Cryptanalysis of SAFER++*

Alex Biryukov[1**], Christophe De Cannière[1***], and Gustaf Dellkrantz[1,2]

[1] Katholieke Universiteit Leuven, Dept. ESAT/SCD-COSIC,
Kasteelpark Arenberg 10,
B–3001 Heverlee, Belgium
{alex.biryukov, christophe.decanniere}@esat.kuleuven.ac.be
[2] Royal Institute of Technology,
Stockholm, Sweden
d98-gde@nada.kth.se

Abstract. This paper presents several multiset and boomerang attacks on SAFER++ up to 5.5 out of its 7 rounds. These are the best known attacks for this cipher and significantly improve the previously known results. The attacks in the paper are practical up to 4 rounds. The methods developed to attack SAFER++ can be applied to other substitution-permutation networks with incomplete diffusion.

1 Introduction

The 128-bit block cipher SAFER++ [?] is a 7-round substitution-permutation network (SPN), with a 128-bit key (the 256-bit key version[3] has 10 rounds). SAFER++ was submitted to the European pre-standardization project NESSIE [?] and was among the primitives selected for the second phase of this project.

SAFER [?] was introduced by Massey in 1993, and was intensively analyzed since then [?, ?, ?, ?, ?]. This resulted in a series of tweaks which lead to several ciphers in the family: SAFER-K (the original cipher), SAFER-SK (key schedule tweak), SAFER+ (key schedule and mixing transform tweak, increased number of rounds, AES candidate), SAFER++ (faster mixing tweak, key schedule tweak, fewer rounds due to more complex mixing). All these ciphers have common S-boxes derived from exponentiation and discrete logarithm functions. They share the Pseudo-Hadamard-like mixing transforms (PHT), although these are constructed in different ways in the different versions. The ciphers in the family also share the idea of performing key-mixing with two non-commutative operations.

The inventors claim that SAFER++ offers "further substantial improvement over SAFER+" [?]. The main feature is a new 4-point PHT transform in place

* The work described in this paper has been supported in part by the Commission of the European Communities through the IST Programme under Contract IST-1999-12324 and by the Concerted Research Action (GOA) Mefisto-666.

** F.W.O. Researcher, sponsored by the Fund for Scientific Research – Flanders.

*** F.W.O. Research Assistant, sponsored by the Fund for Scientific Research – Flanders

[3] A legacy 64-bit block version was also proposed by the designers but is not studied in this paper.

of the 2-point PHT transform that was used previously in the SAFER family. The authors claim that "all 5-round characteristics have probabilities that are significantly smaller than 2^{-128}" and that SAFER++ is secure against differential cryptanalysis [?] after 5 rounds and against linear cryptanalysis [?] after 2.5 rounds.

The best previous attack on SAFER++ is linear cryptanalysis [?], which can break 3 rounds of SAFER++ (with 128-bit keys) with 2^{81} known plaintexts and 2^{101} steps for a fraction 2^{-13} of keys. For 256-bit keys the attack can break the 3.5-round cipher with 2^{81} known plaintexts and 2^{176} steps for a fraction 2^{-13} of keys.

In this paper we study only the 128-bit key version of SAFER++, since we would like to make our attacks as practical as possible. We design several very efficient multiset attacks on SAFER++ following the methodology of the structural attack on SASAS [?] and inspired by the collision attacks on RIJNDAEL [?]. These multiset attacks can break up to 4.5 rounds of SAFER++ with 2^{48} chosen plaintexts and 2^{94} steps, which is much faster than exhaustive search. Attacking 3 rounds is practical and was tested with an actual implementation running in milliseconds on a PC.

In the second half of the paper we show how to apply a cryptanalytic technique called the boomerang attack [?] to SAFER++. We start from ideas which are applicable to arbitrary SPNs with incomplete diffusion (such as RIJNDAEL, SAFER++ or SERPENT) and then extend our results using special properties of the SAFER S-boxes. The attacks thus obtained are more efficient then those we found via the multiset techniques, are practical up to 4 rounds and were confirmed experimentally on a mini-version of the cipher.

The average data complexity of the 5 round attack is 2^{78} chosen plaintexts/adaptive chosen ciphertexts with the same time complexity, most of which is spent encrypting the data. The attack completely recovers the 128-bit secret key of the cipher and can be extended to 5.5 rounds by guessing 30 bits of the secret key. See Table 1 for a summary of results presented in this paper and their comparison with the best previous attack.

This paper is organized as follows: Section 2 provides a short description of SAFER++ and Section 3 shows some interesting properties of the components. In Sections **??** and **??** we design our multiset attacks on SAFER++. Section **??** describes our application of boomerang techniques to SAFER++ reduced to 5 rounds and shows how to use the middle-round S-box trick to obtain even better results. Finally, Section **??** concludes the paper.

2 Description of Safer++

This section contains a short description of SAFER++. For more details, see [?]. In this paper, eXclusive OR (XOR) will be denoted by $\oplus$, addition modulo 256 by $\boxplus$ and subtraction modulo 256 by $\boxminus$. The notion of difference used is subtraction modulo 256. Throughout this paper we will number bytes and S-boxes from left to right, starting from 0.

Table 1. Comparison of our results with the best previous attack on SAFER++.

Attack	Key size	Rounds	Data[a]	Type[b]	Workload[c]	Memory[a]
Our Multiset attack	128	3 of 7	2^{16}	CC	2^{16}	2^{4}
Our Multiset attack	128	4 of 7	2^{48}	CP	2^{70}	2^{48}
Our Multiset attack	128	4.5 of 7	2^{48}	CP	2^{94}	2^{48}
Our Boomerang attack	128	4 of 7	2^{41}	CP/ACC	2^{41}	2^{40}
Our Boomerang attack	128	5 of 7	2^{78}	CP/ACC	2^{78}	2^{48}
Our Boomerang attack	128	5.5 of 7	2^{108}	CP/ACC	2^{108}	2^{48}
Linear attack[d] [?]	128	3 of 7	2^{81}	KP	2^{101}	2^{81}

[a] Expressed in number of blocks.
[b] KP – Known Plaintext, CP – Chosen Plaintext, ACC – Adaptive Chosen Ciphertext.
[c] Expressed in equivalent number of encryptions.
[d] Works for one in 2^{13} keys.

SAFER++ is an iterated product cipher in which every round consists of an upper key layer, a nonlinear layer, a lower key layer and a linear transformation. Fig. 1 shows the structure of one SAFER++ round. After the final round there is an output transformation that is similar to the upper key layer. The upper and lower key layers together with the nonlinear layer make up the *keyed nonlinear layer*, denoted by S. The linear layer is denoted by A.

2.1 The Keyed Nonlinear Layer

The upper key layer combines a 16 byte subkey with the 16 byte block. Bytes 0, 3, 4, 7, 8, 11, 12 and 15 of the subkey are XORed to the corresponding bytes of the block and bytes 1, 2, 5, 6, 9, 10, 13 and 14 are combined using addition modulo 256.

The nonlinear layer is based on two 8-to-8-bit functions, X and L defined as

$$\begin{aligned} \mathrm{X}(a) &= (45^a \bmod 257) \bmod 256\,, \\ \mathrm{L}(a) &= \log_{45}(a) \bmod 257\,, \end{aligned}$$

with the special case that $\mathrm{L}(0) = 128$, making X and L mutually inverse. In the nonlinear layer, bytes 0, 3, 4, 7, 8, 11, 12 and 15 are sent through the function X, and L is applied to bytes 1, 2, 5, 6, 9, 10, 13 and 14.

The lower key layer applies a 16 byte subkey to the 16 byte block using addition modulo 256 for bytes 0, 3, 4, 7, 8, 11, 12 and 15 and XOR for bytes 1, 2, 5, 6, 9, 10, 13 and 14.

2.2 The Linear Layer

The linear transformation of SAFER++ is built from a 4-point Pseudo Hadamard Transform (4-PHT) and a coordinate permutation. The 4-PHT can be implemented with six modular additions.

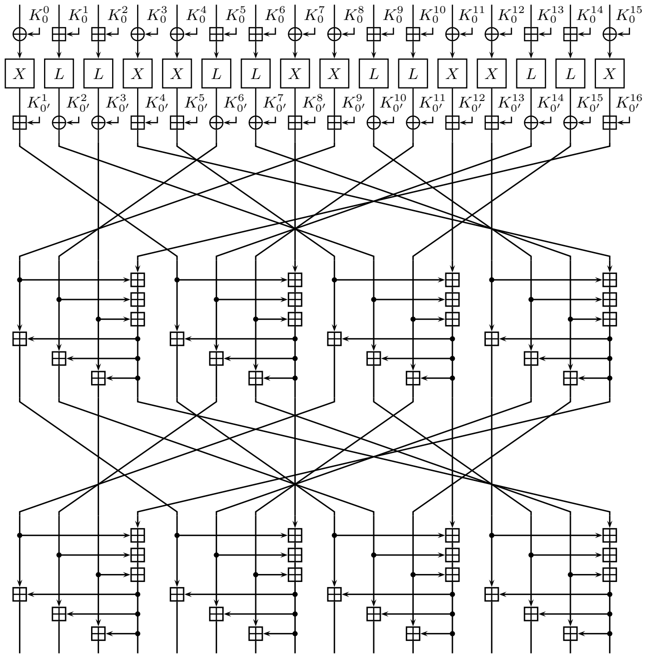

Fig. 1. One round of SAFER++.

The linear layer first reorders the input bytes and then applies the 4-PHT to groups of four bytes. The output of the linear layer is obtained after iterating this operation twice.

The linear layer and its inverse can be represented by the matrices A and A^{-1}. Since the linear layer consists of two iterations of one linear function the matrix A can be written as the square of a matrix $\sqrt{A}$. The matrices A and A^{-1} are shown in Appendix **??**.

2.3 The Key schedule

The key schedule expands the 128 or 256-bit master key into the required number of subkeys. It consists of two similar parts differing only in the way the master key is used to fill the registers. The first part generates the subkeys for the upper key layer and the output transform and the second part generates subkeys for the lower key layer.

It can be noted that the key schedule provides no interaction between bytes of the key and furthermore, there is a big overlap between the key bytes used in different rounds. Therefore, we will not number the bytes of the subkeys according to the order in the subkeys, but according to which master key byte they depend on.

3 Properties of the Components

In this section we show some interesting properties of the components of SAFER++ which will be used later in our analysis.

3.1 Diffusion in the Linear Layer

In [?], the designers show that the choice of the components used in the linear layer provides "optimum transform diffusion" without sacrificing efficiency. In order to measure this diffusion, the authors compute the minimal number of output bytes that are affected by a change in a single input byte. In the case of SAFER++, for example, the linear layer guarantees that a single byte difference at the input of the layer will cause at least ten output bytes to be different.

While the "optimum transform diffusion" defined in this way is certainly a desirable property, it potentially allows some low-weight differentials that might still be useful for an attacker. For example, if two input bytes are changed simultaneously in SAFER++, the number of affected output bytes after the linear layer can be reduced to only three. The adversary might also consider to attack the layer in decryption direction, in which case single byte differences are only guaranteed to propagate to at least five bytes. Neither of these cases is captured by the diffusion criterion used in [?].

3.2 Symmetry of the Linear Layer

Due to the symmetry of the 4-PHT and the coordinate permutation used, there is a four byte symmetry in the linear layer. If the input difference to the linear layer is of the form

$$(a, b, c, d, a, b, c, d, a, b, c, d, a, b, c, d)$$

for any 8-bit values a, b, c, and d the output difference will be of the form

$$(x, y, z, t, x, y, z, t, x, y, z, t, x, y, z, t)$$

The nonlinear layer is symmetric in the same way and were it not for the subkeys, the property would hold for the whole SAFER++ cipher, with an arbitrary number of rounds.

A special illustration of this property are the two eigenvectors of the linear transformation corresponding to the eigenvalue 1:

$$\begin{gathered}(0, 1, -1, 0, 0, 1, -1, 0, 0, 1, -1, 0, 0, 1, -1, 0) \\ (1, 0, -1, 0, 1, 0, -1, 0, 1, 0, -1, 0, 1, 0, -1, 0)\end{gathered}$$

These vectors and all linear combinations of them are fixed points of the linear transform.