A Differential Attack on Reduced-Round SC2000*

Håvard Raddum and Lars R. Knudsen

Department of Informatics, University of Bergen, N-5020 Bergen, Norway
haavardr@ii.uib.no, lars@ramkilde.com

Abstract. SC2000 is a 128-bit block cipher with key length of 128, 192 or 256 bits, developed by Fujitsu Laboratories LTD. For 128-bit keys, SC2000 consists of 6.5 rounds, and for 192- and 256-bit keys it consists of 7.5 rounds. In this paper we demonstrate two different 3.5-round differential characteristics that hold with probabilities 2^{-106} and 2^{-107}. These characteristics can be used to extract up to 32 bits of the first and last round keys in a 4.5-round variant of SC2000.

1 Introduction

SC2000 [1,5] is a 128-bit block cipher designed by Fujitsu Laboratories LTD, and accepts keys of 128, 192 and 256 bits. The cipher has been submitted as a candidate for the Nessie project [2], and was presented at the Nessie workshop in Leuven in November 2000, and at FSE2001 in Yokohama in April 2001. In the submission the designers analysed SC2000 against differential cryptanalysis [3], and gave lower bounds on the complexities of a differential attack based on characteristics. However, this search for differential characteristics does not necessarily reveal those with the highest probabilities. We found two different characteristics over 3.5 rounds, which can be used to extract 32 of the bit s in both the first and the last round key in a 4.5-round variant (the definition of a half round will become clear below). These characteristics have probabilities 2^{-106} and 2^{-107}.

The paper is organised as follows. In Section 2 we give a brief description of the SC2000 algorithm. In Section 3 we give the best characteristic we found for the most complicated part of the Feistel round function, used in the cipher round function. In Section 4 we create the different characteristics based on the findings of Section 3. In Section 5 we extract bits from the first and last round keys by using these characteristics, and we conclude in Section 6 with some remarks on the design of SC2000.

* This work was supported by the European Union fund IST-1999-12324 - Nessie. The information in this document is provided as is, and no warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

S. Vaudenay and A. Youssef (Eds.): SAC 2001, LNCS 2259, pp. 190–198, 2001.

2 Description of SC2000

The plaintext block in SC2000 is broken into four 32-bit words. The plaintext words are first XORed with a round key, and then passed through a layer of 32 parallel 4-bit S-boxes. We will call one of these 4-bit S-boxes S4 in this paper. The input to S4 are the bits that are in the same position in each of the words, see Fig. 1. After this the block is XORed with another round key. The XOR with a round key, the 32 executions of S4, and the XOR with a different round key is what we call one half round. The block is now broken into halves, and passed through a two-round Feistel network. The round function in the Feistel network is depicted in Fig. 2. Each of the two words that are sent into the round function are first passed through a layer of two 6-bit and four 5-bit S-boxes, called S6 and S5 respectively. To create diffusion, each word is then regarded as a vector of length 32 over GF(2), and pre-multiplied with M, a 32x32 matrix over GF(2). Let the two words of output from the multiplication of M be a_1 and a_2. These words are now mixed in a linear function to create the two words of output from the Feistel round function, b_1 and b_2, as follows.

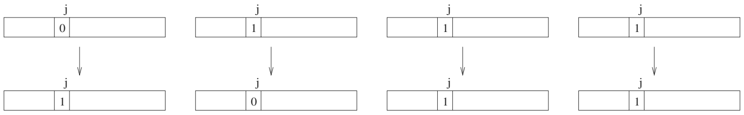

Fig. 1. How the 4–bit S–box works on the bits in position j.

$$b_1 = (a_1 \wedge m) \oplus a_2$$

$$b_2 = (a_2 \wedge \bar{m}) \oplus a_1$$

m is a 32-bit constant, $\bar{m}$ its bitwise complement, and the $\wedge$ denotes the logical AND operation. b_1 and b_2 are now XORed onto the two words in the other half, and the halves are swapped. The other half is then passed through the Feistel round function and XORed onto the first half, but there is no swap after the second Feistel round. This concludes one round of SC2000. For 128-bit keys the cipher consists of six full rounds, plus the first half of the seventh round. For 192- and 256-bit keys the cipher consists of seven full rounds, plus the first half of the eighth round. The constants m used in each round are 55555555_x in the odd numbered rounds and 33333333_x in the even numbered rounds. One round of the cipher is shown in Fig. 3.

We omit the details of the key scheduling. The key schedule in SC2000 is quite complex, and our attack does not depend on how the key schedule works. We note however that the key schedule appears to be very strong, the knowledge of one round key does not seem to leak any information about any other round key, or about the key selected by the user.

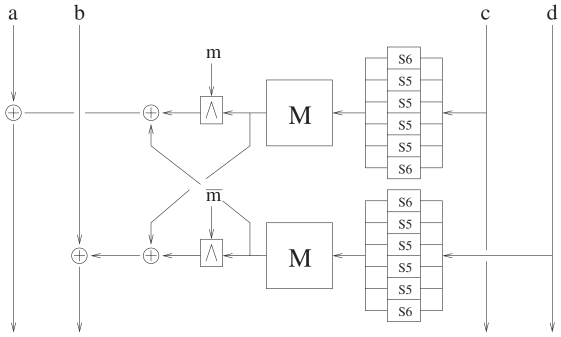

Fig. 2. The Feistel round function.

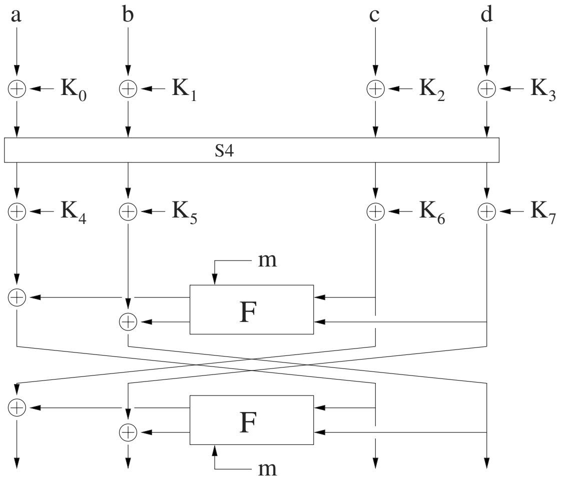

Fig. 3. The round function of SC2000.

3 Searching for Differential Characteristics

In [1] and [5] the designers have performed some differential cryptanalysis of SC2000. However, as shown in the sequel, the designers' search for characteristics was not sufficient, and several differentials exist with probabilities exceeding the bounds of the designers.

In order to explain how we found the differential used in our attack, we first define the *support* of an n-bit string $w = (w_1, w_2, \ldots, w_n)$, written $\chi(w)$ to be the set of coordinates where w has a non-zero value.

$$\chi(w) = \{i | w_i \neq 0\}$$

We concentrated on the two first components of the F-function (see Fig. 2), namely the layer of S5's and S6's, and the multiplication with M. The F-function takes two 32-bit words as input, but they are not mixed with each other until after these two steps are executed, so we focused on only one of the input words. The idea was to find a differential δ with low Hamming weight that mapped to a differential ϵ after the S-boxes and the M-multiplication, such that $\chi(\epsilon) \subseteq \chi(\delta)$. To help us with this we first computed the two differential distribution tables for S5 and S6.

We searched through all 32-bit words of Hamming weight six or less, and for each word we did the following. The word (or differential) ϵ was assumed to be the output of the M-multiplication. Since this is a linear component, we multiplied ϵ with M^{-1} to find the α that would map to ϵ through M. By looking up in the two distribution tables, we then checked whether ϵ could be mapped to α through the layer of the S5 and S6 S-boxes.

We found 11 differentials of Hamming weight five, which only had four of the six S-boxes active and were mapped to themselves with some non-zero probability. None of the ϵ's of weight four or less could not be mapped to themselves, but for each of them we checked how many of the S-boxes that failed to do the required mapping from ϵ to α. We found one ϵ of weight two, namely $\epsilon = 40200000_x$ that mapped to the corresponding $\alpha = f7d30017_x$ in four of the six S-boxes. By adding a 1-bit in the differences going into the two remaining S-boxes, we were able to produce $\delta = \delta_0 = 40220001_x$ of weight four that is mapped to α with probability 2^{-18} (The probabilities are 2^{-5} for the two S6's, and 2^{-4} for the two active S5's). In fact, there are eight different δ's of weight four that can map to ϵ. In one of the two S-boxes that require a non-zero difference we can add the 1-bit in two different ways, and in the other S-box we can add the 1-bit in four different ways. The seven other δ's are

$$\delta_1 = 40220004_x, \delta_2 = 40220010_x, \delta_3 = 40220020_x, \delta_4 = 40300001_x,$$

$$\delta_5 = 40300004_x, \delta_6 = 40300010_x, \delta_7 = 40300020_x$$

Now the idea was to send the differentials 0 and δ into the F-function, and have δ map to ϵ. In the third and last part of the F-function, the AND operation with the fixed masks does not effect the 0 difference. The AND operation applied to ϵ will turn ϵ into $\epsilon_2 = 00200000_x$ when the mask 33333333_x or $aaaaaaaa_x$ is used, and turn ϵ into $\epsilon_4 = 40000000$ when the mask 55555555_x or $cccccccc_x$ is used. Finally, ϵ will be XORed onto the 0 difference and 0 will be XORed onto the difference that is either ϵ_2 or ϵ_4. In total we have the differential characteristics $(\delta, 0) \xrightarrow{F} (\epsilon_4, \epsilon)$ and $(0, \delta) \xrightarrow{F} (\epsilon, \epsilon_2)$ in a round where the mask 55555555_x is used, and the differential characteristics $(\delta, 0) \xrightarrow{F} (\epsilon_2, \epsilon)$ and $(0, \delta) \xrightarrow{F} (\epsilon, \epsilon_4)$ when the mask 33333333_x is used. Each of these characteristics have probability 2^{-18}.

4 Building the One-Round Characteristics

Let us now see how we can use δ and ϵ to create a differential characteristic through several rounds of SC2000.

4.1 Two One-Round Differential Characteristics

The first one-round characteristic, explained below, is shown in Figure 4, where we have omitted the additions of round keys since they do not affect the analysis. s Let $(\delta, 0, 0, 0)$ be the difference in the blocks before the two Feistel rounds.

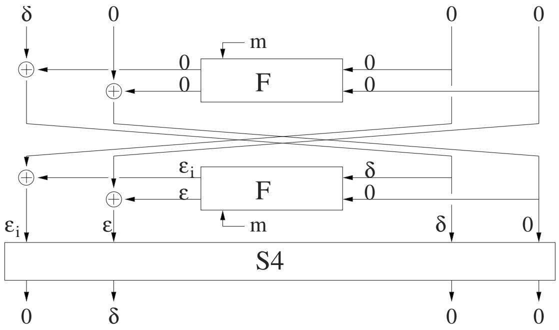

Fig. 4. A one–round differential characteristic with probability 2^{-30}.

First the two rightmost words are sent through the F-function. They have difference $(0, 0)$, so the output will have difference $(0, 0)$ with probability 1. This $(0, 0)$-difference is XORed onto the left half, and the halves are swapped so the difference before the second Feistel round is $(0, 0, \delta, 0)$. The right halves with difference $(\delta, 0)$ are then sent into the F-function, and with probability 2^{-18} the difference after multiplication with M will be $(\epsilon, 0)$. After this ϵ will meet one of the masks 55555555_x or 33333333_x, so the output of F will be (ϵ_2, ϵ) or (ϵ_4, ϵ). These outputs are XORed onto the left halves, and since there is no swap, the difference of the blocks becomes $(\epsilon_2, \epsilon, \delta, 0)$ or $(\epsilon_4, \epsilon, \delta, 0)$ before the S4 layer.

Since δ has weight four and $\chi(\epsilon_i) \subseteq \chi(\epsilon) \subseteq \chi(\delta)$, there will only be four active S-boxes in the layer of the 32 S4's. Two of them will have input difference 2_x, one will have input difference 6_x, and one will have input difference e_x. All the differences 2_x, 6_x and e_x can go to the difference 4_x through S4, each with probability 2^{-3}. So with probability $(2^{-3})^4 = 2^{-12}$ we get the characteristic $(\epsilon_i, \epsilon, \delta, 0) \xrightarrow{S4} (0, \delta, 0, 0)$ through S4. All together, we get the following characteristic with probability $2^{-18} \cdot 2^{-12} = 2^{-30}$.

$$(\delta, 0, 0, 0) \xrightarrow{F-F-S4} (0, \delta, 0, 0)$$

The other useful one-round characteristic is the one that starts with the difference $(0, \delta, 0, 0)$. After the first Feistel round with the swap the difference becomes $(0, 0, 0, \delta)$. With probability 2^{-18} the right half difference $(0, \delta)$ becomes $(0, \epsilon)$ after multiplication with M. The output difference of F will then be (ϵ, ϵ_2) or (ϵ, ϵ_4) and after the XOR with the left halves the difference will be $(\epsilon, \epsilon_i, 0, \delta)$. Two of the input differences to S4 will now be 1_x, one of them will be 9_x and one will be d_x. The 1_x and d_x differences can lead to the difference 8_x with probability 2^{-3}, and the 9_x difference goes to 8_x with probability 2^{-2}. This gives us the following one-round characteristic with probability 2^{-29}.

$$(0, \delta, 0, 0) \overset{F-F-S4}{\longrightarrow} (\delta, 0, 0, 0)$$

4.2 Concatenating the One-Round Characteristics

The differential characteristics above start and end just before the Feistel rounds. The cipher itself begins with the application of the S4 layer, but the characteristics we build by concatenating the one-round characteristics will start after the first half of the first round. The next section explains how to use these characteristics.

The characteristic $(\delta, 0, 0, 0) \overset{F-F-S4}{\longrightarrow} (0, \delta, 0, 0)$ can be concatenated with $(0, \delta, 0, 0) \overset{F-F-S4}{\longrightarrow} (\delta, 0, 0, 0)$. By doing this, we get the following differential characteristic through three and a half rounds with probability 2^{-107}.

$$(\delta,0,0,0) \overset{F-F-S4}{\longrightarrow} (0,\delta,0,0) \overset{F-F-S4}{\longrightarrow} (\delta,0,0,0) \overset{F-F-S4}{\longrightarrow} (0,\delta,0,0) \overset{F-F}{\longrightarrow} (\epsilon,\epsilon_4,\delta,0)$$

The other characteristic is the one starting with input $(0, \delta, 0, 0)$.

$$(0,\delta,0,0) \overset{F-F-S4}{\longrightarrow} (\delta,0,0,0) \overset{F-F-S4}{\longrightarrow} (0,\delta,0,0) \overset{F-F-S4}{\longrightarrow} (\delta,0,0,0) \overset{F-F}{\longrightarrow} (\epsilon_2,\epsilon,\delta,0)$$

This characteristic has probability 2^{-106}.

5 Extracting Bits from the First and Last Round Key

In this section we will explain how to extract up to 32 bits from both the first and last round key in a 4.5-round variant of SC2000.

5.1 How to Find 16 Key Bits of the First and Last Round Key

The characteristics in the previous section do not start with the plaintext difference, but with the difference after the first S4 layer. To use these characteristics, we create structures Σ of 2^{16} plaintexts as follows. Fix the bits going into the 28 S4's that are not affected by δ, and let the 16 bits going into the S4's determined by δ take on all 2^{16} values. Let $\Delta_8 = (\delta, 0, 0, 0)$, $\Delta_4 = (0, \delta, 0, 0)$, $\Omega_1 = (\epsilon, \epsilon_4, 0, \delta)$ and $\Omega_2 = (\epsilon_2, \epsilon, \delta, 0)$. For each plaintext $P \in \Sigma$, the plaintext $P \oplus \Delta_i$ is also in Σ, for $i = 4, 8$. In other words, of the $\binom{2^{16}}{2} \approx 2^{31}$ pairs in Σ there are 2^{15} pairs with difference Δ_4 and 2^{15} pairs with difference Δ_8. Encrypting the plaintexts in Σ through the first S4 layer does not change the structure in any essential way. The 112 fixed bits remain fixed, and the other 16 bits range over all 2^{16} values. So there will be 2^{15} pairs in Σ that have difference Δ_4, and 2^{15} pairs that have difference Δ_8 after encryption through the first S4 layer. A randomly chosen input difference to one S4 will have the possibility to go to the output difference 4_x with probability 1/2, so the probability that a randomly chosen pair of texts from Σ will have the possibility of having difference Δ_4 after S4 is 2^{-4}. By the same argument, the probability that a pair of texts from Σ can have the difference Δ_8 after S4 is approximately 2^{-4}. In total, the probability that a randomly chosen pair of texts from Σ has difference Δ_4 or Δ_8 after S4 is 2^{-3}.

We call a pair of plaintexts that follows either of the two characteristics from Section 4 a *right* pair, and a pair of plaintexts that does not follow any of these characteristics a *wrong* pair.

The probability that a structure contains a right pair is $2^{15} \cdot (2^{-106} + 2^{-107}) = 3 \cdot 2^{-92}$. After encrypting 2^{93} structures, we expect to have $2^{93} \cdot 3 \cdot 2^{-92} = 6$ right pairs among the $2^{31} \cdot 2^{93} = 2^{124}$ pairs we get from the structures. We filter out most of the wrong pairs as follows.

Find potential good pairs by inserting the 2^{16} ciphertexts from one structure in a hash-table according to 20 bits in the first word (see [4]). The ciphertexts in a right pair will be inserted in the same position in the table. If a pair is a right pair, all of the 112 bits corresponding to the inactive S4's must be equal. This gives a filtering factor of 2^{-112}. If a pair of ciphertexts are equal in these 112 bits, check the differences in the four S4's corresponding to δ. If the pair is a right pair, it must be possible that the output difference from the last S4-layer has had input differences Ω_1 or Ω_2. As explained above, a random pair passes this test with probability 2^{-3}. If a ciphertext pair is a right pair, and had difference Ω_1 before the last S4, then the pair of plaintexts must have had the possibility to get the difference Δ_8 after the initial S4. A random pair passes this test with probability 2^{-4}. Likewise, a right pair that has difference Ω_2 before the last S4 must have had difference Δ_4 after the first S4, and the probability that this holds for a random pair is 2^{-4}.

With these steps we have a filtering factor of 2^{-119}. After using this filtering procedure on the 2^{124} different pairs we expect to be left with $2^{124} \cdot 2^{-119} = 32$ pairs, among which we expect six right pairs.

The main part of the work to generate 16 potentially right pairs comes from the 2^{109} encryptions required. The memory requirements to get the 16 potentially good pairs is small. In addition to the potential right pairs, we only need to hold 2^{16} plaintexts and the corresponding 2^{16} ciphertexts in memory at the same time.

The rest of the attack follows along the lines of a standard differential attack. For each of the ciphertexts in the 16 potentially right pairs, guess on the 16 key bits from the last round key corresponding to the active S-boxes. For each guess, decrypt the ciphertext bits in the active S4's, if the decrypted values have one of the input differences Ω_1 or Ω_2, suggest these bits as part of the last round key.

The correct value will be suggested for each right pair. For each Ω_i, there will be 2 - 4 4-bit values suggested for each S-box, and the values suggested by each S-box can be combined in 2^4 - 4^4 different ways. Since we accept both Ω_1 and Ω_2 as input difference, we will get 32 - 512 suggestions for the 16 key bits. The right value will be suggested for every right pair, i.e. six times. The suggestions of the wrong values are expected to be distributed more or less uniformly over the 2^{16} different values, so it is highly unlikely that any wrong value will be suggested six times. Take the most suggested value as the correct bits in the last round key.

We find 16 bits of the first round key in the same manner. Guess on the 16 bits corresponding to the active S4's in the first round, and for each guess, encrypt each pair of plaintexts through the active S4's. If a pair gets the difference Δ_4 or Δ_8, suggest the value as part of the first round key. Again there will be 32 - 512 suggestions for every pair. We expect the correct value to be suggested six times, and the incorrect values to be more or less uniformly distributed over the 2^{16} values. Again we take the most suggested value as the correct one.

5.2 How to Find Another 16 Bits

We can repeat the attack described above for a different δ, say for $\delta_5 = 40300004_x$, to find 16 bits of the first and last round keys. Among the key bits we will find, eight of them will be the same as we found using δ_0, because the two active S-boxes defined by ϵ will overlap in both attacks. So repeating the attack with δ_5 will only yield eight new bits in the two round keys. After this we have found 24 bits of each key. The last eight bits we can get are the ones that correspond to the S4's defined by 00000010_x and 00000020_x. They can be found by repeating the attack with δ's using these S-boxes, like $\delta_2 = 40220010_x$ and $\delta_3 = 40220020_x$. Repeating the attack four times gives an overall complexity of 2^{111}.

6 Conclusions

For a 4.5 round variant of SC2000, we have shown how to find 32 bits of both the first and the last round key, using 2^{111} chosen plaintexts. 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 can not easily deduce the other round keys from them.

This paper may teach us a different lesson, though. Several places in [1], the designers hint that SC2000 can be thought of as an advanced Feistel cipher. The layer of 4-bit S-boxes between every other Feistel round can be regarded as a cryptographically stronger component than the swap of halves found in ordinary Feistel ciphers. This S-box layer certainly gives better confusion than a simple swap, but it introduces another weakness not found in regular Feistel ciphers.

It was shown in [5] that in SC2000 it is possible to have a differential characteristic that feeds every other Feistel round with a 0-difference. This is not possible in a regular Feistel cipher. In this paper we have extended the search done in [5] to two-round iterative characteristics. This resulted in characteristics with higher probabilities than what was found in [5].

Having an S-box layer instead of a swap between some rounds might be a good idea, but one should be careful to make sure that any cryptographically good property of the swap is not lost when replacing it. The designers state that one of the design criteria for S4 is that except for the all-zero difference, an input difference $(\alpha_0, \alpha_1, 0, 0)$ can not lead to an output difference $(\beta_0, \beta_1, 0, 0)$, and an input difference $(0, 0, \alpha_2, \alpha_3)$ can not lead to an output difference $(0, 0, \beta_2, \beta_3)$. This is to make sure that there is some form of "swap" involved when going through S4. However, one should also demand that if α_L and α_R are two non-zero 2-bit values, then the input difference (α_L, α_R) will always lead to an output difference (β_L, β_R) where both β_L and β_R are non-zero.

References

1. http://www.cosic.esat.kuleuven.ac.be/nessie/workshop/
2. http://www.cryptonessie.org/
3. E. Biham and A. Shamir. *Differential Cryptanalysis of the Data Encryption Standard*. Springer Verlag, 1993.
4. L.R. Knudsen and T. Berson. Truncated differentials of SAFER. In Gollmann D., editor, *Fast Software Encryption, Third International Workshop, Cambridge, UK, February 1996, LNCS 1039*, pages 15–26. Springer Verlag, 1995.
5. Shimoyama et al. The Block Cipher SC2000. *Fast Software Encryption, Eighth International Workshop, Yokohama, Japan, April 2001*, preproceedings, pages 326–340.