ՀԱՅԱՍՏԱՆԻ ԳԻՏՈՒԹՅՈՒՆՆԵՐԻ ԱԶԳԱՅԻՆ ԱԿԱԴԵՄԻԱ
НАЦИОНАЛЬНАЯ АКАДЕМИЯ НАУК АРМЕНИИ
NATIONAL ACADEMY OF SCIENCES OF ARMENIA
ДОКЛАДЫ ԶԵԿՈՒՅՑՆԵՐ REPORTS

Հատոր
Том 115 2015 № 1
Volume

CRYPTOGRAPHY

УДК 004.9

K. M. Kyuregyan

Some Modifications of SAFER+

(Submitted by academician G. H. Khachatrian 17/X 2014)

Keywords: *block cipher, encryption, decryption, round, Armenian shuffle, differential cryptanalysis.*

1. Introduction. SAFER+ is one of the block ciphers of SAFER family proposed by Prof. James L. Massey together with Prof. Gurgen H. Khachatrian and Dr. Melsik K. Kyuregian. It is a 128 block size encryption algorithm with three different user-selected-key lengths, namely 128, 192 and 256. SAFER+ was submitted as a candidate for the Advanced Encryption Standard (AES) [3] and was subsequently adopted for use in the challenge/response entity authentication scheme in the Bluetooth protocol for wireless communications [5]. In this paper some modifications of SAFER+ algorithm are proposed resulting about 1.7 times faster algorithm implementation on ARM platform.

2. Brief description of SAFER+ algorithm. SAFER+ is a 128-bit block cipher. In Fig. 1 the encryption structure of the SAFER+ algorithm is introduced.

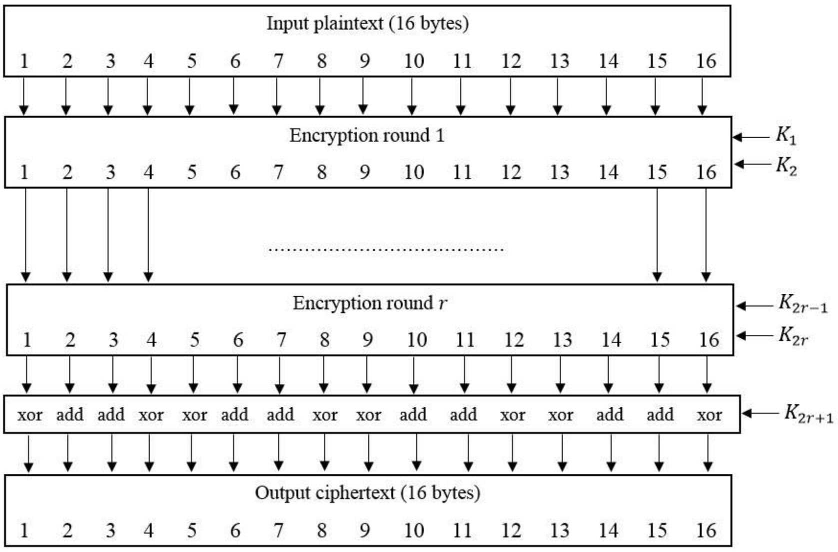

Fig. 1

The 16 byte plaintext block passes through $r=6$ rounds of encryption for 128 bit key and $r=9$ rounds of encryption for 256 bit key. In each round of encryption two subkeys are used. These round subkeys $(K_1, K_2, \dots, K_{2r+1})$ are determined from the user-selected key K according to the key schedule of SAFER+. The details of the key schedule structure are introduced in [3]. The last subkey K_{2r+1} is "added" to the block produced by the r rounds of encryption in the manner that the bytes 1, 4, 5, 8, 9, 12, 13 and 16 are added together bit-by-bit modulo two (the bitwise "exclusive-or" operation) while the bytes 2, 3, 6, 7, 10, 11, 14 and 15 are added together modulo 256 ("byte addition"). This "addition" of round subkey K_{2r+1} constitutes the *output transformation* for encryption and produces the ciphertext block of 16 bytes.

The input for decryption is the ciphertext block of 16 bytes. The decryption begins with the *input transformation* that undoes the *output transformation* in the encryption process. At first the round subkey K_{2r+1} is "subtracted" from the ciphertext block in the manner that the round subkey bytes 1, 4, 5, 8, 9, 12, 13 and 16 are added together bit-by-bit modulo two to the corresponding ciphertext bytes while the round subkey bytes 2, 3, 6, 7, 10, 11, 14 and 15 are subtracted modulo 256 from the corresponding ciphertext bytes. The result of this "subtraction" is the same 16-byte block as was produced from the r rounds of encryption before the output transformation was applied. This block then passes through the r rounds of decryption, the round i of which undoes the round $r-i+1$ of encryption, where $i=1,2,\dots,r$. After the round r we obtain a plaintext block. Note that the round keys for decryption are the same as those for encryption but are used in reverse order.

2.1 SAFER+ encryption round. The SAFER+ round schema is given in Fig. 2. The first operation within the round $i, 1 \le i \le r$, is the "addition" of the round subkey K_{2i-1} to the 16-byte round input in the manner that the bytes 1, 4, 5, 8, 9, 12, 13 and 16 are added together bit-by-bit modulo two while the bytes 2, 3, 6, 7, 10, 11, 14 and 15 are added together modulo 256. The 16-byte result of this "addition" is then processed by *a nonlinear layer* in the manner that the value x of byte j is converted to $45^x \bmod 257$ for bytes $j=1,4,5,8,9,12,13,16$ (with the convention that when $x=128$, then $45^{128} \bmod 257 = 256$ is represented by 0), while the value x of byte j is converted to $\log_{45} x$ for bytes $j=2,3,6,7,10,11,14,15$ (with the convention that when $x=0$, then the output $\log_{45} 0$ is represented by 128). The round key K_{2i} is then "added" to the output of the *nonlinear layer* in the manner that the bytes 2, 3, 6, 7, 10, 11, 14 and 15 are added together bit-by-bit modulo two, while the bytes 1, 4, 5, 8, 9, 12, 13 and 16 are added together modulo 256. The 16-byte result of this "addition"

$$x=\left[x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9,x_{10},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16}\right]$$

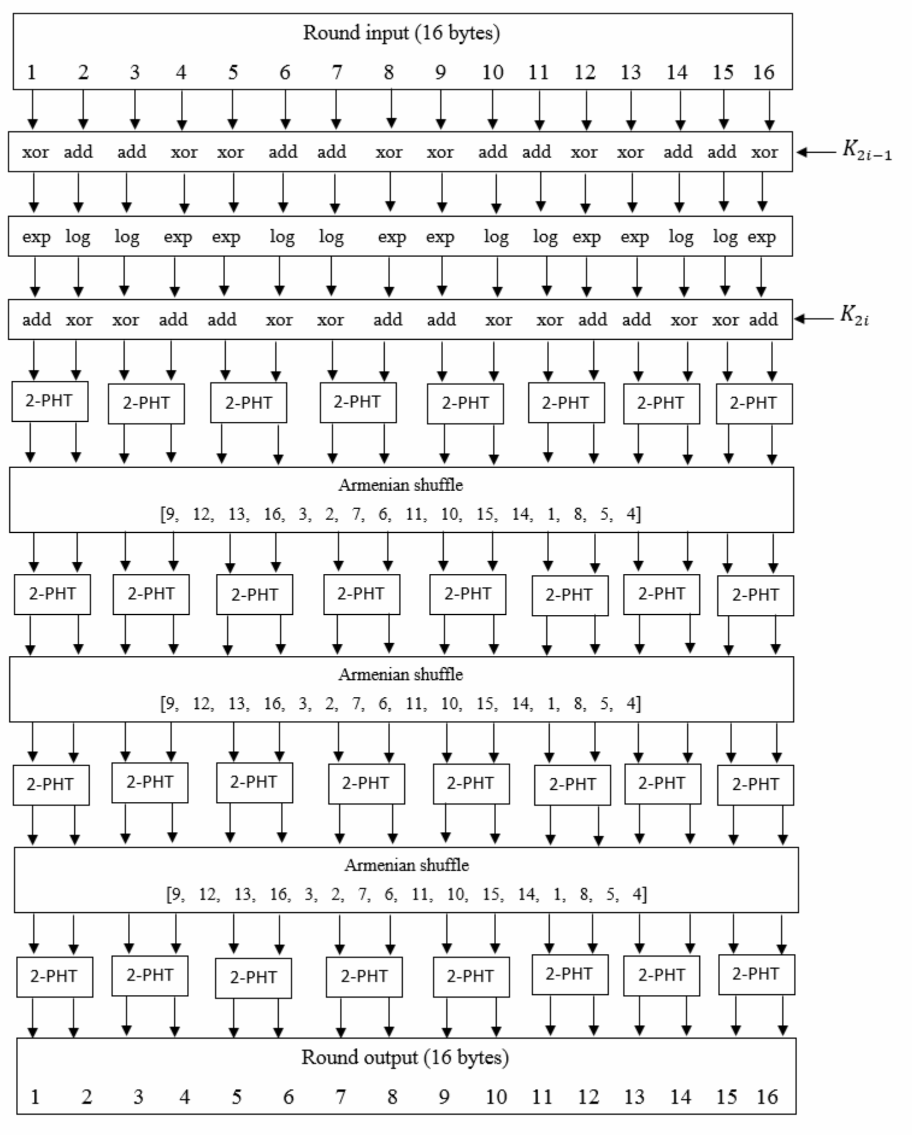

Fig. 2

is then postmultiplied by the matrix M modulo 256 to give the 16-byte round output

$$y = [y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}, y_{11}, y_{12}, y_{13}, y_{14}, y_{15}, y_{16}]$$

in the manner

$$y = Mx,$$

where M is the following 16×16 matrix

$$M=\begin{vmatrix}
2 & 2 & 1 & 1 & 16 & 8 & 2 & 1 & 4 & 2 & 4 & 2 & 1 & 1 & 4 & 4 \\
1 & 1 & 1 & 1 & 8 & 4 & 2 & 1 & 2 & 1 & 4 & 2 & 1 & 1 & 2 & 2 \\
1 & 1 & 4 & 4 & 2 & 1 & 4 & 2 & 4 & 2 & 16 & 8 & 2 & 2 & 1 & 1 \\
1 & 1 & 2 & 2 & 2 & 1 & 2 & 1 & 4 & 2 & 8 & 4 & 1 & 1 & 1 & 1 \\
4 & 4 & 2 & 1 & 4 & 2 & 4 & 2 & 16 & 8 & 1 & 1 & 1 & 1 & 2 & 2 \\
2 & 2 & 2 & 1 & 2 & 1 & 4 & 2 & 8 & 4 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 4 & 2 & 4 & 2 & 16 & 8 & 2 & 1 & 2 & 2 & 4 & 4 & 1 & 1 \\
1 & 1 & 2 & 1 & 4 & 2 & 8 & 4 & 2 & 1 & 1 & 1 & 2 & 2 & 1 & 1 \\
2 & 1 & 16 & 8 & 1 & 1 & 2 & 2 & 1 & 1 & 4 & 4 & 4 & 2 & 4 & 2 \\
2 & 1 & 8 & 4 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 4 & 2 & 2 & 1 \\
4 & 2 & 4 & 2 & 4 & 4 & 1 & 1 & 2 & 2 & 1 & 1 & 16 & 8 & 2 & 1 \\
2 & 1 & 4 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 8 & 4 & 2 & 1 \\
4 & 2 & 2 & 2 & 1 & 1 & 4 & 4 & 1 & 1 & 4 & 2 & 2 & 1 & 16 & 8 \\
4 & 2 & 1 & 1 & 1 & 1 & 2 & 2 & 1 & 1 & 2 & 1 & 2 & 1 & 8 & 4 \\
16 & 8 & 1 & 1 & 2 & 2 & 1 & 1 & 4 & 4 & 2 & 1 & 4 & 2 & 4 & 2 \\
8 & 4 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 1 & 2 & 1 & 4 & 2
\end{vmatrix}$$

This operation gives

$$y_3 = x_1 + x_2 + 4x_3 + 2x_4 + 2x_5 + 2x_6 + 4x_7 + 2x_8 +$$
$$+16x_9 + 8x_{10} + 4x_{11} + 4x_{12} + 2x_{13} + x_{14} + x_{15} + x_{16}$$

(where the arithmetic is modulo 256) as follows from the second column of the matrix M . Multiplication by matrix M provides *the linear layer* of the round that consists of the cascade of *2-PHT* and 3 times "*Armenian shuffle*"+*2-PHT* operations. "*Armenian shuffle*" is the coordinate permutation [9, 12, 13, 16, 3, 2, 7, 6, 11, 10, 15, 14, 1, 8, 5, 4] and *2-PHT* is *Pseudo-Hadamrd matrix* $\begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$, that has as an input 2 bytes (a_1, a_2) and as an output $(2a_1 + a_2, a_1 + a_2)$ 2-bytes over the ring of integers modulo 256 (all operations are modulo 256).

2.2. SAFER+ decryption round. In the decryption round of SAFER+ simply inverts in reverse order the operations from the encryption round. Thus, the first operation in the decryption round is to postmultiply the 16-byte round input

$$y = [y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}, y_{11}, y_{12}, y_{13}, y_{14}, y_{15}, y_{16}]$$

by the matrix M^{-1}, which is modulo 256 inverse of M , to give the 16-byte result

$$x = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}]$$

in the manner

$$x = yM^{-1},$$

where matrix M^{-1} is the 16×16 matrix ($-i$ denote $256-i$ in modulo 256 arithmetic)

$$M^{-1} = \begin{vmatrix}
2 & -2 & 1 & -2 & 1 & -1 & 4 & -8 & 2 & -4 & 1 & -1 & 1 & -2 & 1 & -1 \\
-4 & 4 & -2 & 4 & -2 & 2 & -8 & 16 & -2 & 4 & -1 & 1 & -1 & 2 & -1 & 1 \\
1 & -2 & 1 & -1 & 2 & -4 & 1 & -1 & 1 & -1 & 1 & -2 & 2 & -2 & 4 & -8 \\
-2 & 4 & -2 & 2 & -2 & 4 & -1 & 1 & -1 & 1 & -1 & 2 & -4 & 4 & -8 & 16 \\
1 & -1 & 2 & -4 & 1 & -1 & 1 & -2 & 1 & -2 & 1 & -1 & 4 & -8 & 2 & -2 \\
-1 & 1 & -2 & 4 & -1 & 1 & -1 & 2 & -2 & 4 & -2 & 2 & -8 & 16 & -4 & 4 \\
2 & -4 & 1 & -1 & 1 & -2 & 1 & -1 & 2 & -2 & 4 & -8 & 1 & -1 & 1 & -2 \\
-2 & 4 & -1 & 1 & -1 & 2 & -1 & 1 & -4 & 4 & -8 & 16 & -2 & 2 & -2 & 4 \\
1 & -1 & 1 & -2 & 1 & -1 & 2 & -4 & 4 & -8 & 2 & -2 & 1 & -2 & 1 & -1 \\
-1 & 1 & -1 & 2 & -1 & 1 & -2 & 4 & -8 & 16 & -4 & 4 & -2 & 4 & -2 & 2 \\
1 & -2 & 1 & -1 & 4 & -8 & 2 & -2 & 1 & -1 & 1 & -2 & 1 & -1 & 2 & -4 \\
-1 & 2 & -1 & 1 & -8 & 16 & -4 & 4 & -2 & 2 & -2 & 4 & -1 & 1 & -2 & 4 \\
4 & -8 & 2 & -2 & 1 & -2 & 1 & -1 & 1 & -2 & 1 & -1 & 2 & -4 & 1 & -1 \\
-8 & 16 & -4 & 4 & -2 & 4 & -2 & 2 & -1 & 2 & -1 & 1 & -2 & 4 & -1 & 1 \\
1 & -1 & 4 & -8 & 2 & -2 & 1 & -2 & 1 & -1 & 2 & -4 & 1 & -1 & 1 & -2 \\
-2 & 2 & -8 & 16 & -4 & 4 & -2 & 4 & -1 & 1 & -2 & 4 & -1 & 1 & -1 & 2
\end{vmatrix}$$

For instance, these operations give

$$x_{10} = -4y_1 + 4y_2 - y_3 + y_4 - 2y_5 + 4y_6 - 2y_7 + 4y_8 -$$
$$-8y_9 + 16y_{10} - y_{11} + 2y_{12} - 2y_{13} + 2y_{14} - y_{15} + y_{16}$$

The round subkey $K_{2r-2i+2}$ is then "subtracted" from x in the manner that the round subkey bytes 1, 4, 5, 8, 9, 12, 13 and 16 are subtracted modulo 256 from the corresponding bytes of x while the round subkey bytes 2, 3, 6, 7, 10, 11, 14 and 15 are added bit-by-bit modulo 2 to the corresponding bytes of x. Then the 16-byte result of this "subtraction" is then processed nonlinearly in the manner that the value x of byte j is converted to $\log_{45} x$ for bytes $j = 1,4,5,8,9,12,13,16$ (again with the convention that when $x = 0$, then the output $\log_{45} x$ is represented by 128), while the value x of byte j is converted to $45^x \bmod 257$ for bytes $j = 2,3,6,7,10,11,14,15$ (again with the convention that when $x = 128$, then $45^{128} \bmod 257 = 256$ is represented by 0). The round key $K_{2r-2i+1}$ is then "subtracted" from the 16-byte result in the manner that the round subkey bytes 1, 4, 5, 8, 9, 12, 13 and 16 are added bit-by-bit modulo 2 to the corresponding input bytes while the round subkey bytes 2, 3, 6, 7, 10, 11, 14 and 15 are subtracted modulo 256 from the corresponding input bytes to obtain the 16-byte round output.

3. Some modifications of SAFER+. We propose three major modifications for SAFER+. They concern both nonlinear and liner parts of algorithm:

1. As it can be seen from Fig. 2, for SAFER+ algorithm exp boxes were used for 1, 4, 5, 8, 9, 12, 13, 16 bytes while log boxes were used for 2, 3, 6, 7, 10, 11, 14, 15 bytes. Here we propose to use exp boxes for 1, 2, 3, 4, 9, 10, 11, 12 bytes and log boxes for 5, 6, 7, 8, 13, 14, 15, 16 bytes.
2. It is clear from Fig. 2, that the liner part of algorithm starts so-called 2-PHT operation. We propose to start the liner part from shuffling of bytes immediately. As such in this case we will have 4 times of byte shuffling instead of 3.
3. We propose to use [7, 12, 9, 14, 5, 8, 13, 10, 11, 4, 3, 6, 15, 2, 1, 16] “Armenian shuffle” instead of [9, 12, 13, 16, 3, 2, 7, 6, 11, 10, 15, 14, 1, 8, 5, 4]. As the result of the last two modifications we will have completely different liner transformation matrix.

The properties of these modifications are analyzed in detail in the next section.

4. The result of the modifications. Firstly a differential cryptanalysis of a modified version of SAFER+ has been implemented. The attack by differential cryptanalysis on an r-round cipher relies on being able to find a $r-1$ round differential whose probability is substantially greater than the average probability of such a differential, which is $\frac{1}{2^{128}-1} \approx 2^{-128}$ for a 16-byte block length, i.e. for any selected key after $r-1$ round differential the probability is smaller then 2^{-128}. Providing the count of rounds of SAFER+ we have analyzed all the possible “highly probable” 5-round and 8-round differential chains (see [2]) and have found that due to second and third modifications their probabilities are substantially less than 2^{-128} and 2^{-256} correspondingly then before modifications, but the count of rounds before and after modifications stays the same i.e. after $r=6$ and $r=9$ rounds by differential cryptanalysis it is impossible to find out master key used in the algorithm. Secondly, when implementing a SAFER+ algorithm on ARM platform due to the modified version it is possible to *xor* four bytes simultaneously, which was impossible with a regular SAFER+. The first modification also shortens the count of operations on ARM platform in the liner part of the algorithm structure (multiplication by matrix). As such the main result of these modifications is that SAFER+ will run ≈ 1.7 times faster on ARM platform.

5. Conclusion. In this paper three major modifications of SAFER+ algorithm are implemented resulting an increase of the speed of algorithm implementation on ARM platform about 1.7 times. In addition it is shown that these modifications will not affect the security of SAFER+.

6. Acknowledgment. The author is thankful to Prof. Gurgen H. Khachatrian and Dr. Melsik K. Kyuregian for very useful discussions and comments.

This work was supported by State Committee Science MES RA, in frame of the research project SCS-13-1B352.

Institute for Informatics and Automaton Problems of NAS RA
e-mail: *knarikyuregyan@gmail.com*

K. M. Kyuregyan

Some Modifications of SAFER+

Some modifications of SAFER+ encryption algorithm of SAFER family are presented. After these modifications SAFER+ stays secure against differential cryptanalysis, but these modifications make SAFER+ algorithm implementation ≈1.7 times faster on ARM platform.

Ք. Մ. Կյուրեղյան

SAFER+ համակարգի որոշ ձևափոխություններ

Ներկայացված է SAFER ընտանիքին պատկանող SAFER+ ծածկագրական համակարգի որոշ ձևափոխություններ: Կատարվել է դիֆերենցիալ վերլուծություն, ինչը ցույց է տվել, որ SAFER+ ծածկագրական համակարգը այդ ձևափոխություններից հետո ևս կայուն է դիֆերենցիալ վերլուծության նկատմամբ և շնորհիվ այդ ձևափոխությունների ARM պլատֆորմի վրա ≈1.7 անգամ ավելի արագ է:

К. М. Кюрегян

Некоторые модификации SAFER+

Представлены некоторые модификации криптографической системы SAFER+ из семьи SAFER. Проведенный дифференциальный анализ, показал, что после этих модификаций система SAFER+ также устойчива по отношению к дифференциальному анализу и благодаря им SAFER+ на платформе ARM в ≈1.7 раза быстрее

References

1. *Biham E., Shamir A.* In: Advances in Cryptology-CRYPTO’90 (Eds. A. J. Menezes and S. A. Vanstone), Lecture Notes in Computer Science No. 537, Heidelberg and New York: Springer 1990, p. 212-241.
2. *Massey J. L.* In: Fast Software Encryption II (Ed. B. Prenell), Lecture Notes in Computer Science No. 1008, New York, Springer 1995, p. 212-241.
3. *Massey J. L., Khachatrian G. H., Kyuregian M. K.* “Nomination of SAFER + as Candidate Algorithm for the Advanced Encryption Standard (AES)”, NIST AES Proposal, 1998.
4. *Massey J. L., Khachatrian G. H., Kyuregian M. K.* “Nomination of SAFER++ as Candidate Algorithm for the New European Schemes for Signatures”, Integrity and Encryption (NESSIE), Submission document from Cylink Corporation, 2000.
5. BLUETOOTH SPECIFICATION Version 1.0B, 29 Nov. 1999, http://www.bluetooth.com/link/pec/bluetooth_b.pdf.