Analysis of two Attacks on Reduced-Round Versions of the SMS4

Deniz Toz[1] and Orr Dunkelman[2,3]

[1] Middle East Technical University
Institute of Applied Mathematics, Ankara, Turkey
deniz.toz@gmail.com
[2] Katholieke Universiteit Leuven
Department of Electronical Engineering ESAT SDC-COSIC
and
Interdisciplinary Institute for BroadBand Technology (IBBT)
Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
orr.dunkelman@esat.kuleuven.be
[3] École Normale Supérieure
Département d'Informatique,
CNRS, INRIA
45 rue d'Ulm, 75230 Paris, France.

Abstract. SMS4 is a 128-bit block cipher used in WAPI (the Chinese national standard for wireless networks). Up until recently, the best attacks on SMS4 known, in terms of the number of rounds, were the rectangle attack on 14 rounds and the impossible differential attack on 16 rounds (out of 32 rounds) presented by Lu. While analyzing them, we noticed that these attacks have flaws and that their complexity analysis is inaccurate. In this paper we make a more comprehensive analysis of these attacks and further improve these results.

1 Introduction

SMS4 [1] is a Generalized Feistel Network (GFN) cipher, specified in the Wireless Authentication and Privacy Infrastructure (WAPI), which is mandatory in wireless networks in China. The cipher has block size of 128 bits, and each block is processed in 32 rounds using a secret key of 128 bits long.

The Chinese Standards Association (SAC) submitted WAPI to ISO for recognition as an international standard, at about the same time as the IEEE 802.11i standard. As a result, SMS4 was the subject of an extensive international debate since its introduction. Despite that, up until recently little cryptanalysis of SMS4 was performed. The previously published cryptanalytic results are the differential fault analysis presented in [11], the integral attack on 13 rounds [8], the rectangle attack on 14 rounds and the impossible differential attack on 16 rounds [9], the rectangle attack on 16 rounds and the differential attack on 21 rounds [12], and finally the rectangle attack on 18 rounds, the differential attack and linear attack on 22 rounds [6][1].

[1] We note that the results in [6, 12] were found independently of our line of research.

The cryptanalytic results on SMS4, on which we focus are those of [9]. The proposed rectangle attack on 14 rounds of SMS4 uses $2^{121.82}$ chosen plaintexts and has a claimed time complexity of $2^{116.66}$ 14-round SMS4 computations[2]. The impossible differential attack on 16-round SMS4 from [9] uses 2^{105} chosen plaintexts and its time complexity is conjectured to be 2^{107} 16-round SMS4 computations.

While verifying the results of [9], we found several flaws and possible improvements. In this paper, we show that the actual probability of the 12-round rectangle distinguishers of [9] is $2^{-230.71}$, rather than the claimed probability of $2^{-237.64}$. We also present better 12-round rectangle distinguishers with probability of $2^{-209.78}$. Moreover, we show that the claimed time complexity of the rectangle attack of [9] is flawed due to the deficient process of obtaining candidate quartets, which is not considered in the original time complexity analysis. Therefore, given our improved distinguishers and refined analysis, we present a 14-round rectangle attack that uses $2^{106.89}$ chosen plaintexts pairs and has running time of $2^{107.89}$ encryptions for obtaining the data, $2^{107.89}$ memory accesses to find the pairs, and $2^{87.97}$ encryptions for the analysis. Similarly, we identify several flaws in the impossible differential attack of [9]. We first show that more data is needed than the claimed figures, and then we point out a delicate issue concerning the running time of this attack. We then follow to suggest a corrected attack with data complexity of $2^{117.06}$ chosen plaintexts and time complexity of $2^{117.06}$ encryptions for obtaining the data, $2^{132.06}$ memory accesses for the preliminary elimination, and $2^{95.09}$ encryptions for the analysis.

Independent of our research, SMS4 was analyzed also in [6, 12]. A rectangle attack on 16 rounds of SMS4 which requires 2^{124} chosen plaintexts with a time complexity of 2^{116} encryptions, and a differential attack on 21 rounds of SMS4 with data and time complexities of 2^{118} and $2^{112.83}$, respectively are presented in [12]. These results are improved in [6], by using the early abort technique, to a rectangle attack on 18 rounds of SMS4 with a data complexity of 2^{120} and time complexity of $2^{116.83}$, a differential attack on 22 rounds of SMS4 with a data complexity of 2^{118} chosen plaintexts, and a time complexity of $2^{125.71}$ encryptions. Also, a linear attack on 22 rounds of SMS4 which has data complexity of 2^{117} known plaintexts, and time complexity of $2^{109.86}$ encryptions is described in [6].

This paper is organized as follows: In Section 2, we give a brief description of the SMS4 cipher and its properties. In Section 3, we give an overview of the rectangle attack, followed by the previous rectangle attack of [9] on SMS4. Then, we present our observations and improvements for this attack on SMS4. In Section 4, we follow the same outline for the impossible differential attack. Finally, we conclude this paper and summarize our findings in Section 5.

[2] This is the claimed time complexity in [9], but in fact the actual number should be $2^{121.82}$, which is the time required to obtain the ciphertexts to perform attack, according to [9].

2 A Description of SMS4

2.1 Notation

Throughout this paper, we will use the following notation. Each 128-bit *block* is composed of four 32-bit *words* (X_0, X_1, X_2, X_3). Note that the words and blocks are in a "Chinese"-endian order (i.e., the most significant bit is the leftmost bit numbered 0, and the least significant bit is bit 31 for a 32-bit word). Similarly, the most significant byte of a word is the leftmost byte numbered 0, and least significant byte is numbered 3. We denote the bit rotation of the word w by r positions to the left by $w \lll r$; e_j denotes a word whose all positions except the j-th bit are zero and

$$e_{i_1,\ldots,i_j} = e_{i_1} \oplus \ldots \oplus e_{i_j} \text{ for } 0 \leq i_1, \ldots, i_j \leq 31$$

2.2 The SMS4 Cipher

SMS4 [1] accepts a 128-bit plaintext $P = (P_0, P_1, P_2, P_3)$ and a 128-bit user key as inputs, and is composed of 32 rounds. In each round, the least significant three bytes of the state are xored with the round key and the result passes the S transformation. The S transformation uses an 8-bit to 8-bit bijective SBox four times in parallel to process each byte, then the concatenated bytes are processed using a linear transformation L. Let $X_i = (X_{i,0}, X_{i,1}, X_{i,2}, X_{i,3})$ and $X_{i+1} = (X_{i+1,0}, X_{i+1,1}, X_{i+1,2}, X_{i+1,3})$ denote the 128-bit input and output to the i-th round, respectively. Then the round function may be formally described by the following equations:

$$\begin{aligned}
X_{i+1,0} &= X_{i,1} \\
X_{i+1,1} &= X_{i,2} \\
X_{i+1,2} &= X_{i,3} \\
X_{i+1,3} &= X_{i+1,0} \oplus L(S(X_{i,1} \oplus X_{i,2} \oplus X_{i,3} \oplus RK_i))
\end{aligned}$$

where the S transformation uses the SBox given in [1] and L is the linear transformation:

$$L(x) = x \oplus (x \lll 2) \oplus (x \lll 10) \oplus (x \lll 18) \oplus (x \lll 24) \text{ where } x \in Z_2^{32}$$

The transformation $L \circ S$ is named T in the specification document. RK_i is the 32-bit round sub key for the i-th round, obtained from the key schedule. Decryption is identical to the encryption except for the order of the subkeys, which are used in the reverse order.

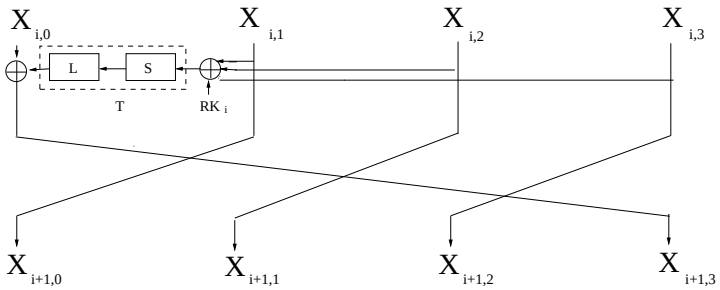

Fig. 1: Round Function

Key Schedule: The key schedule is similar to the encryption function. The only difference is that instead of using the linear transformation L, the following linear transformation L' is used:

$$L'(x) = x \oplus (x \lll 13) \oplus (x \lll 23) \text{ where } x \in Z_2^{32}$$

In addition, the user supplied key K is xored with a system parameter, FK. The subkey RK_j of the j-th round is computed as follows:

$$\begin{aligned} FK &= (0xA3B1BAC6, 0x56AA3350, 0x677D9197, 0xB27022DC) \\ k &= (k_0, k_1, k_2, k_3) = K \oplus FK \\ RK_j &= k_{j+4} = k_j \oplus L'(S(k_{j+1} \oplus k_{j+2} \oplus k_{j+3} \oplus CK_j)) \end{aligned}$$

where $CK_j = (ck_{j,0}, ck_{j,1}, ck_{j,2}, ck_{j,3})$ and $ck_{j,k} = 28j + 7k \pmod{256}$.

2.3 Properties and Definitions

Since SMS4 uses a bijective SBox, thus, $S(\Delta x) = 0$ *if and only if* $\Delta x = 0$. The difference distribution table (DDT) of the SBox contains exactly 127 nonzero output differences for a given nonzero input difference. Only one of these values has probability of 2^{-6} while the other 126 remaining nonzero values have probability of 2^{-7}.

The following definitions are used for observing the propagation of any nonzero input difference to the other rounds. In [9], it is not clearly stated to what these sets refer, and the formulas contain typos. Thus, the reader may find the original terminology confusing. Therefore, we rewrite the equations defining these sets, using the same names for the sets (but with a clearer representation).

Given the input difference $(0, e_\Lambda, e_\Lambda, e_\Lambda)$ to the n-th round, where Λ is an arbitrary but nonempty subset of $\{0, 1, \ldots, 31\}$, the set $\theta(e_\Lambda)$ is composed of all the 32-bit differences that an input difference e_Λ to the T function can cause:

$$\theta(e_\Lambda) = \{x | x = L(\Delta d_1), \Pr[S(e_\Lambda) \to \Delta d_1] > 0 \text{ for } x, e_\Lambda \in Z_2^{32}\}$$

Now, the input difference to the $(n+1)$-th round is $(e_\Lambda, e_\Lambda, e_\Lambda, X)$ where $X \in \theta(e_\Lambda)$. The $\Upsilon(e_\Lambda, X)$ is the set of all 32-bit differences, that an input difference X to the T function may cause after an xor with e_Λ.

$$\Upsilon(e_\Lambda, X) = \{y | y = L(\Delta d_2) \oplus e_\Lambda, \Pr[S(X) \rightarrow \Delta d_2] > 0 \text{ for } X \in \theta(e_\Lambda),\ y, e_\Lambda \in Z_2^{32}\}$$

Similarly, the input difference to the $(n+2)$-th round is of the form $(e_\Lambda, e_\Lambda, X, Y)$ where $X \in \theta(e_\Lambda)$ and $Y \in \Upsilon(e_\Lambda, X)$. The corresponding 32-bit output differences, caused by an input difference $e_\Lambda \oplus X \oplus Y$ to T are denoted by the set $\Pi(e_\Lambda, X, Y)$.

$$\Pi(e_\Lambda, X, Y) = \{z | z = L(\Delta d_3) \oplus e_\Lambda, \Pr[S(e_\Lambda \oplus X \oplus Y) \rightarrow \Delta d_3] > 0 \text{ for } z, e_\Lambda \in Z_2^{32}\}$$

Finally, the input difference to the $(n+3)$-th round is of the form (e_Λ, X, Y, Z) where $X \in \theta(e_\Lambda)$, $Y \in \Upsilon(e_\Lambda, X)$, and $Z \in \Pi(e_\Lambda, X, Y)$. The set of 32-bit differences after the XOR operation in the $(n+3)$-th round by an input difference $X \oplus Y \oplus Z$ to T is denoted by the set $\Omega(X, Y, Z)$.

$$\Omega(X, Y, Z) = \{w | w = L(\Delta d_4) \oplus e_\Lambda, \Pr[S(X \oplus Y \oplus Z) \rightarrow \Delta d_4] > 0 \text{ for } w \in Z_2^{32}\}$$

3 The Rectangle Attack

The amplified boomerang and rectangle attacks [2] are a chosen plaintext attacks, which evolved from the boomerang attack [10]. The main idea in these attacks is to use two short differential characteristics with high probabilities instead of one long characteristic with a lower probability. The only difference is that the boomerang attack generates a *quartet* at an intermediate value halfway through the cipher, whereas the rectangle attack looks for quartets within a given set of pairs.

For this purpose, the block cipher E is treated as a cascade of two sub-ciphers E_0 and E_1 (i.e, $E = E_1 \circ E_0$). Assume that a differential characteristics $\Delta \rightarrow \Delta^*$ with probability p for E_0, and $\nabla^* \rightarrow \nabla$ with probability q for E_1 are known. The boomerang attack is based on generating right quartets (P_1, P_2, P_3, P_4) which satisfy a set of relations:

1. $P_1 \oplus P_2 = \Delta = P_3 \oplus P_4$.
2. $E_0(P_1) \oplus E_0(P_2) = \Delta^* = E_0(P_3) \oplus E_0(P_4)$.
3. $E_0(P_1) \oplus E_0(P_3) = \nabla^* = E_0(P_2) \oplus E_0(P_4)$.
4. $C_1 \oplus C_3 = \nabla = C_2 \oplus C_4$ where $C_i = E_1(E_0(P_i))$.

A right quartet which satisfies the above equations is formed as follows:

1. Choose a random plaintext P_1 and compute $P_2 = P_1 \oplus \Delta$.
2. Ask for the encryptions of P_1 and P_2 to obtain $C_1 = E(P_1)$ and $C_2 = E(P_2)$.
3. Calculate $C_3 = C_1 \oplus \nabla$ and $C_4 = C_2 \oplus \nabla$.
4. Ask for the decryptions of C_3 and C_4 to obtain $P_3 = D(C_3)$ and $P_4 = D(C_4)$.
5. Check whether $P_3 \oplus P_4 = \Delta$.

The amplified boomerang attack is a chosen plaintext attack in which the same differential conditions have to be satisfied. But instead of generating quartets as given above, a set of plaintext pairs with input difference Δ is generated. Then the aim is to find quartets $((P_1, P_2), (P_3, P4))$ such that $C_1 \oplus C_3 = \nabla = C_2 \oplus C_4$ when $P_1 \oplus P_2 = \Delta = P_3 \oplus P_4$ by using birthday paradox.

By a more careful analysis and a better key recovery algorithm, the amplified boomerang attack was evolved into the rectangle attack. For an optimized method of finding the right rectangle quartet, one may refer to [3].

In [2, 10], it is shown that it is possible to use all possible Δ^*'s and ∇^*'s simultaneously. In [2], it is also stated that, if N plaintext pairs with input difference Δ, then the number of expected right quartets is $N^2 2^{-128} \hat{p}^2 \hat{q}^2$ for 128-bit block ciphers, where

$$\hat{p} = \sqrt{\sum_{\Delta^*} Pr^2[\Delta \rightarrow \Delta^*]} \quad \text{and} \quad \hat{q} = \sqrt{\sum_{\nabla} Pr^2[\nabla^* \rightarrow \nabla]}$$

3.1 The Rectangle Attack on 14-Round SMS4 from [9]

The 14-round rectangle attack in [9] uses 12-round rectangle distinguishers with probability[3] $2^{-230.71}$ and requires $2^{121.82}$ chosen plaintexts to attack 14-round SMS4. Let E_0 denote rounds 0 to 7 and let E_1 denote rounds 8 to 11 of SMS4. The differentials used for the 12-round distinguishers of [9] are as follows:

1. **For** E_0: All 8-round differentials of the form $(e_{\psi_1}, e_\psi, e_\psi, e_\psi) \rightarrow (e_{\psi_2}, e_{\psi_3}, e_{\psi_4}, e_{\psi_5})$ where only one byte of e_ψ is nonzero and $e_{\psi_1}, e_{\psi_2} \in \theta(e_\psi)$, $e_{\psi_3} \in \Upsilon(e_\psi, e_{\psi_2})$, $e_{\psi_4} \in \Pi(e_\psi, e_{\psi_2}, e_{\psi_3})$, $e_{\psi_5} \in \Omega(e_{\psi_2}, e_{\psi_3}, e_{\psi_4})$, and e_{ψ_1} is fixed.
2. **For** E_1: All 4-round differentials of the form $(e_\Phi, e_\Phi, e_\Phi, 0) \rightarrow (e_\Phi, e_\Phi, e_\Phi, e_{\Phi_2})$ where only one byte of e_Φ is nonzero and $e_{\Phi_2} \in \theta(e_\Phi)$.

To calculate the overall probability, the sum of the squares of the probabilities of all used differentials is needed. As there are many 8-round differential characteristics, we list the ones that follow the path in Table 1. In Table 2, we list how many differential characteristics of a given probability follow this path.

Therefore, the lower bound can be calculated as:

$$\begin{aligned}\hat{p}^2 &= (2^{-6})^2 \cdot [(2^{-6})^2 + 126 \cdot (2^{-7})^2] \cdot [(2^{-24})^2 + \tbinom{4}{3} \cdot 126 \cdot (2^{-25})^2 \\ &\quad + \tbinom{4}{2} \cdot 126^2 \cdot (2^{-26})^2 + \tbinom{4}{1} \cdot 126^3 \cdot (2^{-27})^2 + 126^4 \cdot (2^{-28})^2]^3 \\ &= 2^{-102.71}\end{aligned}$$

We note that the second differential is a truncated differential with 127 possible output differences and probability one. Therefore:

$$\hat{q}^2 = 1$$

[3] In [9] the probability of these 12-round distinguishers is calculated as $2^{-237.64}$ due to a miscalculation of $\hat{q}$.

e_{ψ_1}		e_{ψ_2} (for a fixed e_ψ)		e_{ψ_3} (for a given e_{ψ_2})		e_{ψ_4} (for a given e_{ψ_2}, e_{ψ_3})		e_{ψ_5} (for a given $e_{\psi_2}, e_{\psi_3}, e_{\psi_4}$)	
No	Pr	No	Pr	No	Pr	No	Pr	No	Pr
1	2^{-6}	1	2^{-6}	1	2^{-24}	1	2^{-24}	1	2^{-24}
		126	2^{-7}	$\binom{4}{3} \cdot 126$	2^{-25}	$\binom{4}{3} \cdot 126$	2^{-25}	$\binom{4}{3} \cdot 126$	2^{-25}
				$\binom{4}{2} \cdot 126^2$	2^{-25}	$\binom{4}{2} \cdot 126^2$	2^{-25}	$\binom{4}{2} \cdot 126^2$	2^{-25}
				$\binom{4}{1} \cdot 126^3$	2^{-27}	$\binom{4}{1} \cdot 126^3$	2^{-27}	$\binom{4}{1} \cdot 126^3$	2^{-27}
				126^4	2^{-28}	126^4	2^{-28}	126^4	2^{-28}

Table 1: The number of differences and their probabilities for the 8-round characteristic.

Probability	2^{-84}	2^{-85}	2^{-86}	2^{-87}	2^{-88}	2^{-89}	2^{-90}
Number	1	$2^{10.678}$	$2^{20.312}$	$2^{29.217}$	$2^{37.514}$	$2^{45.277}$	$2^{52.561}$
Probability	2^{-91}	2^{-92}	2^{-93}	2^{-94}	2^{-95}	2^{-96}	2^{-97}
Number	$2^{59.411}$	$2^{65.872}$	$2^{71.972}$	$2^{77.692}$	$2^{82.920}$	$2^{87.428}$	$2^{90.704}$

Table 2: The number of characteristics and their probabilities for E_0

Thus, the expected number of right rectangle quartets generated by N plaintext pairs is:

$$N^2 \cdot 2^{-128} \cdot \hat{p}^2 \cdot \hat{q}^2 = N^2 \cdot 2^{-230.71}$$

Attack Procedure: The above 12-round distinguishers are used to mount a rectangle attack on 14-round SMS4. Given the 127 input differences[4] $(e_{\Phi}, e_{\Phi}, e_{\Phi}, e_{\Phi_2})$ to round 12, there are 127^5 possible output differences $(e_{\Phi}, e_{\Phi}, e_{\Phi_2}, e_{\Phi_3})$ just after round 12, where $e_{\Phi_3} \in \Upsilon(e_{\Phi}, e_{\Phi_2})$ and 127^9 possible output differences $(e_{\Phi}, e_{\Phi_2}, e_{\Phi_3}, e_{\Phi_4})$, where $e_{\Phi_4} \in \Pi(e_{\Phi}, e_{\Phi_2}, e_{\Phi_3})$. For sake of clarity, we define all these output differences by the set Φ:

$$\Phi = \{(e_{\Phi}, e_{\Phi_2}, e_{\Phi_3}, e_{\Phi_4}) | e_{\Phi_2} \in \Theta(e_{\Phi}), e_{\Phi_3} \in \Upsilon(e_{\Phi}, e_{\Phi_2}), e_{\Phi_4} \in \Pi(e_{\Phi}, e_{\Phi_2}, e_{\Phi_3})\}$$

The proposed attack uses an early abort technique, which allows partially determining whether or not a candidate quartet is a right one by guessing only a small fraction of the subkey, and if not discarding the quartet.

The attack procedure of [9] is as follows:

1. Choose $2^{120.82}$ pairs of plaintexts[5] (P_i, P_i') with input difference $(e_{\psi_1}, e_{\psi}, e_{\psi}, e_{\psi})$

[4] This is the output difference of the distinguisher, and e_{Φ_2} can take any value (of the 127 possible ones). Thus, the probability of the last step is one. In [9], probability is also calculated for this step, leading to a faulty, lower, probability for the distinguisher.

[5] The required number of plaintext pairs is not adapted for the new probability, since it has no effect on our findings.

(a) Obtain the corresponding ciphertext pairs (C_i, C'_i).
(b) Generate all candidate quartets $((C_{i_1}, C'_{i_1}), (C_{i_2}, C'_{i_2}))$.
(c) Check whether $C_{i_1} \oplus C_{i_2} \in \Phi$ and $C'_{i_1} \oplus C'_{i_2} \in \Phi$.

2. For each remaining quartet $((C_{i_1}, C'_{i_1}), (C_{i_2}, C'_{i_2}))$:
   (a) For each pair $((C_{i_1}, C'_{i_1})$ and $(C_{i_2}, C'_{i_2}))$ compute the differences in the 4 bytes of their intermediate values just before the L transformation in round 13, and denote them by $\Delta^{13}_{i_1,i_2}$ and $\Delta'^{13}_{i_1,i_2}$, respectively. (i.e, compute $\Delta^{13}_{i_1,i_2} = L^{-1}(C_{i_1} \oplus C_{i_2})$ and $\Delta'^{13}_{i_1,i_2} = L^{-1}(C'_{i_1} \oplus C'_{i_2})$.
   (b) For j=0 to 3:
      i. Guess the j-th byte of the subkey RK_{13} and partially decrypt every remaining quartet to obtain the j-th byte of their intermediate values just after the S transformation in round 13. Denote them by $((X_{i_1,j}, X_{i_2,j}), (X'_{i_1,j}, X'_{i_2,j}))$.
      ii. Check if $X_{i_1,j} \oplus X_{i_2,j} = \Delta^{13}_{i_1.i_2,j}$ and $X'_{i_1,j} \oplus X'_{i_2,j} = \Delta'^{13}_{i_1.i_2,j}$ and keep only the quartets for which both equalities are satisfied.
3. For each remaining quartet $((T_{i_1}, T'_{i_1}), (T_{i_2}, T'_{i_2}))$ repeat Step 2, for round 12 and RK_{12}.
4. If for a subkey guess (RK_{12}, RK_{13}), there are 6 (or more) remaining quartets try all possible (RK_{10}, RK_{11}) values and perform a trial encryption with one known plaintext/ciphertext pair. If the correct key is not found for all checked (RK_{12}, RK_{13}) values output "failure".

3.2 Improving the 14-Round Attack

By a simple observation, one can conclude that E_0 in the 14-round attack has too many rounds. After round 3, each additional round comes with a cost (in terms of probability) increasing exponentially. Therefore, the attack can be improved by using a shorter characteristic for E_0 with higher probability in exchange for making E_1 longer. We suggest the use of following differential characteristics:

1. **For** E_0: The 6-round differentials $(e_{\psi_1}, e_\psi, e_\psi, e_\psi) \rightarrow (e_\psi, e_\psi, e_{\psi_2}, e_{\psi_3})$ where only one byte of e_ψ is nonzero, $e_{\psi_1}, e_{\psi_2} \in \theta(e_\psi)$, $e_{\psi_3} \in \Upsilon(e_\psi, e_{\psi_2})$, and e_{ψ_1} is fixed.
2. **For** E_1: The 6-round differentials $(e_{\Phi_6}, e_{\Phi_5}, e_\Phi, e_\Phi) \rightarrow (e_\Phi, e_\Phi, e_\Phi, e_{\Phi_2})$ where only one byte of e_Φ is nonzero and $e_{\Phi_5}, e_{\Phi_2} \in \theta(e_\Phi)$, $e_{\Phi_6} \in \Upsilon(e_\Phi, e_{\Phi_5})$.

The details of the rectangle distinguishers for the original attack and the proposed improvement are given in Table 3.

The probability of the new proposed distinguisher can be calculated as follows:

As mentioned earlier in Section 3.1, there exists one possible e_{ψ_2} with probability 2^{-6} and 126 possible e_{ψ_2} values with probability 2^{-7} in round 4. And in round 5, for each of the e_{ψ_2} values, we have one possible e_{ψ_3} with probability 2^{-24}, $\binom{4}{1} \times 126$ possible values with probability 2^{-25}, $\binom{4}{2} \times 126^2$ possible values with probability 2^{-26}, $\binom{4}{1} \times 126^3$ possible values with probability 2^{-27} and 126^4 possible values with probability 2^{-28}. Hence for E_0, we have:

Previous Attack						Improved Attack					
Round	$\Delta X_{i,0}$	$\Delta X_{i,1}$	$\Delta X_{i,2}$	$\Delta X_{i.3}$	Prob	Round	$\Delta X_{i,0}$	$\Delta X_{i,1}$	$\Delta X_{i,2}$	$\Delta X_{i,3}$	Prob
0	e_{ψ_1}	e_{ψ}	e_{ψ}	e_{ψ}	2^{-6}	0	e_{ψ_1}	e_{ψ}	e_{ψ}	e_{ψ}	2^{-6}
1	e_{ψ}	e_{ψ}	e_{ψ}	0	1	1	e_{ψ}	e_{ψ}	e_{ψ}	0	1
2	e_{ψ}	e_{ψ}	0	e_{ψ}	1	2	e_{ψ}	e_{ψ}	0	e_{ψ}	1
3	e_{ψ}	0	e_{ψ}	e_{ψ}	1	3	e_{ψ}	0	e_{ψ}	e_{ψ}	1
4	0	e_{ψ}	e_{ψ}	e_{ψ}	†[a]	4	0	e_{ψ}	e_{ψ}	e_{ψ}	†
5	e_{ψ}	e_{ψ}	e_{ψ}	e_{ψ_2}	†	5	e_{ψ}	e_{ψ}	e_{ψ}	e_{ψ_2}	†
6	e_{ψ}	e_{ψ}	e_{ψ_2}	e_{ψ_3}	†	output	e_{ψ}	e_{ψ}	e_{ψ_2}	e_{ψ_3}	
7	e_{ψ}	e_{ψ_2}	e_{ψ_3}	e_{ψ_4}	†	6	e_{Φ_6}	e_{Φ_5}	e_{Φ}	e_{Φ}	‡[b]
output	e_{ψ_2}	e_{ψ_3}	e_{ψ_4}	e_{ψ_5}		7	e_{Φ_5}	e_{Φ}	e_{Φ}	e_{Φ}	‡
8	e_{Φ}	e_{Φ}	e_{Φ}	0	1	8	e_{Φ}	e_{Φ}	e_{Φ}	0	1
9	e_{Φ}	e_{Φ}	0	e_{Φ}	1	9	e_{Φ}	e_{Φ}	0	e_{Φ}	1
10	e_{Φ}	0	e_{Φ}	e_{Φ}	1	10	e_{Φ}	0	e_{Φ}	e_{Φ}	1
11	0	e_{Φ}	e_{Φ}	e_{Φ}	1	11	0	e_{Φ}	e_{Φ}	e_{Φ}	1
output	e_{Φ}	e_{Φ}	e_{Φ}	e_{Φ_2}		output	e_{Φ}	e_{Φ}	e_{Φ}	e_{Φ_2}	

Table 3: The rectangle attack distinguishers

[a] The probabilities given with dagger are stated in Table 1.

[b] The probability of e_{Φ_5} is equal to the probability of e_{ψ_2}, since they both belong to the same set. Similarly e_{Φ_6} and e_{ψ_3} have the same probability.

$$\begin{aligned}\hat{p}^2 &= (2^{-6})^2 \cdot [(2^{-6})^2 + 126 \cdot (2^{-7})^2] \cdot [(2^{-24})^2 + \tbinom{4}{3} \cdot 126 \cdot (2^{-25})^2 \\ &\quad + \tbinom{4}{2} \cdot 126^2 \cdot (2^{-26})^2 + \tbinom{4}{1} \cdot 126^3 \cdot (2^{-27})^2 + 126^4 \cdot (2^{-28})^2] \\ &= 2^{-46.8881}\end{aligned}$$

Similarly, for E_1, in round 7, we have one possible e_{Φ_5} with probability 2^{-6} and 126 possible e_{Φ_5} with probability 2^{-7}. In round 6, for each of the e_{Φ_5} values, there is one possible e_{Φ_6} with probability 2^{-24}, $\binom{4}{1} \times 126$ possible values with probability 2^{-25}, $\binom{4}{2} \times 126^2$ possible values with probability 2^{-26}, $\binom{4}{1} \times 126^3$ possible values with probability 2^{-27} and 126^4 possible values with probability 2^{-28}. Therefore:

$$\begin{aligned}\hat{q}^2 &= [(2^{-6})^2 + 126 \cdot (2^{-7})^2] \cdot [(2^{-24})^2 + \tbinom{4}{3} \cdot 126 \cdot (2^{-25})^2 + \tbinom{4}{2} \cdot 126^2 \cdot (2^{-26})^2 \\ &\quad + \tbinom{4}{1} \cdot 126^3 \cdot (2^{-27})^2 + 126^4 \cdot (2^{-28})^2] \\ &= 2^{-34.8881}\end{aligned}$$

Thus, the expected number of right quartets generated by N plaintext pairs is:

$$N^2 \cdot 2^{-128} \cdot \hat{p}^2 \cdot \hat{q}^2 = N^2 \cdot 2^{-209.78}$$

In order to have sufficient pairs to perform the improved attack $N = 2^{106.89}$ and from this point on we use this figure throughout the analysis.

A flaw in the preliminary elimination: In the original attack of [9], the time complexity is calculated only for candidates of right quartets after the preliminary elimination (the pairs which enter Step 2), and it does not include the time-complexity of the first elimination itself. However, due to the large amount of data, it is impossible to take all the possible pairs and detect candidates for right quartets immediately. We propose the following algorithm for the detection of right quartet candidates:

A more efficient algorithm for the preliminary elimination:

1. Let $C_i = (C_{i,0}, C_{i,1}, C_{i,2}, C_{i,3})$ and $C'_i = (C'_{i,0}, C'_{i,1}, C'_{i,2}, C'_{i,3})$ denote a ciphertext pair.
2. For each ciphertext pair (C_i, C'_i), insert the following entries into a hash table:
   (a) $C_i||C'_i$ to the bin indexed by $C_{i,0}C_{i,1}||C'_{i,0}C'_{i,1}$.
   (b) $C'_i||C_i$ to the bin indexed by $C'_{i,0}C'_{i,1}||C_{i,0}C_{i,1}$.
3. For every $(e_{\Phi_2}, e'_{\Phi_2})$ pair, where $e_{\Phi_2}, e'_{\Phi_2} \in \Theta(e_\Phi)$:
   (a) initialize for each bin a flag to the state of "active".
   (b) For every "active"bin satisfying $C_{i,0}C_{i,1} \leq C'_{i,0}C'_{i,1}$, go to the corresponding bin $C_{i,0}C_{i,1}||C'_{i,0}C'_{i,1} \oplus e_\Phi e_{\Phi_2}||e_\Phi e'_{\Phi_2} = C_{j,0}C_{j,1}||C'_{j,0}C'_{j,1}$.
      i. For all possible combinations of entries $((C_i, C'_i), (C_j, C'_j))$, check whether:
         A. $C_{i,2} \oplus C_{j,2} \in \Upsilon(e_\Phi, e_{\Phi_2})$ and $C'_{i,2} \oplus C'_{j,2} \in \Upsilon(e_\Phi, e'_{\Phi_2})$
         B. $C_{i,3} \oplus C_{j,3} \in \Pi(e_\Phi, e_{\Phi_2}, e_{\Phi_3})$ and $C'_{i,2} \oplus C'_{j,2} \in \Pi(e_\Phi, e'_{\Phi_2}, e'_{\Phi_3})$
         (Once a condition fails, do not check the remaining conditions.)
      ii. Flag the bins $C_{j,0}C_{j,1}||C'_{j,0}C'_{j,1}$ and $C'_{j,0}C'_{j,1}||C_{j,0}C_{j,1}$ as "analyzed".
4. If two pairs satisfying (i)-(ii) are found, keep them as candidates for right quartets, and apply steps 2–4 of the attack in Section 3.1.

If we have N pairs of ciphertexts, the expected number of entries in each of the 2^{128} bins is $2 \cdot N/2^{128} = N \cdot 2^{-127}$ after Step 2. Therefore, we can form $127^2 \cdot (N/2^{127})^2 = N^2/2^{240.02}$ candidate quartets for each pair of bins. Since we are only forming quartets for the bins whose first two words is smaller than its last two words, we analyze 2^{127} pairs of bins. Flagging in Step (b) also prevents analyzing the same quartet twice. So the number of candidate quartets entering Step (i) is $2^{127}/2 \cdot N^2/2^{240.02} = N^2/2^{114.02}$. Now, the probability of passing Step (A) is $(127^4/2^{32})^2 \approx 2^{-8.08}$ and $2^{-8.08} \cdot N^2/2^{114.02} = N^2/2^{122.1}$ quartets remain. The probabilities of Step (A) and (B) are same, thus we have $N^2/2^{130.18}$ candidates for right quartets in Step 3.

The time complexity of the preliminary elimination is as follows: In Step 2, we have $2N$ memory accesses. In step 4, the number of analyzed quartets, which is the number of required memory access is $N^2/2^{114.02}$. Note that there is no need to go over all bins. In total $2N + N^2/2^{114.02}$ memory accesses is required, and thus for $N = 2^{106.89}$, the total running time of the preliminary elimination is expected to be $2^{107.89} + 2^{99.76}$ memory accesses.

For $N = 2^{106.89}$, we have $(2^{106.89})^2/2^{130.18} = 2^{83.6}$ candidates of right quartets. The time complexity of the attack is dominated by the partial decryptions in Step 2(b) for j=0 in [9]. Therefore, the running time of steps 2-4 of the attack is $2^8 \cdot 2^{83.6} \cdot 1/14 = 2^{87.69}$. The total running time is dominated by Step 1, i.e. $2^{107.89}$ memory accesses.

4 Impossible Differential Attack on 16-Round SMS4

4.1 Impossible Differential Attack

Unlike traditional differential cryptanalysis which tracks differences that propagate through the cipher with high probability, impossible differential cryptanalysis exploits differentials with probability zero.

The attack used in [9] is a combination of the general technique called *miss in the middle*, which is used to construct impossible differential, and the *early abort technique* which partially determines whether or not a candidate pair is useful. The main idea is to find two characteristics with probability one, whose conditions cannot be met together [4]. Then, the key can be found by analyzing the rounds surrounding the impossible event, and guessing the subkeys of these rounds. If the impossible event occurs when a candidate key is used, it is obvious that the suggested key is not the right key.

4.2 The Previous Attack on 16-Round SMS4

The attack uses a set of 12-round impossible differentials of the form $(e_\Gamma, e_\Gamma, e_\Gamma, 0) \not\rightarrow (0, e_\Gamma, e_\Gamma, e_\Gamma)$. Two 6-round differentials with probability one are concatenated for the attack. The first differential used in the construction of the impossible differential is $(e_\Gamma, e_\Gamma, e_\Gamma, 0) \rightarrow (e_\Gamma, x_1, y_1, z_1)$ and the second differential is $(z_2, y_2, x_2, e_\Gamma) \rightarrow (0, e_\Gamma, e_\Gamma, e_\Gamma)$, where $x_i \in \Theta(e_\Gamma)$, $y_i \in \Upsilon(e_\Gamma, x_i)$, $z_i \in \Pi(e_\Gamma, x_i, y_i)$ for $i = 1, 2$. These 12-round differentials are used to conduct an impossible differential attack on SMS4 reduced to 16 rounds by adding two additional rounds before and after the differentials.

The attack uses $\Gamma \subseteq \{0, 1, \ldots, 15\}$. Hence, in round 1, for every Γ, there are 127^2 input differences that may lead to e_Γ as the output difference of T, and they can be generated by 127^6 input differences in round 0, which is denoted by the set $\Sigma_1(\Gamma)$ for each Γ. Similarly, there are 127^2 output differences after round 14, that can be generated by e_Γ, and they cause 127^6 possible output differences after round 15 which, is denoted by the set $\Sigma_2(\Gamma)$ for each Γ.[6]

The attack procedure of [9] is as follows:

1. Choose 2^9 structures of 2^{96} plaintexts each where the most significant 2 bytes of the two rightmost words of the plaintexts in each structure is fixed. (Thus,

[6] In [9], these sets are denoted by Σ_1 and Σ_2. But actually they are not independent of the choice of Γ. Since the attack procedure runs over all possible such Γ's, it is more clear and more accurate to denote these sets as a function of Γ.

Round	$\Delta X_{i.0}$	$\Delta X_{i.1}$	$\Delta X_{i.2}$	$\Delta X_{i.3}$
0	e_Γ	e_Γ	e_Γ	0
1	e_Γ	e_Γ	0	e_Γ
2	e_Γ	0	e_Γ	e_Γ
3	0	e_Γ	e_Γ	e_Γ
4	e_Γ	e_Γ	e_Γ	x_1
5	e_Γ	e_Γ	x_1	y_1
6	e_Γ	x_1	y_1	z_1
6	e_{z_2}	y_2	x_2	e_Γ
7	y_2	x_2	e_Γ	e_Γ
8	x_2	e_Γ	e_Γ	e_Γ
9	e_Γ	e_Γ	e_Γ	0
10	e_Γ	e_Γ	0	e_Γ
11	e_Γ	0	e_Γ	e_Γ
12	0	e_Γ	e_Γ	e_Γ

Table 4: The two 6-round differentials

each structure generates $(2^{96})^2/2 = 2^{191}$ plaintext pairs[7] (P_i, P_j) with the desired input difference $(*, *, e_\Gamma, e_\Gamma)$).

   (a) Obtain the corresponding ciphertext pairs of the structures.
   (b) Choose the pairs that satisfy both $P_i \oplus P_j \in \Sigma_1(\Gamma)$ and $C_i \oplus C_j \in \Sigma_2(\Gamma)$ simultaneously for the same Γ, for all possible Γ's.

2. For all the remaining ciphertext pairs (C_i, C_j):
   (a) Compute the 4-byte difference just before the L transformation in round 15, and denote it by $\Delta^{15}_{i,j}$ (i.e., $\Delta^{15}_{i,j} = L^{-1}(C_{i,3} \oplus C_{j,3})$).
   (b) For l=0 to 3:
      i. Guess the l-th byte of the subkey RK_{15} and partially decrypt (C_i, C_j) to get the l-th byte of the difference just after the S transformation in round 15, denote them by $(T_{i,l}, T_{j,l})$.
      ii. Check if $T_{i,l} \oplus T_{j,l} = \Delta^{15}_{i,j,l}$ and keep the pairs that satisfy the equality.
3. For all the remaining pairs (T_i, T_j):
   (a) Compute the 4-byte difference just before the L transformation in round 14 and denote it by $\Delta^{14}_{i,j}$.
   (b) For l=0 to 1:
      i. Guess the l-th byte of the subkey RK_{14} and partially decrypt (T_i, T_j) to get the l-th byte of their intermediate values just after the S transformation in round 14, denote them by $(Q_{i,l}, Q_{j,l})$.
      ii. Check if $Q_{i,l} \oplus Q_{j,l} = \Delta^{14}_{i,j,l}$ and keep the pairs that satisfy the equality.
4. For all plaintext pairs (P_i, P_j) corresponding to the remaining ciphertexts after Step 3:

[7] In [9], it was claimed that each structure proposes only 2^{190} plaintext pairs.

(a) Compute the 4-byte difference just before the L transformation in round 0 and denote by it $\Delta^0_{i,j}$.
(b) For l=0 to 3:
   i. Guess the l-th byte of the subkey RK_0 and partially encrypt (P_i, P_j) to get the l-th byte of their intermediate values just after the S transformation in round 0, denote them by $(R_{i,l}, R_{j,l})$.
   ii. Check if $R_{i,l} \oplus R_{j,l} = \Delta^0_{i,j,l}$ and keep the pairs that satisfy the equality.

5. For all the remaining pairs (R_i, R_j):
   (a) Compute the 4-byte difference just before the L transformation in round 1 and denote by $\Delta^1_{i,j}$.
   (b) For l=0 to 1:
      i. Guess the l-th byte of the subkey RK_1 and partially encrypt (R_i, R_j) to get the l-th byte of their intermediate values just after the S transformation in round 1. Denote them by $(S_{i,l}, S_{j,l})$.
      ii. Check if $S_{i,l} \oplus S_{j,l} = \Delta^1_{i,j,l}$. If there exists a qualified pair then discard the guess of 96 subkey bits and try another, otherwise proceed to the next step.
6. Guess the user key from the known subkey values, and perform a trial encryption. If a key is suggested then output it. Otherwise, continue with a new guess of RK_{15} (i.e., go to Step 2).

The claimed time complexity of this attack is of 2^{107} 16-round SMS4 computations in [9] and it requires 2^{105} chosen plaintexts.

4.3 Fixing and Improving the 16-Round Attack

Like in the rectangle attack of [9], in the impossible differential attack of [9], the time complexity analysis is also calculated only for candidates of right pairs after preliminary elimination (the pairs which enter Step 2), and it does not include the time complexity of the first elimination itself. Also, the data complexity suggested in [9] is too low.

Data Complexity Issues: Each structure is composed of 2^{96} plaintexts of the form $(*, *, (a, b_i), (c, d_i))$ where $*$ denotes all possible values and a, c denote the chosen constants of the structure, i.e., each structure suggests $(2^{96})^2/2 = 2^{191}$ pairs. In order to have the desired input difference $(*, *, e_\Gamma, e_\Gamma)$, we must have $b_i \oplus b_j = d_i \oplus d_j = \tilde{e_\Gamma}$ where $\tilde{e_\Gamma}$ is the least significant two bytes of e_Γ for each Γ.

In a structure, for each (b_i, b_j, d_i, d_j), there are $2^{64} \cdot 2^{64} = 2^{128}$ possible pairs of plaintexts. There are $2^{16} \cdot 2^{16}/2 = 2^{31}$ possible (b_i, b_j) pairs, and for each pair there exists 2^{16} possible d_i's. Given (b_i, b_j) and d_i, there exists a unique d_j value satisfying the above condition. Hence, only $2^{128} \cdot 2^{31} \cdot 2^{16} = 2^{175}$ of the 2^{191} pairs satisfy the desired input difference.

$\Sigma_1(\Gamma)$ is composed of $127^6 \simeq 2^{42}$ possible input differences for each Γ. Therefore, the probability of a pair to have $P_1 \oplus P_2 \in \Sigma_1(\Gamma)$ is $2^{42}/2^{64} = 2^{-22}$, and $2^{175} \cdot 2^{-22} = 2^{153}$ pairs pass this step. Note that once the plaintext pair is fixed, Γ is also fixed, so does $\Sigma_1(\Gamma)$ and $\Sigma_2(\Gamma)$. Similar to $\Sigma_1(\Gamma)$, $\Sigma_2(\Gamma)$ is composed of $127^6 \simeq 2^{42}$ possible output differences for each Γ. Therefore, the probability of a pair to have $C_1 \oplus C_2 \in \Sigma_2(\Gamma)$ is $2^{42}/2^{128} = 2^{-84}$ and the number of pairs for a given structure passing the Step 1 of the algorithm is $2^{153} \cdot 2^{-84} = 2^{69}$.

Starting with S such structures, the number of plaintext pairs passing the preliminary elimination is $S \cdot 2^{69}$. The probability that a given subkey is discarded by a given structure is thus, $2^{69} \cdot (2^{-7})^{12} = 2^{-15}$, and that it is not discarded by all S structures is $(1 - 2^{-15})^S$. In order to discard all wrong subkeys, we need to make sure that the probability of a wrong key to remain is about 2^{-96}, i.e., $2^{-96} = 1 - (1 - 2^{-15})^S$. Thus for $S = 2^9$, it is not probable to discard most of the subkey guesses.

The number of required structures can be calculated as follows: There are 2^{96} possible subkeys, and $S \cdot 2^{-15}$ pairs are expected for each subkey. In order to have all wrong subkeys with one pair (i.e., suggested by some pair and thus identified as wrong ones), the probability of a wrong key to have no pairs should be less than 2^{-96}. The probability of having no pairs is $e^{-S \cdot 2^{-15}}$. Solving this, we obtain that $S = 2^{21.06}$ structures are needed for the attack.

Algorithm for the detection of candidate pairs: Denote a plaintext by $P_i = (P_{i,0}, P_{i,1}, (a_i, b_i), (c_i, d_i))$, a ciphertext by $C_i = ((w_i, x_i), (y_i, z_i), C_{i,2}, C_{i,3})$.

1. Insert every plaintext-ciphertext pair (P_i, C_i) of each structure, indexed by the least significant 2 bytes of the rightmost two words of the plaintext and the most significant two words of the corresponding ciphertext (i.e, $b_i||d_i||w_i||x_i||y_i||z_i$) into a hash table.
2. For each $\tilde{e_\Gamma}$:
   (a) For every non-empty bin satisfying $b_i < b_j$:
      i. go to the corresponding bin:
      $b_i||d_i||w_i||x_i||y_i||z_i \oplus \tilde{e_\Gamma}||\tilde{e_\Gamma}||e_\Gamma||e_\Gamma = b_j||d_j||w_j||x_j||y_j||z_j$
      (i.e. $w_i = w_j$ and $y_i = y_j$).
      ii. For all possible combinations of entries, pick the plaintext pairs for which:
      A. $P_{i,1} \oplus P_{j,1} \in \theta(e_\Gamma)$
      B. $C_{i,2} \oplus C_{j,2} \in \theta(e_\Gamma)$
      C. $P_{i,0} \oplus P_{j,0} \in \Upsilon(e_\Gamma, P_{i,1} \oplus P_{j,1})$
      D. $C_{i,3} \oplus C_{j,3} \in \Upsilon(e_\Gamma, C_{i,2} \oplus C_{j,2})$
      is satisfied. (If one of them fails, do not check the remaining conditions.)
3. If any pair satisfying (A)-(D) is found, analyze it in Steps 2-6 of the attack.

The time complexity of the preliminary elimination is as follows: In Step 1, we have 2^{96} memory accesses for each structure. There are 2^{96} plaintext-ciphertext

pairs in a structure, therefore, the expected number of entries in each of the 2^{96} bins is 1. The resulting number of required memory accesses for Step 2 is $2^{16} \cdot 2^{96}/2 = 2^{111}$ for a given structure. Therefore, the total number of memory accesses of the algorithm is $S \cdot 2^{111} = 2^{132.06}$.

As mentioned earlier in data complexity issues, the number of pairs passing the preliminary elimination is 2^{69} per structure. Therefore, starting with $S = 2^{21.06}$ structures, which results in $2^{117.06}$ chosen plaintexts, the number of plaintext pairs passing the preliminary elimination is $2^{21.06} \cdot 2^{69} = 2^{90.06}$. The time complexity of the partial encryptions/decryptions in Steps 2(b), 3(b), 4(b) and 5(b) of the algorithm is:

$$\sum_{i=1}^{12} \left(2^{91.06} \cdot \frac{1}{127^{i-1}} \cdot 2^8 \right) \cdot \frac{1}{16} = 2^{95.07}$$

However, the time complexity of the attack is dominated by the $2^{117.06}$ partial encryptions required to obtain the ciphertext pairs in Step 1, and by the $2^{132.06}$ memory accesses performed for the preliminary elimination.

5 Summary

In this paper, we reviewed the rectangle attack on 14-rounds and impossible differential attack on 16-rounds SMS4 presented by Lu. We identified some flaws in the attack algorithms and in the time and data complexity analysis of these attacks. We then followed by correcting and improving these attacks.

We first showed that a better 12-round rectangle distinguisher with probability $2^{-209.78}$ can be found, reducing the amount of required chosen plaintexts to perform the attack from $2^{121.82}$ to $2^{107.89}$. Then, we presented a more efficient algorithm to perform the preliminary elimination.

We also identified some flaws in the previous impossible differential attack of [9]. We first showed that more data is needed for the analysis, and we also presented a more efficient algorithm for the preliminary elimination. The results are summarized in Table 5.

References

1. Beijing Data Security Technology Co. Ltd, *Specification of SMS4* (in Chinese), http://www.oscca.gov.cn/UpFile/200621016423197990.pdf, 2006.
2. E. Biham, O. Dunkelman, N. Keller, *The Rectangle Attack – Rectangling the Serpent*, Advances in Cryptology, Proceedings of EUROCRYPT 2001, Lecture Notes in Computer Science 2045, pp. 340–357, Springer-Verlag, 2001.
3. E. Biham, O. Dunkelman, N. Keller, *New Results on Boomerang and Rectangle Attacks*, Proceedings of FSE '02, Lecture Notes In Computer Science 2365, pp. 1–16, *Springer-Verlag*, 2002.
4. E. Biham, A. Birjukov, A. Shamir, *Miss in the Middle Attacks on IDEA and Khufu*, Proceedings of FSE '99. Lecture Notes in Computer Science 1636, pp. 124–138, *Springer-Verlag*, 1999.

Attack Type	#of Rounds	Complexity		Source
		Data	Time[a]	
Integral Attack	13	2^{16}	2^{114} Enc	[7]
Rectangle Attack	14	$2^{121.82,b}$	$2^{116.66,b}$ Enc	[9]
Impossible Differential Attack	16	$2^{105,b}$	$2^{107,b}$ Enc	[9]
Rectangle Attack [New]	14	$2^{107.89}$	$2^{107.89}$ MA	Section 3.2
Impossible Differential Attack [New]	16	$2^{117.06}$	$2^{132.06}$ MA	Section 4.3
Rectangle Attack	16	2^{125}	2^{116} Enc	[12]
Boomerang Attack	18	2^{120}	$2^{116.83}$ Enc	[6]
Rectangle Attack	18	2^{124}	$2^{112.83}$ Enc	[6]
Differential Attack	21	2^{118}	$2^{126.6}$ Enc	[12]
Linear Attack	22	2^{117}	$2^{109.86}$ Enc	[6]
Differential Attack	22	2^{118}	$2^{125.71}$ Enc	[6]

Enc - Encryptions, MA - Memory Accesses

Table 5: Comparison of the Results for the Existing Attacks.

[a] Time complexities are calculated only for the given algorithms, and they do not include the complexity of obtaining the required data for the attack, which may be higher.

[b] As noted in Sections 3.1 and 4.3, these figures are underestimated.

5. J. Kelsey, T. Kohno, B. Schneier, *Amplified Boomerang Attacks Against Reduced-Round MARS and Serpent*, Proceedings of FSE '00, Lecture Notes in Computer Science 1978, pp. 75–93, *Springer-Verlag*, 2000.
6. T. Kim, J. Kim, S. Hong, J.Sung, *Linear and Differential Cryptanalysis of Reduced SMS4 Block Cipher*, Cryptology ePrint Archive: Report 2008/281, 2008.
7. L.R. Knudsen, D. Wagner,*Integral Cryptanalysis*, Proceedings of FSE '02, Lecture Notes in Computer Science 2365, pp. 112–127, *Springer-Verlag.* 2002.
8. F. Liu et al, *Analysis of the SMS4 Block Cipher*, Proceeding of ACISP 2007, Lecture Notes in Computer Science, pp. 158–170, *Springer-Verlag*, 2007.
9. J. Lu, *Attacking Reduced-Round Versions of the SMS4 Block Cipher in the Chinese WAPI Standart*, Proceedings of ICICS 2007, Lecture Notes in Computer Science 4861, pp. 306–318, *Springer-Verlag*, 2007.
10. David Wagner, *The Boomerang Attack*, Proceedings of FSE '99, Lecture Notes in Computer Science 1636, pp. 156–170, *Springer-Verlag*, 1999.
11. L. Zhang, W. Wu, *Differential Fault Attack on SMS4*, Chinese Journal of Computers, Vol.29, No.9, 2006.
12. L. Zhang, W. Zhang, W. Wu, *Cryptanalysis of Reduced-Round SMS4 Block Cipher*, Proceedings of ACISP '08, Lecture Notes in Computer Science 5107, pp. 216–229, *Springer-Verlag.* 2008.