# Distinguishing Attack on a Modified Version of MAG Stream Cipher

Arash Mirzaei[1], Mohammad Dakhil Alian[2], Mahmoud Modarres Hashemi[3]

Cryptography & System Security Research Lab. (CSSRL)
Isfahan University of Technology, Isfahan, Iran
[1]arash_mirzaei@ec.iut.ac.ir
[2, 3]{mdalian, modarres}@cc.iut.ac.ir

## Abstract

**MAG is a synchronous stream cipher designed by Vuckovac submitted to the eSTREAM project. Vuckovac also proposed two modified versions of MAG to avoid the distinguishing attack on the first version of MAG presented by Fischer. In this paper we show that, changing the Fischer's attack we can apply it to one of the modified versions of MAG. The modified attack requires only 514 successive bytes of known keystream and 5 xor and 2 comparison operations between 16 bit words. In addition, we show that distinguishing and key recovery attack proposed by Simpson and Henricksen on all versions of MAG is feasible just by considering an assumption on initialization of MAG that simplifies this step so much. Therefore, their attack cannot be performed in general.**

## Keywords

**Cryptanalysis, MAG Stream Cipher, Distinguishing Attack, eSTREAM**

## 1. Introduction

MAG is a stream cipher which its internal state consists of 127 registers $R_i$ of 32 bit size, as well as a 32 bit carry register C. To produce the keystream, MAG is clocked iteratively. In each iteration, carry C and one of the registers, $R_i$ , are modified. Then $R_i$ is used to produce one byte of the keystream.

In [2], Fischer presented a distinguishing attack on MAG, requiring 129 successive bytes of the keystream. Then Vuckovac, the designer of MAG, proposed two modified versions of MAG to resist against Fischer's attack [3,4]. After that Simpson and Henricksen presented a distinguishing and key recovery attack on all MAG versions. This attack just can be performed when an assumption is considered in the initialization of MAG [5]. The assumption simplifies this step and contradicts with the proposed initialization of Vuckovac.

In this paper, using the Fischer's attack, a distinguishing attack on one of the modified versions of MAG is presented. This attack requires 514 successive bytes of keystream and the computational complexity of that is 5 xor and 2 comparison operations between 16 bit words.

The rest of the paper is organized as follows. In Section 2 the MAG stream cipher is described. Then in Section 3 the Fischer's attack is explained. Two modified versions of MAG are described in Section 4. The Simpson and Henricksen's assumption for initialization of MAG is discussed in Section 5 as well as their attack. In Section 6, a distinguishing attack on one of the modified versions of MAG is presented. Finally, Section 7 presents a summary of the paper.

## 2. Description of the MAG Stream Cipher

The internal state of MAG contains 127 main registers $R_i, 0 \le i < 127$ of 32 bit size and a 32 bit carry C. The secret key and IV are used to initialize $R_0, \dots, R_{126}$ and C. In each iteration of the cipher, carry and one of the main registers are updated and also used to produce one of the keystream bytes. The updating process in [1] is a little different from the corresponding one implemented in C code by Vuckovac. The Fischer's attack and then the Simpson and Henricksen's one are matched with implementation plan as well as the modified versions of MAG in [3,4]. Therefore, the implementation design is presented and discussed here. Of course all of the attacks can be changed slightly to be performed on MAG in [1].

The way of updating in the initialization step is a little different from the keystream generation step.

The way of updating $R_i$ and C in the keystream generation step is as follows:

$$C' = \begin{cases} C \oplus R_{i+1} & if\ R_{i+2} > R_{i+3} \\ C \oplus \overline{R}_{i+1} & otherwise \end{cases} \tag{1}$$

$$R_i' = R_i \oplus C' \tag{2}$$

where $R_i'$ and $C'$ are the updated values of $R_i$ and C, respectively. Then, the first byte of $R_i'$ is used as one byte of the keystream. Above process is iterated after changing i to i+1 (this addition of indices is performed modulo 127) until enough amount of the keystream is produced. In 127 iterations of system, each register $R_i$ is updated one time but the carry is updated 127 times.

In the initialization step after updating $R_i$ and C to $R_i'$ and $C'$, carry changes to:

$$C' = C' + E \bmod 2^{32} \tag{3}$$

where E in the C code is 0x11111111. Therefore this process is different from the corresponding one in the keystream generation step. This difference is only a modular addition. If E is equal to 0, then these two processes will be the same. In the initialization step, the above process is iterated $2^{14}$ times. The details of the key and IV loading are not mentioned here.

## 3. Fischer's attack.

In this section the Fischer's attack on MAG is explained. Suppose in an arbitrarily time t, the state of the registers is $R_0,...,R_{126}$ and C. The goal is updating $R_0$ and producing $k_i$ as a byte of the keystream. So first the carry C is updated as follows:

$$C' = \begin{cases} C \oplus R_1 & if\ R_2 > R_3 \\ C \oplus \overline{R}_1 & otherwise \end{cases} \tag{4}$$

$C'$ shortly is

$$C' = C \oplus R_1 \oplus \mathbf{a}_1 \tag{5}$$

where $\mathbf{a}_1$ is an all zero or all one vector with the same probability (its value depends on the values of $R_2$ and $R_3$). Then $R_0$ is updated as follows:

$$R_0' = R_0 \oplus C' \tag{6}$$

Now similarly $C'$ and $R_1$ are updated:

$$C'' = C' \oplus R_2 \oplus \mathbf{a}_2 \tag{7}$$

and

$$R_1' = R_1 \oplus C'' \tag{8}$$

where $\mathbf{a}_2$ is an all zero or all one vector with the same probability (its value depends on values of $R_3$ and $R_4$). Combining (6), (7) and (8), we have:

$$R_1' = R_0 \oplus R_1 \oplus R_2 \oplus R_0' \oplus \mathbf{a}_2 \tag{9}$$

Each byte of the keystream is the first byte of a register $R_i$, so there is a similar relation between bytes of the keystream like below (the corresponding values for $R_0$, $R_1$, $R_2$, $R_0'$ and $R_1'$ are $k_i$, $k_{i+1}$, $k_{i+2}$, $k_{i+127}$ and $k_{i+128}$, respectively.):

$$k_{i+128} = \begin{cases} k_i \oplus k_{i+1} \oplus k_{i+2} \oplus k_{i+127} \\ k_i \oplus k_{i+1} \oplus k_{i+2} \oplus k_{i+127} \oplus \mathbf{1} \end{cases} \tag{10}$$

where **1** shows an all one vector. For the keystream of the MAG, the probability that one of the above relations be hold is 1, while for a random sequence this probability is $2^{-7}$. Therefore having 129 successive bytes of the output sequence, $k_i$, $k_{i+1}$, $k_{i+2}$, $k_{i+127}$ and $k_{i+128}$, we should obtain the values $k_i \oplus k_{i+1} \oplus k_{i+2} \oplus k_{i+127}$ and $k_i \oplus k_{i+1} \oplus k_{i+2} \oplus k_{i+127}$ $\oplus \mathbf{1}$ and compare them with the value of $k_{i+128}$. If $k_{i+128}$ is equal to one of the mentioned values it is concluded that the available sequence is output of MAG with false alarm probability of $2^{-7}$. Otherwise, this sequence is a random one with miss probability of 0.

## 4. Modification of the MAG

After Fischer's attack, MAG designer proposed two modified versions for MAG to resist against this attack [3,4]. First proposal is that output bytes are harvested from different positions within 32 bit words (instead of using only first bytes of registers as keystream). Fig. 1 shows these positions.

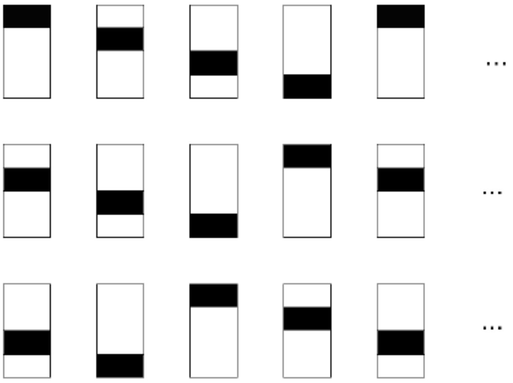

**Fig. 1. First Proposal of Vucovac**

Second proposal is that from each register two bytes are used as keystream. The way of choosing output bytes is shown in Figure 2. In section 6 a distinguishing attack on this plan is presented.

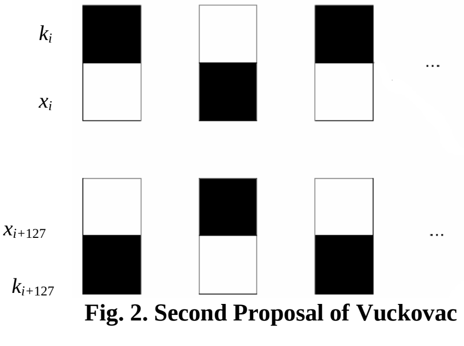

**Fig. 2. Second Proposal of Vuckovac**

## 5. Simpson and Henricksen's Attack

The other attack on MAG was presented by Simpson and Henricksen. In this attack it is assumed that the value of E in the initialization step is equal to 0, like the key generation step. So the initialization process becomes linear (the only nonlinear operation in GF(2), i.e. addition modulo $2^{32}$, is eliminated). Therefore, number of cases of output bytes will be limited. For example when the key size is 128 bit and IV is not used, registers $R_i, 0 \le i < 127$ and C are completed as follows:

$(R_0, R_1, ..., R_{126}, C) =$
$(K_0, ..., K_3, K_0, ..., K_3, ..., K_0, ..., K_3)$

where each $K_i$ shows ith 32 bit block of the secret key. Then $2^{14}$ clocks are applied to the system. Tables 1 and 2 show the updating process of carry and main registers through the first 8 clocks in the initialization step with E=0 and the key size of 128 and no IV, respectively.

**Table 1. Updating Process of Carry Register Through the First 8 Clocks in the Initialization Step with E=0 and the Key Size of 128 and no IV**

| $i$ | $R_{i+2}$ | $R_{i+3}$ | $C$ |
|---|---|---|---|
| 0 | $K_2$ | $K_3$ | $K_3 \oplus K_1 \oplus \mathbf{a}_1$ |
| 1 | $K_3$ | $K_0$ | $K_3 \oplus K_2 \oplus K_1 \oplus \mathbf{a}_1 \oplus \mathbf{a}_2$ |
| 2 | $K_0$ | $K_1$ | $K_2 \oplus K_1 \oplus \mathbf{a}_1 \oplus \mathbf{a}_2 \oplus \mathbf{a}_3$ |
| 3 | $K_1$ | $K_2$ | $K_2 \oplus K_1 \oplus K_0 \oplus \mathbf{a}_1 \oplus \mathbf{a}_2 \oplus \mathbf{a}_3 \oplus \mathbf{a}_4$ |
| 4 | $K_2$ | $K_3$ | $K_2 \oplus K_0 \oplus \mathbf{a}_1 \oplus \mathbf{a}_2 \oplus \mathbf{a}_3 \oplus \mathbf{a}_4 \oplus \mathbf{a}_5$ |
| 5 | $K_3$ | $K_0$ | $K_0 \oplus \mathbf{a}_1 \oplus \mathbf{a}_2 \oplus \mathbf{a}_3 \oplus \mathbf{a}_4 \oplus \mathbf{a}_5 \oplus \mathbf{a}_6$ |
| 6 | $K_0$ | $K_1$ | $K_3 \oplus K_0 \oplus \mathbf{a}_1 \oplus \mathbf{a}_2 \oplus \mathbf{a}_3 \oplus \mathbf{a}_4 \oplus \mathbf{a}_5 \oplus \mathbf{a}_6 \oplus \mathbf{a}_7$ |
| 7 | $K_1$ | $K_2$ | $K_3 \oplus \mathbf{a}_1 \oplus \mathbf{a}_2 \oplus \mathbf{a}_3 \oplus \mathbf{a}_4 \oplus \mathbf{a}_5 \oplus \mathbf{a}_6 \oplus \mathbf{a}_7 \oplus \mathbf{a}_8$ |

where $\mathbf{a}_1$, $\mathbf{a}_2$, … , $\mathbf{a}_8$ are all zero or all one vectors. It is clear that each register can only have 32 different values because combinations of the vectors $\mathbf{a}_1$, $\mathbf{a}_2$, … , $\mathbf{a}_8$ can have 2 values and 16 values are possible for combinations of $K_0, ..., K_3$ (for a fixed secret key).

**Table 2. Updating Process of Main Registers Through the First 8 Clocks in the Initialization Step with E=0 and the Key Size of 128 and no IV**

| $i$ | $R_{i+2}$ | $R_{i+3}$ | $R'_i$ |
|---|---|---|---|
| 0 | $K_2$ | $K_3$ | $K_3 \oplus K_1 \oplus K_0 \oplus \mathbf{a}_1$ |
| 1 | $K_3$ | $K_0$ | $K_3 \oplus K_2 \oplus \mathbf{a}_1 \oplus \mathbf{a}_2$ |
| 2 | $K_0$ | $K_1$ | $K_1 \oplus \mathbf{a}_1 \oplus \mathbf{a}_2 \oplus \mathbf{a}_3$ |
| 3 | $K_1$ | $K_2$ | $K_3 \oplus K_2 \oplus K_1 \oplus K_0 \oplus \mathbf{a}_1 \oplus \mathbf{a}_2 \oplus \mathbf{a}_3 \oplus \mathbf{a}_4$ |
| 4 | $K_2$ | $K_3$ | $K_2 \oplus \mathbf{a}_1 \oplus \mathbf{a}_2 \oplus \mathbf{a}_3 \oplus \mathbf{a}_4 \oplus \mathbf{a}_5$ |
| 5 | $K_3$ | $K_0$ | $K_0 \oplus K_1 \oplus \mathbf{a}_1 \oplus \mathbf{a}_2 \oplus \mathbf{a}_3 \oplus \mathbf{a}_4 \oplus \mathbf{a}_5 \oplus \mathbf{a}_6$ |
| 6 | $K_0$ | $K_1$ | $K_3 \oplus K_2 \oplus K_0 \oplus \mathbf{a}_1 \oplus \mathbf{a}_2 \oplus \mathbf{a}_3 \oplus \mathbf{a}_4 \oplus \mathbf{a}_5 \oplus \mathbf{a}_6 \oplus \mathbf{a}_7$ |
| 7 | $K_1$ | $K_2$ | $\mathbf{a}_1 \oplus \mathbf{a}_2 \oplus \mathbf{a}_3 \oplus \mathbf{a}_4 \oplus \mathbf{a}_5 \oplus \mathbf{a}_6 \oplus \mathbf{a}_7 \oplus \mathbf{a}_8$ |

This fact also will be hold after initialization. Therefore, the output bytes can have at most 32 different values, but in the random case, each byte can have 256 different values. So if we know about 32 bytes of the output, it is possible to distinguish output of MAG from a random sequence. It is possible to improve the attack to a key recovery one with a precomputation of complexity $2^{14}$ [5].

The main problem of Simpson and Henricksen's attack is that it is assumed that the value of E is equal to 0. This assumption simplifies the initialization of MAG too much, because the value of E has been determined by designer of MAG and is equal to 0x11111111. Any nonzero value for E increases the complexity of initialization so much, because it adds a numeric operation to this step. It means that after $2^{14}$ clocks in the initialization, the registers can have a lot of different values (instead of only 32 values). Tables 3 and 4 show the updating process of carry and main registers through the first 4 clocks in the initialization step with $E \neq 0$ and key size of 128 and no IV, respectively.

**Table 3. Registers Updating Process Through the First 4 Clocks in the Initialization Step with $\mathbf{E \neq 0}$ and Key Size of 128 and no IV**

| $i$ | $R_{i+2}$ | $R_{i+3}$ | $C$ |
|---|---|---|---|
| 0 | $K_2$ | $K_3$ | $K_3 \oplus K_1 \oplus \mathbf{a}_1$ |
| 1 | $K_3$ | $K_0$ | $(K_3 \oplus K_1 \oplus \mathbf{a}_1 + \mathrm{E}) \oplus K_2 \oplus \mathbf{a}_2$ |
| 2 | $K_0$ | $K_1$ | $((K_3 \oplus K_1 \oplus \mathbf{a}_1 + \mathrm{E}) \oplus K_2 \oplus \mathbf{a}_2 + \mathrm{E}) \oplus K_3 \oplus \mathbf{a}_3$ |
| 3 | $K_1$ | $K_2$ | $(((K_3 \oplus K_1 \oplus \mathbf{a}_1 + \mathrm{E}) \oplus K_2 \oplus \mathbf{a}_2 + \mathrm{E}) \oplus K_3 \oplus \mathbf{a}_3 + \mathrm{E}) \oplus K_0 \oplus \mathbf{a}_4$ |

**Table 4. Updating Process of Main Registers Through the First 4 Clocks in the Initialization Step with $\mathbf{E \neq 0}$ and Key Size of 128 and no IV**

| $i$ | $R_{i+2}$ | $R_{i+3}$ | $R'_i$ |
|---|---|---|---|
| 0 | $K_2$ | $K_3$ | $K_3 \oplus K_1 \oplus K_0 \oplus \mathbf{a}_1$ |
| 1 | $K_3$ | $K_0$ | $(K_3 \oplus K_1 \oplus \mathbf{a}_1 + \mathrm{E}) \oplus K_2 \oplus \mathbf{a}_2 \oplus K_1$ |
| 2 | $K_0$ | $K_1$ | $((K_3 \oplus K_1 \oplus \mathbf{a}_1 + \mathrm{E}) \oplus K_2 \oplus \mathbf{a}_2 + \mathrm{E}) \oplus K_3 \oplus \mathbf{a}_3 \oplus K_2$ |
| 3 | $K_1$ | $K_2$ | $(((K_3 \oplus K_1 \oplus \mathbf{a}_1 + \mathrm{E}) \oplus K_2 \oplus \mathbf{a}_2 + \mathrm{E}) \oplus K_3 \oplus \mathbf{a}_3 + \mathrm{E}) \oplus K_0 \oplus \mathbf{a}_4 \oplus K_3$ |

where $\mathbf{a}_1$, $\mathbf{a}_2$, $\mathbf{a}_3$ and $\mathbf{a}_4$ are all zero or all one vectors and E is a fixed nonzero value.

Table 4 obviously represents that after $2^{14}$ clocks, number of possible values for the main registers is too much more than 32 (ignoring a few values of registers which are randomly equal, these registers have 127 different values). Therefore after initialization each updated register can have about $2^{32}$ different values and each output byte can get about 256 different values. So it sounds that this attack cannot be performed for nonzero values of E in spite of Simpson and Henricksen's claim which have predicted that the attack can be extended for these values of E. In the next Section a distinguishing attack on one of the modified versions of MAG is presented which is the first practical one on this stream cipher, by the present time.

## 6. Distinguishing Attack on a Modified Version of MAG

In this section a distinguishing attack on the second modified version of MAG, described in section 4, is presented. The way of choosing output bytes from the registers of this version was shown in Fig. 2. In this figure the keys produced from $R_0$ ,$R_0'$ and $R_0''$ are $k_i$ ,$k_{i+127}$ and $k_{i+254}$ , respectively. The parts of these registers which aren’t sent to the output are $x_i$ ,$x_{i+127}$ and $x_{i+254}$ . We can use the similar notation for other registers. In this section $\mathbf{a}_1$, $\mathbf{a}_2$, … , $\mathbf{a}_5$ are all zero or all one vectors with the same probability of 0.5.

We try to use (9), in the Fischer’s attack. So we have:

$$R_3' = R_2' \oplus R_2 \oplus R_3 \oplus R_4 \oplus \mathbf{a}_1 \quad (11)$$

or

$$R_2' \oplus R_3 = R_2 \oplus R_3' \oplus R_4 \oplus \mathbf{a}_1 \quad (12)$$

Similarly we have the following relations:

$$R_2' = R_1' \oplus R_1 \oplus R_2 \oplus R_3 \oplus \mathbf{a}_2 \quad (13)$$

or

$$R_1 = R_1' \oplus R_2 \oplus R_2' \oplus R_3 \oplus \mathbf{a}_2 \quad (14)$$

and

$$R_1' = R_0' \oplus R_0 \oplus R_1 \oplus R_2 \oplus \mathbf{a}_3 \quad (15)$$

or

$$R_0' = R_0 \oplus R_1' \oplus R_2 \oplus R_1 \oplus \mathbf{a}_3 \quad (16)$$

Relations (12), (14) and (16) are easily obtained from (11), (13) and (15), respectively. Substituting (12) in (14) and then in (16) we have:

$$R_0' = R_0 \oplus R_2 \oplus R_3' \oplus R_4 \oplus \mathbf{a}_1 \oplus \mathbf{a}_2 \oplus \mathbf{a}_3 \quad (17)$$

Relation (17) is hold between the top halves of the registers. The top half of $R_0'$ ($x_{i+127}$) is not available to the attacker so the corresponding relation is as follows:

$$x_{i+127} = k_i \oplus k_{i+2} \oplus k_{i+130} \oplus k_{i+4} \oplus \mathbf{a}_1 \oplus \mathbf{a}_2 \oplus \mathbf{a}_3 \quad (18)$$

where $x_{i+127}$ , $k_i$ , $k_{i+2}$ , $k_{i+4}$ and $k_{i+130}$ are top halves of $R_0'$ , $R_0$ , $R_2$ , $R_4$ and $R_3'$, respectively, and $\mathbf{a}_1 \oplus \mathbf{a}_2 \oplus \mathbf{a}_3$ is an all zero or all one vector so two cases exist for the value of $x_{i+127}$ that are complementary of each other. For the second and the third rows of the registers in Fig. 2 we have:

$$R_2'' = R_1'' \oplus R_1' \oplus R_2' \oplus R_3' \oplus \mathbf{a}_4 \quad (19)$$

or

$$R_1'' \oplus R_2' = R_2'' \oplus R_1' \oplus R_3' \oplus \mathbf{a}_4 \quad (20)$$

and

$$R_1'' = R_0'' \oplus R_0' \oplus R_1' \oplus R_2' \oplus \mathbf{a}_5 \quad (21)$$

or

$$R_0' = R_1' \oplus R_0'' \oplus R_1'' \oplus R_2' \oplus \mathbf{a}_5 \quad (22)$$

Substituting (20) in (22) we have:

$$R_0' = R_0'' \oplus R_2'' \oplus R_3' \oplus \mathbf{a}_4 \oplus \mathbf{a}_5 \quad (23)$$

Writing (23) for the top halves of the registers, following relation is concluded:

$$x_{i+127} = k_{i+254} \oplus k_{i+256} \oplus k_{i+130} \oplus \mathbf{a}_4 \oplus \mathbf{a}_5 \quad (24)$$

where $k_{i+130}$ , $k_{i+254}$ and $k_{i+256}$ are top halves of $R_3'$, $R_0''$ and $R_2''$, respectively, and $\mathbf{a}_4 \oplus \mathbf{a}_5$ is an all zero or all one vector. Thus two cases are possible for $x_{i+127}$ . If the sequence is output of the MAG stream cipher, the possible cases of (18) and (24) are matched but for a random sequence these 4 possible 16 bit words concluded from these relations are matched with the probability of $2^{-15}$. Therefore having 514 successive bytes of the output sequence , i.e. $k_i$ , $k_{i+1}$ , … and $k_{i+256}$, (each $k_i$ is a 2 byte word) it is possible to distinguish the keystream of MAG from a random sequence with the miss and false alarm probability of 0 and $2^{-15}$, respectively. If we use more keystream it is possible to reduce the false probability. For example having just two more 16 bit words we can find similar relations for $x_{i+129}$ and reduce the false alarm probability to $2^{-30}$ .

## 7. Conclusion

In this paper, using the Fischer’s attack on the first version of MAG, a distinguishing attack on a modified version of this stream cipher was presented. The attack requires 514 successive bytes of the keystream to distinguish it from a random sequence. Computational complexity of the attack is 5 xor and 2 comparison operations between 16 bit words. It was shown that the miss and false alarm probability of the attack are 0 and $2^{-15}$, respectively. In addition, we showed that it seems that performing the Simpson and Henricksen’s attack on all versions of MAG is impossible in general (nonzero values of E). Therefore, the attack presented in this paper on the modified version of MAG is the only one, by the present time.

**Acknowledgment** The work was supported by Sa-Iran company. We thank them for their patronage.

## References

[1] R. Vuckovac, “MAG: My Array Generator (a new strategy for random number generation)”, eSTREAM, ECRYPT Stream Cipher Project, Report 2005/014, 2005. http://www.ecrypt.eu.org/stream.

[2] R. Vuckovac, “MAG Alternating Methods Notes”, eSTREAM, ECRYPT Stream Cipher Project, Report 2005/068, http://www.ecrypt.eu.org/stream.

[3] R. Vuckovac, “MAG Cipher Design Notes”, eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, http://www.ecrypt.eu.org/stream

[4] S. Fischer, “Analysis of Lightweight Stream Ciphers” Ph. D. thesis, Lund University (Sweden), 2008.

[5] L. Simpson, M. Henricksen, “Improved Cryptanalysis of MAG” In L. Batten and R. Safavi-Naini, editors, ACISP, LNCS 4058, pp. 64-75, Springer- Verlag Berlin Heideberg 2006.