytic results on SC2000, where ACPC, CP and KP respectively refer to the required numbers of adaptive chosen plaintexts and ciphertexts, chosen plaintexts, and known plaintexts, and Enc. refers to the required number of encryption operations of the relevant reduced version of SC2000.

Table 1. Cryptanalytic Results on SC2000

Attack Type	Rounds	Data	Time	Source
Boomerang	3.5	2^{67} ACPC	$2^{116.74}$ Enc.†	[7]
Rectangle	3.5	$2^{84.6}$ CP	$2^{116.74}$ Enc.†	[7]
Linear	4.5	$2^{104.3}$ KP	$2^{121.33}$ Enc.†	[12]
Differential	4.5	2^{111} CP	$2^{118.33}$ Enc.†	[6]
	4.5	2^{104} CP	$2^{121.33}$ Enc.†	[12]
	5.0	$2^{125.68}$ CP	$2^{125.75}$ Enc.	Ours

†: The complexity is for obtaining the user key by using Property 1 of this paper.

The remainder of this paper is organised as follows. In the next section, we give the notation, and describe the SC2000 block cipher and differential cryptanalysis. In Section 3, we discuss how to recover the user key from a few subkey bits of SC2000. In Section 4, we give the 4.75-round differential characteristics. In Section 5, we present our differential attack on 5-round SC2000. Section 6 concludes the paper.

2 Preliminaries

In this section we give the notation used throughout this paper, and then briefly describe the SC2000 block cipher and differential cryptanalysis.

2.1 Notation

In all descriptions we assume that the bits of an n-bit value are numbered from 0 to $n-1$ from left to right, the most significant bit is the 0-th, a number without a prefix expresses a decimal number, and a number with prefix $0x$ expresses a hexadecimal number. We use the following notations:

$\oplus$ – bitwise logical exclusive OR (XOR) operation;

$\wedge$ – bitwise logical AND operation;

$\boxplus$ – addition modulo 2^{32};

$\boxminus$ – subtraction modulo 2^{32};

$\boxtimes$ – multiplication modulo 2^{32};

$\lll$ – left rotation of a bit string;

$\lfloor x \rfloor$ – the largest integer that is less than or equal to a value x;

$\circ$ – functional composition; (When composing functions X and Y, $X \circ Y$ denotes the function obtained by first applying X and then applying Y.)

$\bowtie$ – exchange of the left and right halves of a bit string;

$\overline{X}$ – bitwise logical complement of a bit string X.

2.2 SC2000 Block Cipher

SC2000 takes as input a 128-bit plaintext. For simplicity, we describe the plaintext P as four 32-bit words (d, c, b, a). The following three elementary functions I, B and R are used to define the SC2000 round function; as shown in Fig.1 the round function of SC200 is made up of two I functions, one B function and two R functions.

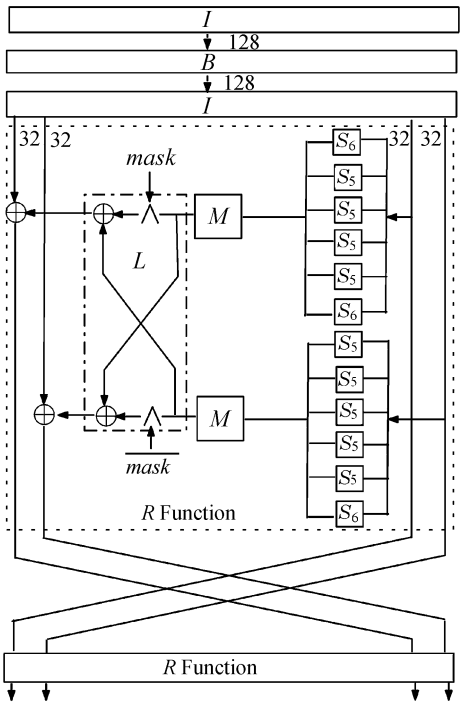

Fig.1. Round function of SC2000.

I function: the bitwise logical XOR ($\oplus$) operation of the 128-bit input with a 128-bit round subkey of four 32-bit words.

B function: a non-linear substitution, which applies the same 4×4 S-box S_4 32 times in parallel to the input. For a 128-bit input (d', c', b', a'), the output (d'', c'', b'', a'') is obtained in the following way: $(d''_k, c''_k, b''_k, a''_k) = S_4(d'_k, c'_k, b'_k, a'_k)$, where X_k is the k-th bit of the word X $(0 \leqslant k \leqslant 31)$.

R function: a substitution-permutation Feistel structure, which consists of three subfunctions S, M and L. Each of the right two 32-bit words of the input to the R function is divided into 6 groups containing 6, 5, 5, 5, 5 and 6 bits, respectively. These six groups are then passed sequentially through the S function, consisting of two 6×6 S-boxes S_6 and four 5×5 S-boxes S_5, and the linear M function that consists of

thirty-two 32-bit words $(M[0], \ldots, M[31])$. Given an input a, the output of the M function is defined as $a_0 \times M[0] \oplus \cdots \oplus a_{31} \times M[31]$. The outputs of the two M functions are then input to the L function. For a 64-bit input (a^*, b^*) the output of the L function is defined as $((a^* \wedge mask) \oplus b^*, (b^* \wedge \overline{mask}) \oplus a^*)$, where $mask$ is a constant (and $\overline{mask}$ is the complement of $mask$). Two masks $0x55555555$ and $0x33333333$ are used in SC2000, in the even and odd rounds, respectively. Finally, the output of the L function is XORed with the left two 32-bit words of the input to the R function, respectively. We denote the L and R functions with mask $0x55555555$ as L_5 and R_5, respectively, and the L and R functions with mask $0x33333333$ as L_3 and R_3, respectively.

SC2000 (with a 128-bit key) uses a total of fourteen 128-bit subkeys K_l^i, $(0 \leqslant i \leqslant 6,\ l = 0, 1)$, all derived from a user key of four 32-bit words $(uk[0], uk[1], uk[2], uk[3])$. The key schedule is as follows; see Fig.2 for a pictorial illustration.

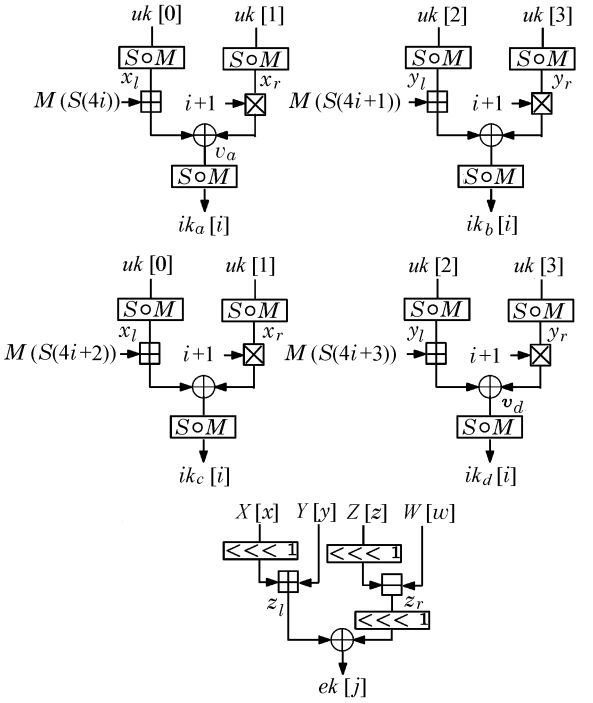

Fig.2. Intermediate-key and extended-key generation functions.

1) Generate 12 intermediate keys $ik_a[i]$, $ik_b[i]$, $ik_c[i]$, $ik_d[i]$ by the intermediate-key generation function, $(i = 0, 1, 2)$:

$$\begin{aligned} ik_a[i] = {} & M(S((M(S(uk[0])) \boxplus M(S(4i))) \oplus \\ & (M(S(uk[1])) \boxtimes (i+1)))), \\ ik_b[i] = {} & M(S((M(S(uk[2])) \boxplus M(S(4i+1))) \oplus \\ & (M(S(uk[3])) \boxtimes (i+1)))), \\ ik_c[i] = {} & M(S((M(S(uk[0])) \boxplus M(S(4i+2))) \oplus \\ & (M(S(uk[1])) \boxtimes (i+1)))), \\ ik_d[i] = {} & M(S((M(S(uk[2])) \boxplus M(S(4i+3))) \oplus \\ & (M(S(uk[3])) \boxtimes (i+1)))). \end{aligned}$$

2) Generate 56 extended keys $ek[j]$ by the following extended-key generation function, $(j = 0, 1, \ldots, 55)$, where $s, t, X, Y, Z, W, x, y, z, w$ are variables, $Order[4][12]$ is defined in Table 2, and $Index[4][9]$ is defined in Table 3.

$s = j \bmod 9$.

$t = (j + \lfloor \frac{j}{36} \rfloor) \bmod 12$.

$X = Order[0][t]$, $Y = Order[1][t]$, $Z = Order[2][t]$, $W = Order[3][t]$.

$x = Index[0][s]$, $y = Index[1][s]$, $z = Index[2][s]$, $w = Index[3][s]$.

$ek[j] = ((X[x] \lll 1) \boxplus Y[y]) \oplus (((Z[z] \lll 1) \boxminus W[w]) \lll 1)$.

3) $K_l^i = (ek[8i + 4l], ek[8i + 4l + 1], ek[8i + 4l + 2], ek[8i + 4l + 3])$.

Table 2. $Order[4][12]$

	0	1	2	3	4	5	6	7	8	9	10	11
0	ik_a	ik_b	ik_c	ik_d	ik_a	ik_b	ik_c	ik_d	ik_a	ik_b	ik_c	ik_d
1	ik_b	ik_a	ik_d	ik_c	ik_c	ik_d	ik_a	ik_b	ik_d	ik_c	ik_b	ik_a
2	ik_c	ik_d	ik_a	ik_b	ik_d	ik_c	ik_b	ik_a	ik_b	ik_a	ik_d	ik_c
3	ik_d	ik_c	ik_b	ik_a	ik_b	ik_a	ik_d	ik_c	ik_c	ik_d	ik_a	ik_b

Table 3. $Index[4][9]$

	0	1	2	3	4	5	6	7	8
0	0	1	2	0	1	2	0	1	2
1	0	1	2	1	2	0	2	0	1
2	0	1	2	0	1	2	0	1	2
3	0	1	2	1	2	0	2	0	1

The full 6.5-round encryption procedure of SC2000 can be described as: $I_{K_0^0} \circ B \circ I_{K_1^0} \circ R_5 \bowtie R_5 \circ I_{K_0^1} \circ B \circ I_{K_1^1} \circ R_3 \bowtie R_3 \circ I_{K_0^2} \circ B \circ I_{K_1^2} \circ R_5 \bowtie R_5 \circ I_{K_0^3} \circ B \circ I_{K_1^3} \circ R_3 \bowtie R_3 \circ I_{K_0^4} \circ B \circ I_{K_1^4} \circ R_5 \bowtie R_5 \circ I_{K_0^5} \circ B \circ I_{K_1^5} \circ R_3 \bowtie R_3 \circ I_{K_0^6} \circ B \circ I_{K_1^6}$. Note that we refer to the first round as Round 0.

We write K_l^i for the subkey used in the l-th I function of Round i, and write $K_{l,j}^i$ for the j-th bit of K_l^i, where $0 \leqslant i \leqslant 6, l = 0, 1, 0 \leqslant j \leqslant 127$. We number the 32 S_4 S-boxes in a B function from 0 to 31 from left to right.

2.3 Differential Cryptanalysis

Differential cryptanalysis was introduced in 1990 by Biham and Shamir[14]; it was the first cryptanalytic method more effective than an exhaustive key search to be proposed for the full DES[15] block cipher[5]. A

similar method was used a little earlier by Murphy[16] to analyse the FEAL block cipher[17].

Differential cryptanalysis takes advantage of how a specific difference in a pair of inputs of a cipher can affect a difference in the pair of outputs of the cipher, where the pair of outputs is obtained by encrypting the pair of inputs using the same key. The notion of difference can be defined in several ways; the most widely discussed is with respect to the XOR operation. The difference between the inputs is called the input difference, and the difference between the outputs of a function is called the output difference. The combination of the input difference and the output difference is called a differential. The probability of a differential is defined as follows.

Definition 1. *Suppose $\boldsymbol{E}$ is an n-bit block cipher and $K \in \{0,1\}^k$ is a key for $\boldsymbol{E}$. If x and y are n-bit blocks, then the probability of the differential (x, y) for $\boldsymbol{E}$, written $\Delta x \rightarrow \Delta y$, is defined to be*

$$\Pr_{\boldsymbol{E}_k}(\Delta x \rightarrow \Delta y) = \Pr_{P \in \{0,1\}^n}(\boldsymbol{E}_k(P) \oplus \boldsymbol{E}_k(P \oplus x) = y).$$

The following result follows trivially from Definition 1:

Proposition 1. *If $\boldsymbol{E}$ is an n-bit block cipher, and $K \in \{0,1\}^k$ is a key for $\boldsymbol{E}$, and x and y are n-bit blocks, then*

$$\Pr_{\boldsymbol{E}_k}(\Delta x \rightarrow \Delta y) = \frac{|\{P | \boldsymbol{E}_k(P) \oplus \boldsymbol{E}_k(P \oplus x) = y, P \in \{0,1\}^n\}|}{2^n}.$$

For a random function, the expected probability of a differential for any pair (x, y) is 2^{-n}. Therefore, if $\Pr_{\boldsymbol{E}_k}(\Delta x \rightarrow \Delta y)$ is larger than 2^{-n}, we can use the differential to distinguish $\boldsymbol{E}_k$ from a random function, given a sufficient number of chosen plaintext pairs.

Sometimes, we simply write $\Delta x \xrightarrow{\boldsymbol{E}/p} \Delta y$ to denote the differential $\Delta x \rightarrow \Delta y$ with probability p for $\boldsymbol{E}$.

Proposition 1 gives the accurate probability values of a differential from a theoretical point of view. However, it is usually hard to apply it to a block cipher with a large block size in reality, for example, $n = 64$ or 128 which is currently being widely used, and even harder when the differential operates on many rounds of the cipher. In practice, a multi-round differential is usually obtained by concatenating a few one-round differentials and (particularly for a Markov cipher[18]), the probability of the multi-round differential is regarded as the product of the probabilities of the one-round differentials under the following Assumption 1.

Assumption 1. *The round keys are independent and uniformly distributed.*

Assumption 1 connotes that the involved rounds are treated as independent. Usually, the round keys are actually dependent, being generated from a global user key under the key schedule algorithm of the cipher. As mentioned in [19], this is "most often not exactly the case, but as often it is a good approximation".

In 2008, Selçuk[20] formulated the success probability of a differential cryptanalysis attack.

Theorem 1 (from [20]). *For a differential attack on m key bits that uses a differential with probability p and N plaintext-ciphertext pairs and ranks the correct m-bit key value among the top r out of the 2^m possible key values, if p_r is the average probability that a given key value is suggested by a randomly chosen pair with the input difference, then under the assumption that the counters for the 2^m possible key values are independent and are identically distributed for all wrong key values, the success probability of the attack, denoted by P_S, is*

$$P_S = \Phi\Big(\frac{\sqrt{\mu \times S_N} - \Phi^{-1}(1 - 2^{-v})}{\sqrt{S_N + 1}}\Big),$$

where $\mu = p \times N$, $S_N = \frac{p}{p_r}$, $v = m - \log_2^r$, and $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution.

3 How to Recover the User Key from a Few Subkey Bits of SC2000

In general, a successful differential attack can reveal a few subkey bits of the attacked cipher, and a step after that is to deduce the user key from the subkey bits obtained. This can be easily done by exhaustive search when the cipher has such a key schedule that its constituent operations are invertible, e.g., the DES[15] and AES[21] block ciphers, but nevertheless it is tough for SC2000 — we cannot invert the operations for computing a round subkey to get the corresponding user key. None of the previously published studies has addressed this problem, and Raddum and Knudsen mentioned in [6]: "The strong key schedule in SC2000 prevents us from actually breaking 4.5 rounds by searching exhaustively for the remaining 96 bits in the first or last round key, since we cannot easily deduce the other round keys from them".

In this section we discuss how to recover the user key when a few subkey bits of SC2000 are given. We assume that the time complexity of an SC2000 encryption/decryption is evaluated by the numbers of B and S operations, and the time complexity of a computation of the key schedule is evaluated by the number of S operations. An optimised computation of the key schedule involves a minimum of 16 S operations, and a one-round SC2000 encryption/decryption involves 1 B operation and 4 S operations. Thus, a computation

of the key schedule is not negligible compared with an encryption/decryption, and from the designers' performance evaluation in [2] we learn that it takes more time than a full-round encryption/decryption. It looks like that every subkey bit of SC2000 depends on the entire 128 bits of the user key. Once a few subkey bits are obtained, there are seemingly only two solutions to recover the user key, as follows.

• A straightforward solution is to try each of the 2^{128} possible values for the user key, and we check whether it can generate the obtained subkey bits by the key schedule of SC2000; if so, then we further test it with trial encryptions using one or more known plaintext-ciphertext pairs, and if it passes this test then the trial value is very likely to be the correct key value. This solution has a time complexity of 2^{128} computations for the extended key(s) containing the obtained subkey bits.

• An alternative solution is to pre-compute and maintain a table of the concerned subkey bits for all the 2^{128} possible values of the user key, and then given the obtained subkey bits, we can find out the possible key values by looking up in the table; the correct key value can be further identified with an exhaustive search. This solution requires 2^{128} 128-bit memory, which is also very costly.

However, we observe that there exists a better way to recover the user key in certain circumstances, and our result is given as follows.

Property 1. *For a q-round SC2000 with* 128 *key bits* ($1 \leqslant q \leqslant 6.5$)*, if an extended key* $ek[\cdot]$ *whose intermediate-key inputs* $X[\cdot], Y[\cdot]$ *belong to the set* $\{ik_a[\cdot], ik_c[\cdot]\}$ *or* $\{ik_b[\cdot], ik_d[\cdot]\}$ *and* h *other subkey bits are known* ($h \geqslant 0$)*, then the correct value for* $(uk[0], uk[1], uk[2], uk[3])$ *can be obtained with an expected time complexity of approximately* $(5 \times 2^{96} + 4 \times \lfloor 2^{96-h} \rfloor)$ *S operations and* $\lfloor 2^{96-h} \rfloor$ *q-round SC2000 encryptions* (*provided that a known plaintext-ciphertext pair is available*).

Proof. Without loss of generality, we assume that $ek[51]$ and h bits of $ek[50]$ are known (this is the case for our attack given in Section 5), here $0 \leqslant h \leqslant 32$. Observe that the intermediate-key inputs for $ek[51]$ are $X[0] = ik_a[0]$, $Y[2] = ik_c[2]$, $Z[0] = ik_d[0]$, $W[2] = ik_b[2]$, and thus $X[0], Y[2] \in \{ik_a[\cdot], ik_c[\cdot]\}$. Let us consider the following algorithm for obtaining the correct value for $(uk[0], uk[1], uk[2], uk[3])$.

1) Define six public 32-bit constants $c_0, c_1, \ldots, c_5$, six unknown 32-bit constants $x_l, x_r, y_l, y_r, z_l, z_r$ and two 32-bit variables v_a, v_d (see Fig.2).

$$c_0 = M(S(0)), \quad c_1 = M(S(10)),$$
$$c_2 = M(S(3)), \quad c_3 = M(S(9)),$$
$$c_4 = M(S(2)), \quad c_5 = M(S(11)),$$
$$x_l = M(S(uk[0])), \quad x_r = M(S(uk[1])),$$
$$y_l = M(S(uk[2])), \quad y_r = M(S(uk[3])),$$
$$z_r = (ik_a[0] \lll 1) \boxplus ik_c[2],$$
$$z_l = (ik_d[0] \lll 1) \boxminus ik_b[2].$$

2) Guess a value for z_l, then compute $z_r = (z_l \oplus ek[51]) \lll 1$, and perform the following two sub-steps in parallel.

(a) Guess a value for $ik_a[0]$, compute $v_a = S^{-1}(M^{-1}(ik_a[0]))$, and do as follows:

(i) Guess a value for x_l, and compute $x_r = (x_l \boxplus c_0) \oplus v_a$.

(ii) Check whether (x_l, x_r) meets the following (1):

$$M(S((x_l \boxplus c_1) \oplus (x_r \boxtimes 3))) \boxplus (ik_a[0] \lll 1) = z_l. \quad (1)$$

If not, repeat Step 2(a)–(i) with another guess for x_l, (repeat the above step if all the possible guesses are tested in a step).

(b) Guess a value for $ik_d[0]$, compute $v_d = S^{-1}(M^{-1}(ik_d[0]))$, and do as follows.

(i) Guess a value for y_l, and compute $y_r = (y_l \boxplus c_2) \oplus v_d$.

(ii) Check whether (y_l, y_r) meets the following (2):

$$M(S((y_l \boxplus c_3) \oplus (y_r \boxtimes 3))) \boxplus z_r = (ik_d[0] \lll 1). \quad (2)$$

If not, repeat Step 2(b)–(i) with another guess for y_l, (repeat the above step if all the possible guesses are tested in a step).

3) For each value (x_l, x_r) passing Step 2(a)–(ii) and each value (y_l, y_r) passing Step 2(b)–(ii), check whether the resulting value for (x_l, x_r, y_l, y_r) can produce the given h bits of $ek[50]$ by the key schedule. If so, execute Step 4 with the value for (x_l, x_r, y_l, y_r); otherwise, repeat Step 2 with another guess.

4) For the value (x_l, x_r, y_l, y_r) passing Step 3, compute

$$uk[0] = S^{-1}(M^{-1}(x_l)), \quad uk[1] = S^{-1}(M^{-1}(x_r)),$$
$$uk[2] = S^{-1}(M^{-1}(y_l)), \quad uk[3] = S^{-1}(M^{-1}(y_r)),$$

and then test $(uk[0], uk[1], uk[2], uk[3])$ with a trial encryption using a known plaintext-ciphertex pair. If it yields the correct correspondence, output it as the correct value, and terminate the algorithm; otherwise, discard it and go to Step 2.

The algorithm requires a negligible memory. For each guess of $(z_l, ik_a[0])$ in Step 2(a), it is expected that there is only $2^{32} \times 2^{-32} = 1$ value for (x_l, x_r) meeting (1); and for each guess of $(z_l, ik_d[0])$ in Step 2(b), it is expected that there is only $2^{32} \times 2^{-32} = 1$ value for (y_l, y_r) meeting (2). There are $2^{32} \times 2^{32} \times 2^{32} = 2^{96}$ possible values for $(z_l, ik_a[0], ik_d[0])$, and thus it is expected

that there are 2^{96} possible values for (x_l, x_r, y_l, y_r) passing Step 2. On average, $2^{96} \times 2^{-h} = 2^{96-h}$ possible values for (x_l, x_r, y_l, y_r) will pass Step 3.

Step 2(a) has a computational complexity of approximately $2^{32} \times 2^{32} \times 2^{32} = 2^{96}$ S operations, and so is Step 2(b). Step 3 has a computational complexity of approximately $2^{32} \times 2^{32} \times 2^{32} \times 4 = 2^{98}$ S operations. Step 4 has a computational complexity of approximately $2^{96-h} \times 4 = 2^{98-h}$ S^{-1} operations and 2^{96-h} trial encryptions. Since Steps 2(a) and 2(b) are executed in parallel, the algorithm has a total time complexity of approximately $2^{96} + 2^{98} + 2^{98-h} = (5 + 2^{2-h}) \times 2^{96}$ S operations and 2^{96-h} q-round SC2000 encryptions.

The result follows trivially when we observe that if $h > 96$ there is no need to do a trial encryption in Step 4. □

Note that Property 1 is mainly due to the observation that the left two intermediate-key inputs for $ek[\cdot]$ are dependent on a different set of 64 user-key bits from the right two intermediate-key inputs.

We now apply Property 1 to some previously published cryptanalytic results on SC2000, as follows.

• Biham *et al.*'s boomerang and rectangle attacks on 3.5-round SC2000[7] retrieved 10 bits for each of the eight extended keys $ek[0]$, $ek[1]$, $ek[2]$, $ek[3]$, $ek[28]$, $ek[29]$, $ek[30]$, $ek[31]$. After a simple analysis, we know that each of $ek[28]$, $ek[29]$, $ek[30]$, $ek[31]$ meets the condition that the intermediate-key inputs $X[\cdot], Y[\cdot]$ belong to the set $\{ik_a[\cdot], ik_c[\cdot]\}$ or $\{ik_b[\cdot], ik_d[\cdot]\}$. Thus, we have 2^{22} possible values for each of the four extended keys and $h = 70$ in this attack, and it is expected to take approximately $5 \times 2^{96} \times 2^{22} \times \frac{1}{4} \times \frac{1}{3} \approx 2^{116.74}$ 3.5-round SC2000 encryptions to obtain the user key from the 80 subkey bits.

• Raddum and Knudsen's differential attack on 4.5-round SC2000[6] retrieved 8 bits for each of the eight extended keys $ek[0]$, $ek[1]$, $ek[2]$, $ek[3]$, $ek[36]$, $ek[37]$, $ek[38]$, $ek[39]$, a total of 64 subkey bits. Among the eight extended keys, only $ek[39]$ meets the condition that the intermediate-key inputs $X[\cdot], Y[\cdot]$ belong to the set $\{ik_a[\cdot], ik_c[\cdot]\}$ or $\{ik_b[\cdot], ik_d[\cdot]\}$. Thus, there are 2^{24} possible values for $ek[39]$ and $h = 56$ in this attack, so it is expected to take approximately $5 \times 2^{96} \times 2^{24} \times \frac{1}{4} \times \frac{1}{4} \approx 2^{118.33}$ 4.5-round SC2000 encryptions to obtain the user key from the 64 subkey bits.

• Yanami *et al.*'s differential attack on 4.5-round SC2000[12] retrieved 5 bits for each of the eight extended keys $ek[0]$, $ek[1]$, $ek[2]$, $ek[3]$, $ek[36]$, $ek[37]$, $ek[38]$, $ek[39]$, and their linear attacks on 4.5-round SC2000 retrieved 5 bits for each of the eight extended keys $ek[0]$, $ek[1]$, $ek[2]$, $ek[3]$, $ek[36]$, $ek[37]$, $ek[38]$, $ek[39]$ or for each of the four extended keys $ek[36]$, $ek[37]$, $ek[38]$, $ek[39]$. Similarly, we learn that it is expected to take approximately $5 \times 2^{96} \times 2^{27} \times \frac{1}{4} \times \frac{1}{4} \approx 2^{121.33}$ 4.5-round SC2000 encryptions to obtain the user key from the 40 or 20 subkey bits (where $h = 35$ or 15, respectively).

4 4.75-Round Differential Characteristics of SC2000

In this section we describe the 4.75-round differential characteristics.

First note that the differential distribution table of the S_4 S-box is given in [12], and the differential distribution table of the S_5 S-box is shown in Table A in Appendix. (The characteristics do not make an active S_6 S-box, so we do not give its differential distribution table.)

4.1 2-Round Iterative Differential Characteristic of Yanami *et al.*

In 2002, Yanami *et al.*[12] described the results of a search over all the possible two-round iterative differential characteristics with only one active S function in every round for any two consecutive rounds $I \circ B \circ I \circ R_5 \bowtie R_5 \circ I \circ B \circ I \circ R_3 \bowtie R_3$. Their result is that the best two-round iterative differential characteristic (i.e., that with the highest probability) is $(\alpha, \beta, \beta, 0) \rightarrow (\alpha, \beta, \beta, 0)$ with probability 2^{-58}: $(\alpha, \beta, \beta, 0) \xrightarrow{I \circ B \circ I/2^{-15}} (0, \beta, 0, 0) \xrightarrow{R_5 \bowtie R_5/2^{-16}} (\beta, \gamma, 0, \beta) \xrightarrow{I \circ B \circ I/2^{-11}} (\beta, 0, 0, 0) \xrightarrow{R_3 \bowtie R_3/2^{-16}} (\alpha, \beta, \beta, 0)$, where $\alpha = 0x01120000$, $\beta = 0x01124400$ and $\gamma = 0x00020000$.

4.2 4.75-Round Differential Characteristics

As a result, we can obtain a 4-round differential characteristic $(\alpha, \beta, \beta, 0) \rightarrow (\alpha, \beta, \beta, 0)$ with probability 2^{-116} by concatenating the above two-round iterative differential twice. It is essential to try to exploit an efficient (i.e., with a relatively high probability) differential operating over more than four rounds in order to break more rounds of SC2000. However, this 4-round differential cannot be extended to a differential characteristic operating over more than four rounds with a probability larger than 2^{-128}, as appending even a half round $R_3 \bowtie R_3$ at the beginning will cost a probability of 2^{-16} and appending a B function at the end will cost at least a probability of 2^{-13}.

Nevertheless, observe that from the above two-round iterative differential characteristic it follows that two-round iterative differential characteristic $(\beta, \gamma, 0, \beta) \rightarrow (\beta, \gamma, 0, \beta)$ for any two consecutive rounds $I \circ B \circ I \circ R_3 \bowtie R_3 \circ I \circ B \circ I \circ R_5 \bowtie R_5$ also holds with a probability of 2^{-58}: $(\beta, \gamma, 0, \beta) \xrightarrow{I \circ B \circ I/2^{-11}} (\beta, 0, 0, 0) \xrightarrow{R_3 \bowtie R_3/2^{-16}} (\alpha, \beta, \beta, 0) \xrightarrow{I \circ B \circ I/2^{-15}} (0, \beta, 0, 0) \xrightarrow{R_5 \bowtie R_5/2^{-16}} (\beta, \gamma, 0, \beta)$. It might seem counter-intuitive at first, but there is a

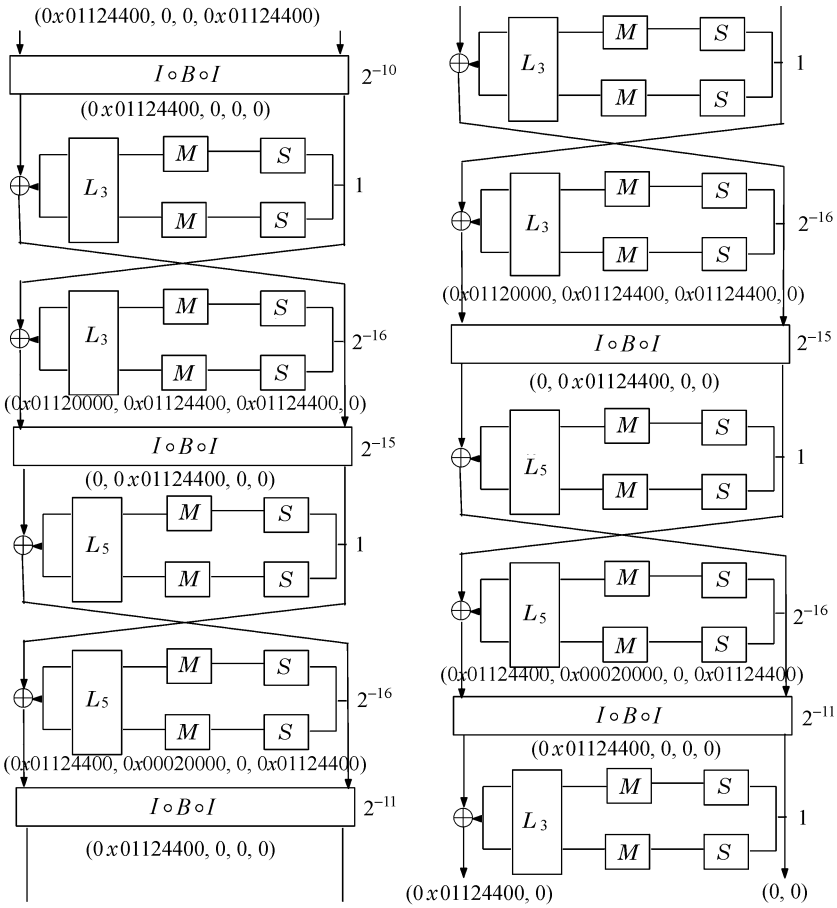

Fig.3. 4.75-round differential characteristic with probability 2^{-126}.

major difference between this and the previous iterative 2-round differential characteristic: we can append a 0.75-round differential characteristic $(\beta,\gamma,0,\beta) \overset{I\circ B\circ I\circ R_3}{\rightarrow} (\beta,0,0,0)$ with a probability of 2^{-11} at the end of this differential characteristic! Therefore, we can obtain a 4.75-round differential characteristic $(\beta,\gamma,0,\beta) \rightarrow (\beta,0,0,0)$ with probability 2^{-127}. Further, by changing the input difference to the difference $(\beta,0,0,\beta)$ we can get a 4.75-round differential characteristic with probability 2^{-126}: $(\beta,0,0,\beta) \rightarrow (\beta,0,0,0)$, and this 4.75-round differential characteristic is depicted in Fig.3.

Additionally, since $\Pr_{S_4}(\Delta 0x9 \rightarrow \Delta 0x8) = 2^{-2}$ and $\Pr_{S_4}(\Delta 0xD \rightarrow \Delta 0x4) = \Pr_{S_4}(\Delta 0xD \rightarrow \Delta 0x8) = 2^{-3}$ by the differential distribution table of the S_4 S-box given in [12], thus by changing the output difference of the last B function of the above 4.75-round differential characteristic with probability 2^{-126} to $(\theta,\phi,0,0)$, we get another 4.75-round differential characteristic with probability 2^{-126}: $(\beta,0,0,\beta) \rightarrow (\theta,\phi,0,0)$, where $\theta = 0x01104400$ and $\phi = 0x00020000$.

By the differential distribution table of the S_4 S-box, we have $\Pr_{S_4}(\Delta 0x9 \rightarrow \Delta 0x4) = \Pr_{S_4}(\Delta 0x9 \rightarrow \Delta 0xC) = 2^{-3}$. So when we change the output difference for only one of the four active S-boxes (7, 11, 17, 21) in the last B function of the above two 4.75-round differential characteristics with probability 2^{-126} to a value in $\{0x4, 0xC\}$, we get a total of $2 \times 4 \times 2 = 16$ 4.75-round differential characteristics with probability 2^{-127}.

We denote by Θ the set of the output differences of the two 4.75-round differential characteristics with probability 2^{-126} and the sixteen 4.75-round differential characteristics with probability 2^{-127}.

Note that when we change the input difference for only one of the five active S_4 S-boxes in the first B function of the two 4.75-round differential characteristics with probability 2^{-126} to a value in $\{0x1, 0x2, 0x6, 0x7, 0xD, 0xF\}$, we get $6 \times 5 \times 2 = 60$ additional 4.75-round differential characteristics with probability 2^{-127} by the differential distribution table of the S_4 S-box.

In summary, we obtain two 4.75-round differential characteristics with probability 2^{-126} and seventy-six 4.75-round differential characteristics with probability 2^{-127}, as follows.

• Two 4.75-round differential characteristics with probability 2^{-126}:

1. $(\beta, 0, 0, \beta) \rightarrow (\beta, 0, 0, 0)$,
2. $(\beta, 0, 0, \beta) \rightarrow (\theta, \phi, 0, 0)$,

where $\beta = 0x01124400$, $\theta = 0x01104400$, $\phi = 0x00020000$.

• Seventy-six 4.75-round differential characteristics with probability 2^{-127}:

1) Sixteen 4.75-round differential characteristics obtained by changing the output difference for only one of the four active S-boxes (7, 11, 17, 21) in the last B function of the above two 4.75-round differential characteristics with probability 2^{-126} to a value in $\{0x4, 0x\mathrm{C}\}$.

2) Sixty 4.75-round differential characteristics obtained by changing the input difference for only one of the five active S_4 S-boxes in the first B function of the above two 4.75-round differential characteristics with probability 2^{-126} to a value in $\{0x1, 0x2, 0x6, 0x7, 0x\mathrm{D}, 0x\mathrm{F}\}$.

In a natural way, we might try to find a better differential characteristic on greater than four rounds by first exploiting short differentials with similar structures and then concatenating them, for the above 4.75-round differential obtained from the two-round iterative differential is just a special case among these. Motivated by this idea, we perform a computer search over all the possible differentials for such one round $R \bowtie R \circ I \circ B \circ I$ with only one R function active and the right two 32-bit input differences and one of the left two 32-bit input differences being zero; moreover, in order to ensure that the resulting differential is capable of being concatenated with itself, we also require that the right two 32-bit output words and one of the left two 32-bit output words have a zero difference. Surprisingly, we find that the differential characteristics $(\beta, 0, 0, 0) \overset{R_3 \bowtie R_3 \circ I \circ B \circ I/2^{-31}}{\longrightarrow} (0, \beta, 0, 0)$ and $(0, \beta, 0, 0) \overset{R_5 \bowtie R_5 \circ I \circ B \circ I/2^{-27}}{\longrightarrow} (\beta, 0, 0, 0)$ in the above two-round iterative differential are the best (i.e., with the highest probabilities) among those with the same forms, respectively. Our search for other similar forms gives no better result.

5 Differential Attack on 5-Round SC2000

In this section, we present a differential cryptanalysis attack on the following 5 rounds of SC2000 when used with a 128-bit key: $I_{K_0^1} \circ B \circ I_{K_1^1} \circ R_3 \bowtie R_3 \circ I_{K_0^2} \circ B \circ I_{K_1^2} \circ R_5 \bowtie R_5 \circ I_{K_0^3} \circ B \circ I_{K_1^3} \circ R_3 \bowtie R_3 \circ I_{K_0^4} \circ B \circ I_{K_1^4} \circ R_5 \bowtie R_5 \circ I_{K_0^5} \circ B \circ I_{K_1^5} \circ R_3 \bowtie R_3 \circ I_{K_0^6}$. (Strictly speaking, this is a little more than 5 rounds.)

5.1 Preliminary Results

First observe that the output differences in the set Θ have a constant zero value in 54 bit positions of the left half and have a zero value in the 64 bit positions of the right half, (see Subsection 4.2 for definition of Θ). Among the remaining 10 bit positions of the left half, there are a total of 18 possible values, corresponding to the 18 output differences in Θ; we denote by Γ the set of the 18 possible values. The left half of an output difference in Θ will become the right half of the output difference after the following $R_3 \circ I_{K_0^6}$ operation.

On the other hand, having known the 128-bit difference after the $I_{K_0^6}$ function for a ciphertext pair, we only need to guess the 64 subkey bits $(K^6_{0,64}, \cdots, K^6_{0,127})$ of K_0^6 to check whether this pair could produce an expected difference just before the adjacent R_3 function. In our case, for a candidate difference whose right half is equal to the left half of one difference in $\boldsymbol{\Theta}$, we only need to guess at most the 40 subkey bits $(K^6_{0,70}, \cdots, K^6_{0,89}, K^6_{0,102}, \cdots, K^6_{0,121})$ corresponding to the eight S_5 S-boxes in the adjacent R_3 function to determine whether a ciphertext pair with a candidate difference could produce one of the output differences of the eighteen 4.75-round differential characteristics.

5.2 Attack Procedure

By using the eighteen 4.75-round differential characteristics with input difference $(\beta, 0, 0, \beta)$, we can mount a differential attack on the 5-round SC2000. The attack procedure is as follows.

1) Initialize 2^{40} counters for the 2^{40} possible values of the 40 subkey bits $(K^6_{0,70}, \ldots, K^6_{0,89}, K^6_{0,102}, \ldots, K^6_{0,121})$ in the $I_{K_0^6}$ function.

2) Choose $2^{124.68}$ plaintext pairs with difference $(\beta, 0, 0, \beta)$. In a chosen-plaintext attack scenario, obtain the corresponding ciphertexts for every plaintext pair, and do as follows.

(a) Check whether the ciphertext pair has a zero difference in the following 54 bit positions of the right half: $(0, 1, \ldots, 6, 8, \ldots, 10, 12, 13, 15, 16, 18, \ldots, 20, 22, \ldots, 38, 40, \ldots, 42, 44, 45, 47, 48, 50, \ldots, 52, 54, \ldots, 63)$. If so, execute Step 2(b); otherwise, discard it. (These 54 bit positions correspond to the 54 bit positions of the left half of Θ that have a constant zero value.)

(b) Check whether the ciphertext pair has a difference belonging to Γ in the 10 bit positions (7, 11, 14, 17, 21, 39, 43, 46, 49, 53) of the right half. If so, execute Step 2(c); otherwise, discard it. (These 10 bit positions correspond to the remaining 10 bit positions of the left half of Θ.)

(c) For each possible value of the 40 subkey bits, partially decrypt the ciphertext pair through the $I_{K_0^6}$ function and the eight S_5 S-boxes in the adjacent R_3 operation, compute the 64-bit difference just after the L_3 operation in the R_3 operation, then XOR it with

the left 64-bit difference of the ciphertext pair, and finally check whether the resultant 64-bit difference is zero. If so, increase 1 to the counter corresponding to the possible value for the 40 subkey bits.

3) For the values of $(K^6_{0,70},\ldots,K^6_{0,89},K^6_{0,102},\ldots,K^6_{0,121})$ corresponding to the 2^r counters with the top 2^r numbers, (a specific value of r will be given in Subsection 5.3), compute possible values for $ek[51]$, and apply the algorithm in Section 3 to find the correct user key.

5.3 Complexity Analysis

The attack requires $2^{125.68}$ chosen plaintexts, and requires about 2^{40} bytes of memory, used for the 2^{40} counters. It is expected that $2^{124.68}\times 2^{-54}=2^{70.68}$ ciphertext pairs pass the condition in Step 2(a), and $2^{70.68}\times\frac{18}{2^{10}}\approx 2^{64.85}$ ciphertext pairs pass the condition in Step 2(b). The time complexity of Step 2 is dominated by the partial decryptions in Step 2(c), which is approximately $2\times 2^{64.85}\times 2^{40}\times\frac{1}{2}\times\frac{1}{5}\approx 2^{102.53}$ 5-round SC2000 encryptions.

The signal-to-noise ratio for the attack is $\frac{2\times 2^{-126}+16\times 2^{-127}}{18\times 2^{-128}}\approx 2^{1.15}$. In Step 2(b), there are $2^{124.68}\times(2\times 2^{-126}+16\times 2^{-127})\approx 16$ right ciphertext pairs for the correct key guess.

Now we analyse the time complexity of Step 3. As mentioned in Section 3, the extended key $ek[51]$ meets the condition that the intermediate-key inputs $X[\cdot]$, $Y[\cdot]$ belong to the set $\{ik_a[\cdot],ik_c[\cdot]\}$. For each possible value of $(K^6_{0,102},\cdots,K^6_{0,121})$, there are 2^{12} possible values for $ek[51]$, because of 12 unknown bits $(K^6_{0,96},\ldots,K^6_{0,101},K^6_{0,121},\ldots,K^6_{0,127})$.

When we set $r=15$, among the $2^r=2^{15}$ values for $(K^6_{0,70},\ldots,K^6_{0,89},K^6_{0,102},\ldots,K^6_{0,121})$ in Step 3 there are at most 2^{15} values for $(K^6_{0,102},\ldots,K^6_{0,121})$; and for each possible value of $(K^6_{0,102},\ldots,K^6_{0,121})$, on average there is only one value for $(K^6_{0,70},\ldots,K^6_{0,89})$, that is we have $h=20-\log_2^1=20$ bits information of $ek[50]$. Consequently, there are at most $2^{15}\times 2^{12}=2^{27}$ possible values for $ek[51]$. So by Property 1 we learn that Step 3 has an expected time complexity of approximately $2^{27}\times[(5\times 2^{96}+4\times 2^{96-20})\times\frac{1}{4}\times\frac{1}{5}+2^{96-20}]\approx 2^{121}$ 5-round SC2000 encryptions. Therefore, the attack has a total time complexity of at most $2^{125.68}+2^{121}\approx 2^{125.74}$ 5-round SC2000 encryptions and has a success probability of $\Phi(\frac{\sqrt{16\times 2^{1.15}}-\Phi^{-1}(1-2^{-(40-15)})}{\sqrt{2^{1.15}+1}})\approx 62\%$ by Theorem 1.

Below we consider the case when setting $r=30$. In the extremely conservative circumstance, among the $2^r=2^{30}$ values for $(K^6_{0,70},\ldots,K^6_{0,89},K^6_{0,102},\ldots,K^6_{0,121})$ in Step 3 we have all the 2^{20} possible values for $(K^6_{0,102},\ldots,K^6_{0,121})$, because $(K^6_{0,102},\ldots,K^6_{0,121})$ involves only twenty subkey bits; and for each of the 2^{20} values of $(K^6_{0,102},\ldots,K^6_{0,121})$, on average there are about 2^{10} values for $(K^6_{0,70},\ldots,K^6_{0,89})$, that is we have $h=20-\log_2^{2^{10}}=10$ bits information of $ek[50]$. Thus, there are 2^{32} possible values for $ek[51]$, and hence, Step 3 has an expected time complexity of approximately $2^{32}\times[(5\times 2^{96}+4\times 2^{96-10})\times\frac{1}{4}\times\frac{1}{5}+2^{96-10}]\approx 2^{126}$ 5-round SC2000 encryptions, and the attack has a total time complexity of approximately $2^{125.68}+2^{126}\approx 2^{126.85}$ 5-round SC2000 encryptions (in the extremely conservative circumstance), with a success probability of $\Phi(\frac{\sqrt{16\times 2^{1.15}}-\Phi^{-1}(1-2^{-(40-30)})}{\sqrt{2^{1.15}+1}})\approx 94.5\%$. Nevertheless, we can expect there are about 2^{15} possible values for $(K^6_{0,70},\ldots,K^6_{0,89})$ and about 2^{15} possible values for $(K^6_{0,102},\ldots,K^6_{0,121})$. Then, there are $2^{15}\times 2^{12}=2^{27}$ possible values for $ek[51]$, and we have $h=20-\log_2^{2^{15}}=5$ bits information of $ek[50]$. Thus, Step 3 has an expected time complexity of approximately $2^{27}\times[(5\times 2^{96}+4\times 2^{96-5})\times\frac{1}{4}\times\frac{1}{5}+2^{96-5}]\approx 2^{121.21}$ 5-round SC2000 encryptions, and the attack has a total time complexity of approximately $2^{125.68}+2^{121.21}\approx 2^{125.75}$ 5-round SC2000 encryptions, with a success probability of 94.5%.

6 Conclusions

SC2000 is one of the CRYPTREC e-Government Recommended Ciphers, which has a total of 6.5 rounds if a 128-bit key is used. In this paper we have described a few 4.75-round differential characteristics with a probability of larger than 2^{-128}. Finally, we have presented a differential attack on 5-round SC2000 when used with 128 key bits. The presented attack is theoretical, like most cryptanalytic attacks on block ciphers; and the attack does not threat the security of the full SC2000 cipher, for it has 6.5 rounds. Anyway, from a cryptanalytic view it suggests for the first time that the safety margin of SC2000 with a 128-bit key decreases within one and a half rounds.

Acknowledgments The author is very grateful to Prof. Chris Mitchell and the anonymous referees for their comments on earlier versions of the paper.

References

[1] Lu J. Differential attack on five rounds of the SC2000 block cipher. In *Proc. INSCRYPT 2009*, Beijing, China, Dec. 12-15, 2009, pp.50-59.

[2] Shimoyama T, Yanami H, Yokoyama K, Takenaka M, Itoh K, Yajima J, Torii N, Tanaka H. The block cipher SC2000. In *Proc. FSE 2001*, Yokohama, Japan, Apr. 2-4, 2001, pp.312-327.

[3] Fujitsu Laboratories. http://jp.fujitsu.com/group/labs/en/techinfo/technote/crypto/sc2000.html.

[4] Cryptography research and evaluation committees — CRYPTREC report 2002.

[5] Biham E, Shamir A. Differential Cryptanalysis of the Data Encryption Standard. Springer-Verlag, 1993.

[6] Raddum H, Knudsen L R. A differential attack on reduced-round SC2000. In *Proc. SAC 2001*, Ontario, Canada, Aug. 16-17, 2001, pp.190-198.

[7] Biham E, Dunkelman O, Keller N. New results on boomerang and rectangle attacks. In *Proc. FSE 2002*, Leuven, Belgium, Feb. 4-6, 2002, pp.1-16.

[8] Wagner D. The boomerang attack. In *Proc. FSE 1999*, Rome, Italy, Mar. 24-26, 1999, pp.156-170.

[9] Kelsey J, Kohno T, Schneier B. Amplified boomerang attacks against reduced-round MARS and Serpent. In *Proc. FSE 2000*, New York, USA, Apr. 10-12, 2000, pp.75-93.

[10] Biham E, Dunkelman O, Keller N. The rectangle attack — Rectangling the Serpent. In *Proc. EUROCRYPT 2001*, Innsbruck, Austria, May 6-10, 2001, pp.340-357.

[11] Dunkelman O, Keller N. Boomerang and rectangle attacks on SC2000. In *the 2nd Open NESSIE Workshop*, Surrey, UK, Sept. 12-13, 2001.

[12] Yanami H, Shimoyama T, Dunkelman O. Differential and linear cryptanalysis of a reduced-round SC2000. In *Proc. FSE 2002*, Leuven, Belgium, Feb. 4-6, pp.34-48.

[13] Matsui M. Linear cryptanalysis method for DES cipher. In *Proc. EUROCRYPT 1993*, Lofthus, Norway, May 23-27, 1993, pp.386-397.

[14] Biham E, Shamir A. Differential cryptanalysis of DES-like cryptosystems. In *Proc. CRYPTO 1990*, Santa Barbara, USA, Aug. 11-15, 1990, pp.2-21.

[15] Data encryption standard (DES), FIPS-46. National Institute of Standards and Technology (NIST), 1977.

[16] Murphy S. The cryptanalysis of FEAL-4 with 20 chosen plaintexts. *Journal of Cryptology*, 1990, 2(3): 145-154.

[17] Shimizu A, Miyaguchi S. Fast data encipherment algorithm FEAL. In *Proc. EUROCRYPT 1987*, Amsterdam, The Netherlands, Apr. 13-15, 1987, pp.267-278.

[18] Lai X, Massey J L, Murphy S. Markov ciphers and differential cryptanalysis. In *Proc. EUROCRYPT 1991*, Brighton, UK, Apr. 8-11, pp.17-38.

[19] Handschuh H, Naccache D. SHACAL. In *the First Open NESSIE Workshop*, Leuven, Belgium, Nov. 13-14, 2000.

[20] Selçuk A A. On probability of success in linear and differential cryptanalysis. *Journal of Cryptology*, 2008, 21(1): 131-147.

[21] Advanced encryption standard (AES), FIPS-197. National Institute of Standards and Technology (NIST), 2001.

Ji-Qiang Lv received the B.Sc. degree in applied mathematics from Yantai University, China, in July 2000, the M.Eng. degree in information and communication engineering from Xidian University, China, in March 2003, and the Ph.D. degree from the University of London, UK, in July 2008. He was a government officer in the Intellectual Property Office of Department of Science & Technology of Shandong Province, China, a research assistant in Information and Communication University, Korea, a software engineer in ONETS Wireless & Internet Security Co. Ltd., China, and the Beijing R&D Institute of Huawei Technologies, Co. Ltd., China, and a postdoctoral researcher in Eindhoven University of Technology, The Netherlands. Currently, he is a postdoctoral researcher in the Département d'Informatique, École Normale Supérieure, France, and his research interests center on cryptology and information security.

Appendix

Table A. Differential Distribution Table of the S_5 S-Box

Input	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	32	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	0
1	0	0	0	2	2	2	0	0	2	0	0	2	0	0	2	0	2	2	2	0	2	0	2	0	2	0	0	2	2	0	2	2
2	0	0	0	2	2	2	2	0	2	0	2	0	2	2	0	0	2	0	0	2	0	0	0	0	0	2	2	2	0	2	2	2
3	0	0	0	0	0	0	2	0	0	0	2	2	2	2	2	0	0	2	2	2	2	0	2	0	2	2	2	0	2	2	0	0
4	0	0	0	2	2	0	2	2	0	2	0	0	0	2	0	0	2	0	2	0	2	2	2	2	0	0	2	2	2	2	0	0
5	0	0	0	0	0	2	2	2	2	2	0	2	0	2	2	0	0	2	0	0	0	2	0	2	2	0	2	0	0	2	2	2
6	0	0	0	0	0	2	0	2	2	2	2	0	2	0	0	0	0	0	2	2	2	2	2	2	0	2	0	0	2	0	2	2
7	0	0	0	2	2	0	0	2	0	2	2	2	2	0	2	0	2	2	0	2	0	2	0	2	2	2	0	2	0	0	0	0
8	0	2	2	2	2	2	2	0	0	2	2	0	2	0	2	0	0	2	0	0	2	0	2	2	0	0	0	0	0	2	0	2
9	0	2	2	0	0	0	2	0	2	2	2	2	2	0	0	0	2	0	2	0	0	0	0	2	2	0	0	2	2	2	2	0
10	0	2	2	0	0	0	0	0	2	2	0	0	0	2	2	0	2	2	0	2	2	0	2	2	0	2	2	2	0	0	2	0
11	0	2	2	2	2	2	0	0	0	2	0	2	0	2	0	0	0	0	2	2	0	0	0	2	2	2	2	0	2	0	0	2
12	0	2	2	0	0	2	0	2	0	0	2	0	2	2	2	0	2	2	2	0	0	2	0	0	0	0	2	2	2	0	0	2
13	0	2	2	2	2	0	0	2	2	0	2	2	2	2	0	0	0	0	0	0	2	2	2	0	2	0	2	0	0	0	2	0
14	0	2	2	2	2	0	2	2	2	0	0	0	0	0	2	0	0	2	2	2	0	2	0	0	0	2	0	0	2	2	2	0
15	0	2	2	0	0	2	2	2	0	0	0	2	0	0	0	0	2	0	0	2	2	2	2	0	2	2	0	2	0	2	0	2
16	0	2	0	0	2	2	2	0	2	2	2	2	0	2	0	2	0	2	0	2	2	2	0	0	0	0	0	2	2	0	0	0
17	0	2	0	2	0	0	2	0	0	2	2	0	0	2	2	2	2	0	2	2	0	2	2	0	2	0	0	0	0	0	2	2
18	0	2	0	2	0	0	0	0	0	2	0	2	2	0	0	2	2	2	0	0	2	2	0	0	0	2	2	0	2	2	2	2
19	0	2	0	0	2	2	0	0	2	2	0	0	2	0	2	2	0	0	2	0	0	2	2	0	2	2	2	2	0	2	0	0
20	0	2	0	2	0	2	0	2	2	0	2	2	0	0	0	2	2	2	2	2	0	0	2	2	0	0	2	0	0	2	0	0
21	0	2	0	0	2	0	0	2	0	0	2	0	0	0	2	2	0	0	0	2	2	0	0	2	2	0	2	2	2	2	2	2
22	0	2	0	0	2	0	2	2	0	0	0	2	2	2	0	2	0	2	2	0	0	0	2	2	0	2	0	2	0	0	2	2
23	0	2	0	2	0	2	2	2	2	0	0	0	2	2	2	2	2	0	0	0	2	0	0	2	2	2	0	0	2	0	0	0
24	0	0	2	2	0	0	0	0	2	0	0	2	2	2	2	2	0	0	0	2	0	2	2	2	0	0	0	2	2	2	0	2
25	0	0	2	0	2	2	0	0	0	0	0	0	2	2	0	2	2	2	2	2	2	2	0	2	2	0	0	0	0	2	2	0
26	0	0	2	0	2	2	2	0	0	0	2	2	0	0	2	2	2	0	0	0	0	2	2	2	0	2	2	0	2	0	2	0
27	0	0	2	2	0	0	2	0	2	0	2	0	0	0	0	2	0	2	2	0	2	2	0	2	2	2	2	2	0	0	0	2
28	0	0	2	0	2	0	2	2	2	2	0	2	2	0	2	2	2	0	2	2	2	0	0	0	0	0	2	0	0	0	0	2
29	0	0	2	2	0	2	2	2	0	2	0	0	2	0	0	2	0	2	0	2	0	0	2	0	2	0	2	2	2	0	2	0
30	0	0	2	2	0	2	0	2	0	2	2	2	0	2	2	2	0	0	2	0	2	0	0	0	0	2	0	2	0	2	2	0
31	0	0	2	0	2	0	0	2	2	2	2	0	0	2	0	2	2	2	0	0	0	0	2	0	2	2	0	0	2	2	0	2