

# Cryptanalysis of C2

## Extended Abstract

Julia Borghoff\*, Lars R. Knudsen, Gregor Leander, Krystian Matusiewicz\*

Department of Mathematics  
Technical University of Denmark

### Introduction

C2 [1] is the short name for Cryptomeria, a proprietary block cipher defined and licensed by the 4C Entity. It is used for encrypting DVD Audio discs and Secure Digital cards. C2 is a 10-round Feistel cipher with 64-bit blocks and a 56-bit key. The S-box is kept secret and might be considered as part of the secret key.

We present three different attack scenarios for C2

1. The 56-bit key can be chosen by the attacker, who will attempt to determine the values in the secret S-box.
2. The S-box is known to the attacker, who will attempt to determine the value of a secret 56-bit key.
3. The 56-bit key and the S-box are unknown to the attacker, who will attempt to determine the values of both.

The time complexities are respectively  $2^{24}$ ,  $2^{48}$  and  $2^{53.5}$ , where one time unit corresponds to the time it takes to do one encryption with C2.

### The block-cipher C2

C2 [1] is a block cipher with 64-bit blocks and 56-bit keys. It consists of 10 Feistel rounds, each one using a 32-bit round key  $rk_i$ . The round function can be described as

$$\begin{aligned}L_{i+1} &= R_i \\X_i &= (R_i \boxplus rk_i) \oplus 0x2765ca00 \\Z_{i,0..7} &= S[X_{i,0..7}] \\Z_{i,8..15} &= X_{i,8..15} \oplus \text{rotl}_8(Z_{i,0..7}, 1) \\Z_{i,16..23} &= X_{i,16..23} \oplus \text{rotl}_8(Z_{i,0..7}, 5) \\Z_{i,24..31} &= X_{i,24..31} \oplus \text{rotl}_8(Z_{i,0..7}, 2) \\R_{i+1} &= L_i \boxplus (Z \oplus \text{rotl}_{32}(Z_i, 9) \oplus \text{rotl}_{32}(Z, 22)), \quad i = 0, \dots, 9\end{aligned}$$

and is illustrated in Fig. 1. Here  $L_i, R_i$  denotes the left and right word after  $i$  rounds of encryption.

Key scheduling (Fig. 2) produces 10 round keys  $rk_0, \dots, rk_9$  out of 56-bit master key  $K$  in the following way.

$$\begin{aligned}K'_i &= \text{rotl}_{56}(K, 17 \cdot i) , \\rk_i &= K'_{i,0..31} \boxplus (S[K'_{i,32..39} \oplus i] \ll 4), \quad i = 0, \dots, 9 .\end{aligned}$$

Both the round transformation and the key scheduling use an 8-bit secret S-box  $S$ .

---

\* The author is supported by a grant from the Danish research council for Technology and Production Sciences grant number 274-07-0246.

## Recovering the secret S-box with chosen key attack

This attack depends on the details of the key schedule. Some master keys generate only three different inputs to the secret S-box in the key scheduling. We assume that we are allowed to set the key. Then we fix the master key to one of those which just generate three different S-box inputs and guess the possible outputs of the three different S-box inputs.

For each possible guess we generate one plaintext that, under the assumption that our guess is correct, does not trigger any additional entries in the secret S-box for 7 rounds. For such a plaintext, again under the assumption that our guess is correct, we know the output of the encryption process after 7 rounds, i.e.  $(L_7, R_7)$ . The time complexity of this precomputation is approximately  $2^{43.25}$  and the tables uses less than 400 MByte memory. When attacking an actual device or implementation using a secret S-box we proceed as follows. We encrypt each plaintext in the table –corresponding to one possible guess of the three S-box entries– using the device and observe the ciphertext. If our guess is correct we know the output after round 7. It is possible to check if the observed ciphertext fits to our guess of the 7th round output using a system of linear equations. This test will never fail for the right guess and has a (heuristic) probability of accepting a wrong guess with a probability of  $2^{-29}$ . Thus, on average, only the right guess will survive. Using the outlined approach we can recover three S-box entries with  $2^{24}$  encryptions using the actual device and marginal overhead for the test.

After the first three entries have been recovered we continue in a very similar way.

## Key recovery attack for a known S-box

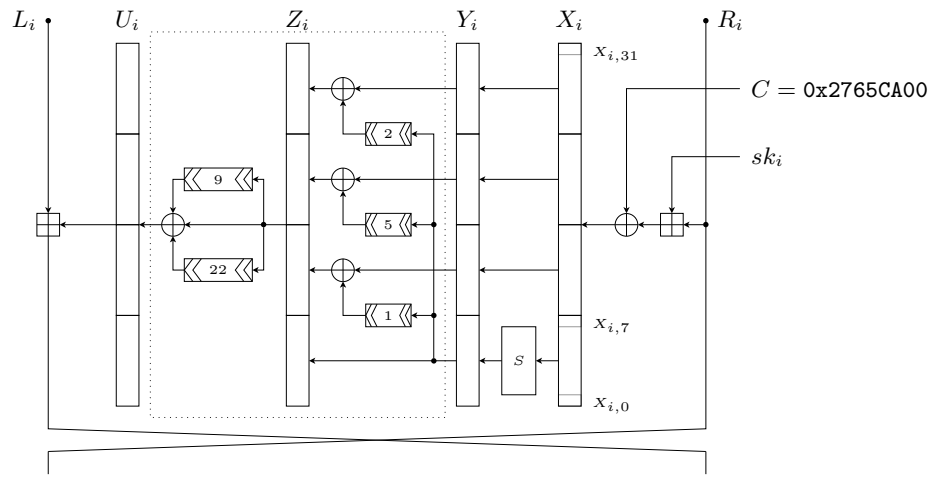
We mount a boomerang attack [4] on the whole cipher using a five round characteristic which is independent of the S-boxes. We find such a characteristic by considering a linearized model of the cipher (i.e. all modular addition are replaced by XORs) and looking for low weight characteristics which have zero input differences to the S-boxes in each round. Using the formulas from [3] one gets a probability of  $2^{-12}$  for independent round inputs and keys which depends on the Hamming weight of the characteristic. Experimentally we can show that the characteristic we are using has probability  $2^{-11.17}$  over five rounds and yields a boomerang with average probability of  $2^{-44.5}$ . Since the characteristic has zero input differences to the S-box in all rounds the only remaining non-linear functions are the modular additions. The possibility of finding boomerangs enables us to test if the differences in the first round propagate according to the characteristics. If not, we do not expect to get any boomerangs. We will use this observation to recover 22 bits of the first round key by a careful analysis of the carries appearing in the addition  $R_0 \boxplus rk_0$ . This method resembles the approach used by Contini and Yin to partially recover HMAC keys using a pseudo-collision differential for MD5 [2]. The total complexity of this attack is  $2^{48}$  and  $2^{44}$  adaptive chosen plaintext/ciphertexts are used.

## Key and S-box recovery with chosen ciphertext attack

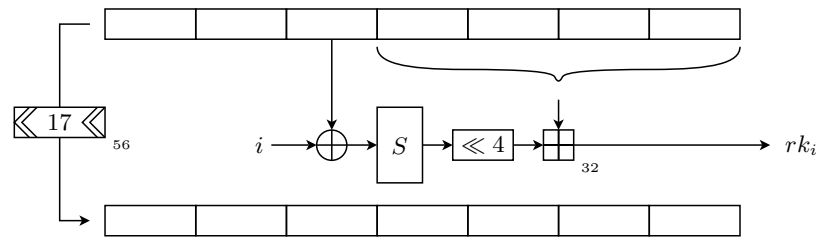
This attack is again based on the boomerang attack. We can recover the least significant 22 bits of the first round key with an average complexity of  $2^{50.59}$  and by turning the boomerang upside down we can recover 22 bits of the last round key. It is then possible to recover the remaining bits of these two round keys and one entry of the secret S-box with an average complexity of  $2^{52}$ . This knowledge enables us to also recover the second round key. The first two and the last round key determine the master key uniquely. After recovering 4 additional S-box entries we can use an approach similar to the S-box recovery attack to recover the remaining entries of the S-box. The total complexity of the attack is  $2^{53.5}$ .

## References

1. C2 Block Cipher Specification, Revision 1.0. <http://www.4Centity.com>, 2003. used to be available online from 4C Entity, can be downloaded e.g. from: <http://edipermadi.files.wordpress.com/2008/08/cryptomeriac2-spec.pdf>.
2. CONTINI, S., AND YIN, Y. L. Forgery and partial key-recovery attacks on HMAC and NMAC using hash collisions. In *Advances in Cryptology – ASIACRYPT 2006* (2006), vol. 4284 of *LNCS*, Springer, pp. 37–53.
3. LIPMAA, H., AND MORIAI, S. Efficient algorithms for computing differential properties of addition. In *Fast Software Encryption – FSE 2001* (2002), vol. 2355 of *LNCS*, Springer, pp. 35–45.



**Fig. 1.** Equivalent description of the round transformation of C2



**Fig. 2.** One step of the key scheduling algorithm generates 32-bit round key  $rk_i$ .

4. WAGNER, D. The boomerang attack. In *Fast Software Encryption – FSE 1999* (1999), vol. 1636 of *LNCS*, Springer, pp. 156–170.