Improved Linear Distinguishers for SNOW 2.0

Kaisa Nyberg[1,2] and Johan Wallén[1]

[1]Helsinki University of Technology and [2]Nokia Research Center, Finland
Email: kaisa.nyberg@nokia.com; johan.wallen@tkk.fi

Abstract. In this paper we present new and more accurate estimates of the biases of the linear approximation of the FSM of the stream cipher SNOW 2.0. Based on improved bias estimates we also find a new linear distinguisher with bias $2^{-86.9}$ that is significantly stronger than the previously found ones by Watanabe et al. (2003) and makes it possible to distinguish the output keystream of SNOW 2.0 of length 2^{174} words from a truly random sequence with workload 2^{174}. This attack is also stronger than the recent distinguishing attack by Maximov and Johansson (2005). We also investigate the diffusion properties of the MixColumn transformation used in the FSM of SNOW 2.0 and present some evidence why much more efficient distinguishers may not exist.
Keywords. Stream cipher, SNOW 2.0, linear masking method, modular addition

1 Introduction

Key stream generators are widely used in practise for random number generation and data encryption as stream ciphers. The history of this type of cryptographic primitive has not always been glorious. Most recently, algebraic cryptanalysis method has been successfully applied to a number of stream ciphers. On the other hand, there is no scientific evidence that stream ciphers are inherently less secure than block ciphers. To strengthen the scientific foundations of the security of stream ciphers the ECRYPT NoE launched in November 2004 a new multi-year project eSTREAM, the ECRYPT Stream Cipher Project, to identify new stream ciphers that might become suitable for widespread adoption [8].

In this paper new results of the strength of the stream cipher SNOW 2.0 against linear approximation are presented. SNOW 2.0 was proposed by Ekdahl and Johansson in [3] as a strengthened version of SNOW 1.0, which was a NESSIE candidate. Currently SNOW 2.0 is considered as one of the most efficient stream ciphers. It is used for benchmarking the performance of stream ciphers by the eSTREAM project. SNOW 2.0 has also been taken as a starting point for the ETSI project on a design of a new UMTS encryption algorithm [4].

Linear methods have been widely used to analyse stream ciphers. In addition to the traditional methods such as linear complexity and correlation analysis, attacks based on linear cryptanalysis method have been succesfully launched against stream ciphers. One of the reasons why SNOW 1.0 was rejected by the NESSIE project was its vulnerability against a distinguishing attack using linear cryptanalysis [2, 3].

Distinguishing attacks using linear cryptanalysis (linear masking) were previously applied to SNOW 2.0 by Watanabe et al., to see if the designers of the algorithm learnt

their lesson [11]. An efficient distinguisher can be used to detect statistical bias in the key stream, and in some cases, also derive the key or initial state of the key stream generator. In this paper we show that the estimates of the strength of the linear approximations given in [11] were not accurate. Their best masking was estimated to have bias $2^{-112.25}$, while the true value is closer to $2^{-107.26}$. Further we find a linear masking of the FSM of SNOW 2.0 with bias $2^{-86.89}$. Using this masking a distinguishing attack on SNOW 2.0 can be given which requires 2^{179} bits of the key stream and 2^{174} operations. This attack also superceeds the attack by Maximov an Johansson in [7].

The paper is structured as follows. In Section 2 we present the details of SNOW 2.0 as needed in the investigations of this paper. Section 3 explains the linear masking method on SNOW 2.0 and summarises our results. In Section 4 we analyse assumptions under which the bias values in [11] were computed, and show that the assumptions do not hold. We also give examples of large deviations from correct values and investigate the behaviour of linear approximation of modular addition with three inputs. The main too is an algorithm for computing the correlations for modular addition with an arbitrary number of inputs, which we present in Annex A. In Section 5 we present our observations about the structure of SNOW 2.0 and other results from mask searches. Finally, in Section 6 we give some results about resistance against linear distinguishing attacks for SNOW 3G, which is a modification of SNOW 2.0 by ETSI SAGE intended to become a second encryption algorithm for the UMTS system. A draft version of SNOW 3G can be found in [4]. The description of the final version of SNOW 3G and rationale of its design can be found in the design and development report [5].

2 The stream cipher SNOW 2.0

The structure of SNOW 2.0 is depicted in Figure 1. The running engine is a linear feedback shift register (LFSR) consisting of 16 words of length 32 bits each. The LFSR is defined over $GF(2^{32})$ with feedback polynomial

$$\alpha x^{16} + x^{14} + \alpha^{-1} x^5 + 1 \in GF(2^{32})[x]$$

where $\alpha \in GF(2^{32})$ is a root of the polynomial

$$x^4 + \beta^{23} x^3 + \beta^{245} x^2 + \beta^{48} x + \beta^{239} \in GF(2^8)[x]$$

and β is a root of the polynomial

$$x^8 + x^7 + x^5 + x^3 + 1 \in GF(2)[x].$$

The bitwise xor of two 32-bit blocks is denoted by $\oplus$ and addition modulo 2^{32} is denoted by $\boxplus$. The LFSR feeds into a finite state machine (FSM). The FSM has two 32-bit registers $R1$ and $R2$. The state of the LFSR at time t is denoted by $(s_{t+15}, \ldots, s_t)$. The input to the FSM is s_{t+15} and s_{t+5} and the output F_t of the FSM is calculated as

$$F_t = (s_{t+15} \boxplus R1_t) \oplus R2_t,$$

for all $t \geq 0$, where we have denoted by $R1_t$ and $R2_t$ the contents of the registers $R1$ and $R2$, respectively, at time t. Then the output z_t of the keystream generator is given as

$$z_t = F_t \oplus s_t.$$

The contents of $R1$ is updated as $s_{t+5} \boxplus R2_t$ and the contents of $R2$ is updated as $S(R1_t)$ where the transformation S is composed of four parallel AES S-boxes followed by the AES MixColumn transformation. For the purposes of this paper only the details of the LFSR and the FSM are needed. For a complete description of SNOW 2.0 we refer to the paper [3] by Ekdahl and Johansson.

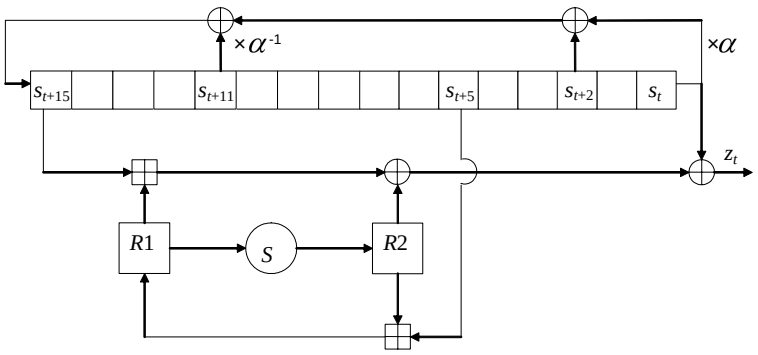


Fig. 1. SNOW 2.0

3 The linear masking method on SNOW 2.0

3.1 Linear masking of the FSM

We denote $\mathbf{F}_2 = GF(2)$. Let n be a non-negative integer. Given two vectors $x = (a_{n-1}, \ldots, a_0)$ and $y = (b_{n-1}, \ldots, b_0) \in \mathbf{F}_2^n$, let $x \cdot y$ denote the standard inner product $x \cdot y = a_{n-1}b_{n-1} \oplus \ldots \oplus a_0 b_0$. A constant vector which is used to compute inner product

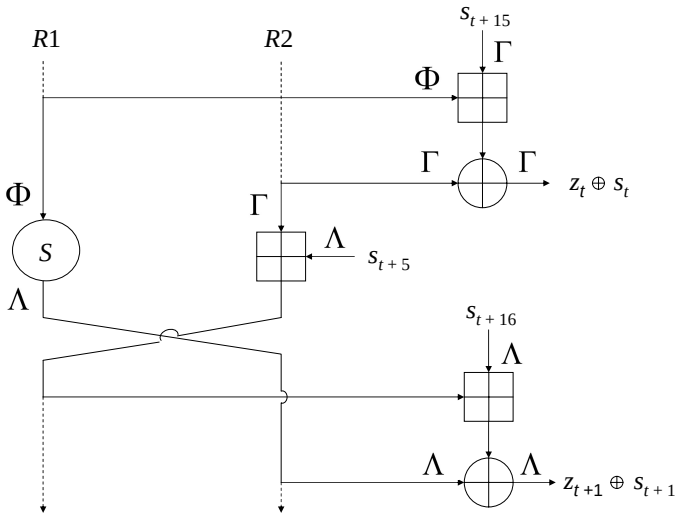


Fig. 2. Linear masking of SNOW 2.0

with inputs (outputs) of a function is called a linear input (output) mask of the function. Given a linear mask $\Gamma \in \mathbf{F}_2^n$ and an element $\alpha \in \mathbf{F}_2^n$, we denote by $\Gamma\alpha$ the linear mask, which satisfies the following equality

$$\Gamma\alpha \cdot x = \Gamma \cdot \alpha x, \text{ for all } x \in \mathbf{F}_2^n,$$

where the product αx is taken in $GF(2^{32})$. Let m and n be positive integers. Given a functional dependency $F : \mathbf{F}_2^n \to \mathbf{F}_2^m$, a linear input mask $\Lambda \in \mathbf{F}_2^n$ and a linear output mask $\Gamma \in \mathbf{F}_2^m$, the strength of the linear approximate relation $\Gamma \cdot F(x) = \Lambda \cdot x$, for $x \in \mathbf{F}_2^n$, is measured using its correlation

$$\begin{aligned}&\mathrm{cor}_F(\Lambda, \Gamma) = \mathrm{cor}(\Gamma \cdot F(x) \oplus \Lambda \cdot x)\\ &= 2^{-n}(\#\{x \in \mathbf{F}_2^n : \Gamma \cdot F(x) \oplus \Lambda \cdot x = 0\} - \#\{x \in \mathbf{F}_2^n : \Gamma \cdot F(x) \oplus \Lambda \cdot x = 1\}).\end{aligned}$$

For the purposes of this paper we use a derived value $\epsilon_F(\Lambda, \Gamma) = |\mathrm{cor}_F(\Lambda, \Gamma)/2|$ and call it the bias of the linear approximate relation $\Gamma \cdot F(x) = \Lambda \cdot x$.

The linear masking method was applied to SNOW 2.0 in [11]. The linear approximation of the FSM of SNOW 2.0 used in [11] is depicted in Figure 2 in a slightly generalised form. In [11], it was always assumed that the output masks Γ at time t and Λ time $t + 1$ are equal, for all $t \geq 0$. In case when they are allowed to be different, it is straightforward to verify that the main distinguishing equation, [11], Equation (12),

takes the following form

$$\begin{aligned} &\Gamma \cdot (z_{t+16} \oplus z_{t+2}) \oplus \Gamma\alpha \cdot z_t \oplus \Gamma\alpha^{-1} \cdot z_{t+11} \oplus \\ &\Lambda \cdot (z_{t+17} \oplus z_{t+3}) \oplus \Lambda\alpha \cdot z_{t+1} \oplus \Lambda\alpha^{-1} \cdot z_{t+12} = 0, \end{aligned} \tag{1}$$

for all $t \geq 0$. This relation is obtained by using the approximation depicted in Figure 2 four times: firstly, two times with the mask pair Γ, Λ at time $t+2$ and $t+16$, then once with the mask pair $\Gamma\alpha, \Lambda\alpha$ at time t, and finally once with the mask pair $\Gamma\alpha^{-1}, \Lambda\alpha^{-1}$ at time $t+11$. Given the biases, these four approximations can be combined and the total bias value computed using the Piling Up Lemma [6]. Similarly as in [11] we denote by $\epsilon_{FSM}(\Lambda, \Gamma)$ the bias of the linear approximate relation of Figure 2. Hence the total bias $\epsilon(\Lambda, \Gamma)$ of the linear distinguisher (1) is calculated as

$$\epsilon(\Lambda, \Gamma) = 8\epsilon_{FSM}(\Lambda, \Gamma)^2 \epsilon_{FSM}(\Lambda\alpha, \Gamma\alpha)\epsilon_{FSM}(\Lambda\alpha^{-1}, \Gamma\alpha^{-1}).$$

We also introduce a new mask Φ, see Figure 2, whose role will be explained in subsection 4.2.

3.2 Our Results

We implemented a new wider mask search over the FSM SNOW 2.0 to achieve more accurate and improved estimates of the total bias of the linear distinguisher (1). In particular,

- we allow output masks Γ and Λ be different; and
- we improve the accuracy of the estimates of the bias values.

The effect of the first change turns out not to be significant. Suitable candidates for Γ were searched by first identifying Γ such that it performs reasonably well with Λ in the linear approximation. Here algorithms from [10] were used. Still, for a given Λ, the total bias of the distinguisher of Equation (1) is usually higher with $\Gamma = \Lambda$ than with any $\Gamma \neq \Lambda$. In some cases higher biases where obtained with $\Gamma \neq \Lambda$ but the achieved bias values were far from the best. Two show that such cases exist we give an example in Table 1.

Λ	Γ	$\epsilon(\Lambda, \Gamma)$
0x04400240	0x04400240	0
0x04400240	0x04400360	$2^{-122.29}$
0x08400280	0x08400280	$2^{-140.67}$
0x08400280	0x084003a0	$2^{-124.41}$

Table 1. Two masks Λ with higher bias with $\Gamma \neq \Lambda$.

The strongest linear approximation of the FSM of SNOW 2.0 found in our search is using the distinguisher (1) with $\Lambda = \Gamma =$ 0x00018001. The values of the biases of

	mask value	ϵ_{FSM}
Λ	0x00018001	$2^{-15.496}$
$\Lambda\alpha$	0xc7000180	$2^{-27.676}$
$\Lambda\alpha^{-1}$	0x0180015c	$2^{-31.221}$

Table 2. The mask $\Lambda = \Gamma$ with the highest bias $2^{-86.89}$ for the linear distinguisher (1).

	mask value	ϵ_{FSM} estimate in [11]	ϵ_{FSM} our estimate
Λ	0x0303600c	$2^{-27.61}$	$2^{-24.48}$
$\Lambda\alpha$	0x0c030360	$2^{-27.61}$	$2^{-24.49}$
$\Lambda\alpha^{-1}$	0x03600c63	$2^{-32.42}$	$2^{-36.82}$

Table 3. Improved estimates of the biases for the best mask in [11]

the linear approximation of the FSM are given in Table 2. They result in the total bias value of $2^{-86.89}$. This value is significantly higher than the bias value $2^{-112.25}$ achieved using the best linear mask 0x0303600 reported in [11]. The difference of results is due to the fact that Watanabe et al., used different and less accurate estimates of bias values as will be explained in more detail in Section 4. In Table 3 we give the new and more accurate bias values for the best mask in [11] which show that its strength was originally underestimated. Our new estimate of the total bias is $2^{-107.26}$.

We also looked at other linear approximations of SNOW 2.0 FSM than the one depicted in Figure 1. These will be discussed in Section 5.

4 Improved approximation over dependent functions

In [11] the biases of the linear approximations of the FSM were calculated under the assumption that all linear approximations over various nonlinear functions involved in the approximation depicted in Figure 2, the S-box ensemble S and the three additions modulo 2^{32}, are independent. Such an assumption is often reasonable in practise since there is no evidence of dependency although it is not possible to prove the opposite either. However, in the linear approximation of the FSM of SNOW 2.0, see Figure 2, there are two cases where combinations of two subsequent approximations are strongly dependent. The first such case is when one approximates over two subsequent additions modulo 2^{32}, that is, the output from the first addition is an input to the second addition. We will show in the next subsection that this must be handled as addition modulo 2^{32} with three inputs and not as two independent additions with two inputs. The second case is due to the fact that the value in register $R1$ is both an input to the modular addition $\boxplus$ and an input to the S-box ensemble S.

4.1 Linear approximation over two subsequent modular additions

In this section we investigate the behaviour of two consecutive modular additions with two inputs each, where the output from the first addition is input to the second one. Clearly such a composition is equivalent to one modular addition with three inputs. Previously, results and algorithms for computing biases of linear approximations have been presented only for the case with two inputs, see [10]. The basic algorithm for computing the bias for given input and output masks can be straightforwardly generalised to the case with an arbitrary finite number of inputs, and is given in Annex A. Our results show that the behaviour of modular addition under linear approximation depends to a large extent on the number of inputs. As a first result we demonstrate in Table 4 the reasons why the best linear mask found by us was not found by Watanabe et al. We denote by ϵ_+ the bias of linear approximation of modulo 2^{32} addition with two inputs and by ϵ_{++} the same value with three inputs using the same given mask value for all input and output masks. The value $2\epsilon_+^2$ in the middle is the one used in [11] in place of ϵ_{++}.

	mask value	ϵ_+	$2\epsilon_+^2$	ϵ_{++}
Λ	0x00018001	2^{-2}	2^{-3}	$2^{-2.58}$
$\Lambda\alpha$	0xc7000180	2^{-26}	2^{-51}	$2^{-6.75}$
$\Lambda\alpha^{-1}$	0x0180015c	2^{-7}	2^{-13}	$2^{-7.71}$

Table 4. Biases of linear approximation of addition with 2 inputs and 3 inputs for the best mask.

mask value	ϵ_+	ϵ_{++}
0x00000010	2^{-5}	$2^{-2.61}$
0x00000100	2^{-9}	$2^{-2.59}$
0x00001000	2^{-13}	$2^{-2.58}$
0x00010000	2^{-17}	$2^{-2.58}$
0x00100000	2^{-21}	$2^{-2.58}$
0x01000000	2^{-25}	$2^{-2.58}$

Table 5. Biases of linear approximation of addition with 2 inputs and 3 inputs for some 1-bit masks.

It is also interesting to observe how differently linear approximation with one-bit masks behave over modular addition. In the two input case, the strength of the linear approximation degrades when moving towards the most significant bits. For addition with three inputs the bias values are almost the same in all positions as shown in Table 5. Moreover, we observed that linear approximation over modular addition with

three inputs is more flexible and gives better bias values also when not all input masks are the same. The same holds in general for very sparse masks. Therefore we made an exhaustive search over all masks Λ with at most five non-zero bits. The results are given in Section 5.

4.2 Linear approximation over composition of different functions

The modular addition $\boxplus$ and an input to the S-box ensemble S have the contents of register $R1$ as common inputs. Since S is invertible, we can compose these functions as follows:

$$f : y, z \mapsto S^{-1}(y) \boxplus z,$$

for all y that are output from the S-box ensemble and for all $z = s_{t+15}$. The task is to compute the correlation between the following linear combination of inputs and linear combination of outputs

$$\mathrm{cor}(\Gamma \cdot f(y, z) \oplus \Gamma \cdot z \oplus \Lambda \cdot y).$$

By applying a well-known theorem about correlations over composed functions, see e.g. [9], Theorem 3, we get that the correlation can be computed as a sum of partial correlations over all intermediate linear masks Φ as follows:

$$\begin{aligned} &\mathrm{cor}(\Gamma \cdot f(y, z) \oplus \Gamma \cdot z \oplus \Lambda \cdot y) \\ &= \sum_{\Phi} \mathrm{cor}(\Gamma \cdot (w \boxplus z) \oplus \Phi \cdot w \oplus \Gamma \cdot z)\mathrm{cor}(\Phi \cdot S^{-1}(y) \oplus \Lambda \cdot y) \\ &= \sum_{\Phi} \mathrm{cor}(\Gamma \cdot (w \boxplus z) \oplus \Phi \cdot w \oplus \Gamma \cdot z)\mathrm{cor}(\Phi \cdot x \oplus \Lambda \cdot S(x)). \end{aligned} \tag{2}$$

Considering the addition modulo 2^{32} and the S-box ensemble S as independent functions is equivalent of taking just one term in the sum (2). Moreover, in [11] this one term was selected with $\Phi = \Lambda = \Gamma$. We observed that this may cause large deviations from the true value. On the other hand, including all terms of the sum would mean unnecessarily large amount of work. It turns out that including all terms with $\mathrm{cor}(\Gamma \cdot (w \boxplus z) \oplus \Phi \cdot w \oplus \Gamma \cdot z) \geq 2^{-24}$ yields sufficiently accurate estimates of the total correlation over the composed function. To search for all such linear masks Φ we used the algorithms by Wallén [10] (see Annex A). This explains the role of the linear mask Φ in Figure 2.

5 More searches

5.1 Reducing the number of active S-boxes

One strategy to increase the total bias of the linear approximation would be to limit the number of active S-boxes in the S-box ensemble S. Given an output mask Λ of S let

us denote by Ω the mask such that $\Omega \cdot x = \Lambda \cdot Mx$, for all 32-bit values x, where M denotes the MixColumn transformation of the AES. Our best mask $\Lambda =$ `0x00018001` corresponds to the mask $\Omega =$ `0x0041c01`. It means that only three S-boxes are active in the approximation of the FSM. However, in linear approximation with $\Lambda\alpha$ and $\Lambda\alpha^{-1}$ all four S-boxes are active. This means that in our best approximation of relation (1) the total number of active S-boxes is 14.

The MixColumn transformation is known to have good diffusion properties, more precicely, the total number of nonzero octets in (Ω, Λ) mask pairs is at least five. For linear approximation of SNOW 2.0 the diffusion properties of MixColumn must be investigated in combination with multiplication by α and α^{-1}. More precisely, it would be interesting to know exactly how many S-boxes at least are involved in the linear approximation (1). For this purpose, we studied all masks Λ such that the corresponding Ω has at most two non-zero octets. For each such Λ we computed the masks $\Lambda\alpha$ and $\Lambda\alpha^{-1}$ and their related Ω-masks, for which the number of non-zero octets was determined. Finally the total number of non-zero octets involved in the four FSM approximations in (1) was computed for each Λ. The same search was performed also for all Λ such that the input mask to M corresponding to $\Lambda\alpha$ ($\Lambda\alpha^{-1}$, respectively) has at most two non-zero octets. The minimum number of active S-boxes was found to be 7, and there are four masks Λ having this property. They are `0x64ad5846`, `0xad584664`, `0x55bcc50d` and `0x0d55bcc5`, and their respective one-octet input masks to MixColumn M are: `0xd7000000`, `0x000000d7`, `0x00210000` and `0x00002100`. However, none of these four masks Λ has a second mask Γ with a non-zero total bias over approximation (1). This follows from the results of a wider search we explain next.

We made a comprehensive search over all masks Λ such that the input mask Ω to M has at most two non-zero octets. This means limiting the search for such masks Λ that in approximation over the S-box-ensemble S at most two S-boxes are active. For any such Λ there was no Γ such that the linear approximate relation (1) would have a non-zero bias. This is obviously a strength in the structure of SNOW 2.0. We can only give a heuristic explanation of the reasons why this happens. Assume Λ is such that only two S-boxes are active. Then one can find an approximation over the FSM with a pretty good bias. Then Γ typically has two or three nonzero octets. Four non-zero octets may be possible in theory (we could not find any examples) but then in two of the octets only the least or the most significant bit is non-zero. In other words, the mask is sparse. Also when modified by α (or α^{-1}) the sparse structure is preserved. On the other hand, when Λ is modified by α (or α^{-1}) then almost always all four S-boxes will be active, and consequenty, the mask Φ has four nonzero octets. Therefore in about all cases, if not all, the mask $\Gamma\alpha$ and the mask Φ that fits over the S-box ensemble with $\Lambda\alpha$ have different structure. The same holds for the approximation of the FSM with $\Gamma\alpha^{-1}$ and $\Lambda\alpha^{-1}$. Since both of them should work to make the entire approximation work, the chances are negligible.

5.2 Sparse masks

As explained above we were not able to significantly reduce the number of S-boxes that are active in the linear approximate relation (1). On the other hand, we observed that the modular addition with three inputs can be efficiently approximated using sparse masks.

This is also well exemplified by the best linear distinguisher we found, which is based on a three-bit linear mask. Motivated by this observation we made a complete search over all masks Λ with at most five non-zero bits, allowing as in all our searches the mask Γ to be different from Λ. For one and two-bit masks there were no results. For the three-bit masks it turned out that we already found the best one. The best masks with four or five nonzero bits and their respective bias values are given in Table 6. The total bias of the linear distinguisher (1) with the four-bit mask $\Lambda = \Gamma =$ `0x40100060` is $2^{-89.95}$, and with the five-bit mask $\Lambda = \Gamma =$ `0x00040701` it is $2^{-89.25}$.

	mask value	ϵ_{FSM}
Λ	`0x40100060`	$2^{-18.49}$
$\Lambda\alpha$	`0x02401000`	$2^{-26.94}$
$\Lambda\alpha^{-1}$	`0x10006029`	$2^{-29.02}$
Λ	`0x00040701`	$2^{-18.72}$
$\Lambda\alpha$	`0x75000407`	$2^{-27.47}$
$\Lambda\alpha^{-1}$	`0x04070100`	$2^{-27.35}$

Table 6. The best four-bit and five-bit linear masks Λ

5.3 Three-round distinguisher

Also other more complex distinguishers were investigated. In particular, we looked at the distiguisher which involves output at time $t-1$, t and $t+1$, two instances of the S-box ensemble S and five (or four, if $\Pi = 0$) modular additions $\boxplus$ out of which two collapse into one addition with three inputs, see Figure 4 in Annex B. The resulting linear approximative relation is given in equation (3) in Annex B. Such a three-round distinguisher could compete with the two-round one only if the number of active S-boxes could be significantly reduced. However, this does not seem to be possible, for the same reason why the two-round distinguisher does not have non-zero bias with small number of active S-boxes as explained above in Section 5.1. Moreover, in approximations for (3), approximation over the latter S-box ensemble involves at least seven active S-boxes. The absolute minimum for the first S-box ensemble is four active S-boxes. Since the largest achievable bias of linear approximation over the AES S-box is 2^{-4} we get a theoretical upperbound of $2^{10}(2^{-4})^{11} = 2^{-34}$ to the bias of (3) for SNOW 2.0. The largest bias for the approximation (3) we have seen in practise is $2^{-202.17}$. In this case, the masks Γ, Π and Λ had 2,3 and 4 non-zero octets, respecively. In the first S-box ensemble, totally 6 S-boxes were active, and in the second S-box ensemble 10 S-boxes were active.

6 Linear distinguishers for SNOW 3G

During its still relatively short lifetime SNOW 2.0 has gained confidence as demonstrated by the fact that it has been selected as a starting point for a few new designs,

see [8]. Most prominently, a draft for a new encryption algorithm for the UMTS system was recently made public in [5]. It is called SNOW 3G in and is depicted in Figure 3. This design preserves all features of SNOW 2.0, e.g. $S1 = S$, but adds a third register $R3$ to the FSM and a transform denoted by $S2$. The function $S2$ has been selected to strengthen the FSM against algebraic cryptanalysis as response to the concerns expressed in [1, 5].

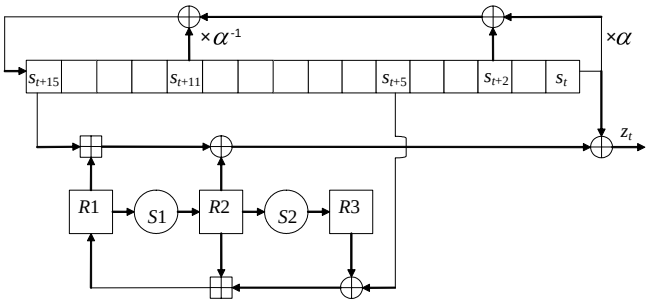


Fig. 3. SNOW 3G

The first, later rejected, choice for $S2$ was a non-bijective 32-to-32-bit S-box [4]. It was constructed from a single eight-to-one bit Boolean function $V8$ by selecting for each output bit a set of eight input bits. Then the output bit value is computed from the selected input bits using $V8$. The input sets are selected in such a way that any two sets has at most three bits in common. The Boolean function $V8$ is not balanced, hence $S2$ is not a bijection.

The simplest distinguisher for this version of SNOW 3G is obtained by using linear masking over two and half rounds of the FSM and it is depicted in Figure 5. The linear approximate relation is the same as in (1). In addition to the linear distinguisher depicted in Figure 2 it involves approximations over $S2$, where the input mask is all-zero. The mask search is very similar to the two round distinguisher for SNOW 2.0. We just need to add the bias of approximation over $S2$ to it. Hence it is no surprise that the best Λ, Γ pair we found for this distinguisher is the same as for SNOW 2.0, that is, $\Lambda = \Gamma =$ `0x00018001`. The total bias of the linear approximation of Figure 5 is $2^{-137.01}$.

In the final version of SNOW 3G the transformation $S2$ is otherwise identical to $S1$ but the AES S-box is replaced by a bijective mapping derived from a Dickson polyno-

mial. This S-box has maximum linear bias of 2^{-3}. A three-round linear distinguisher for SNOW 3G is depicted in Figure 6. We showed in Section 5.3 that the minimum number of S-boxes in the approximations over $S2$ is at least seven. The same holds for the second instance of $S1$. However, the input and output masks of the first instance of $S1$ are not modified by α or α^{-1}. Nevertheless, at least four active S-boxes are needed. Hence there are always at least eleven active AES S-boxes and seven active $S2$ S-boxes, giving an upper bound of $2^{17}(2^{-4})^{11}(2^{-3})^7 = 2^{-48}$ to the bias of any three-round linear approximation of SNOW 3G. This bound is not tight. The true bias values will most likely be significantly reduced due to the biases of linear approximations over the modular additions.

7 Conclusions

It is well known that the Piling Up Lemma cannot be applied to combine linear approximations over consecutive functions in cipher constructions unless there is some evidence that the output from the first function is practically independent of the input to the second function. We showed that in [11] the Piling Up Lemma was used in case where the output from the first function is identical to the input to the second function. We showed not only how to compute correctly the estimates of the bias values but also implemented wide mask searches to find new and significantly stronger distinguishers that escaped the searches by Watanabe et al.

Some mask searches that were limited to certain types of linear masks failed to produce any results with non-zero bias. For example, we could demonstrate that it is impossible to significantly reduce the number of active S-boxes when approximating over the S-box ensemble S of SNOW 2.0. The same holds to other more complex distinguishers of SNOW 2.0 as well as to the recently presented new SNOW variant SNOW 3G, and is preserved as long as the feedback polynomial does not have a low degree multiple, which is a trinomial or a four-term polynomial with only two different coefficients. This gives some evidence about the strength of the SNOW design against cryptanalysis using the linear masking method.

Acknowledgements

We wish to thank Jukka Valkonen for implementing all mask searches needed to complete this work and the Krypto project of the Finnish Defence Forces for making it possible. We also wish to thank Emilia Käsper for providing optimised implementations of some specific parts. Finally, we owe many thanks to the members of the SAGE group, and to Matt Robshaw, in particular, for invaluable discussions.

References

1. Olivier Billet and Henri Gilbert. Resistance of SNOW 2.0 against algebraic attacks. In *CT-RSA 2005*, pages 19–28, 2004.

2. Don Coppersmith, Shai Halevi, and Charanjit Jutla. Cryptanalysis of stream ciphers with linear masking. In *Advances in Cryptology - Crypto 2002, LNCS 2442, Springer-Verlag* , pages 515–532, 2002.
3. Patrick Ekdahl and Thomas Johansson. A new version of the stream cipher SNOW. In *Selected Areas in Cryptography, SAC 2002, LNCS 1233, Springer-Verlag* , pages 37–46, 2002.
4. ETSI/SAGE. Specification of the 3GPP confidentiality and integrity algorithms UEA2 & UIA2. Document 2: SNOW 3G specification, draft version 0.5, August 2005. `http://www.3gpp.org/ftp/tsg_sa/WG3_Security/TSGS3_40_Slovenia/Docs/S3-050579%%20.zip`.
5. ETSI/SAGE. Specification of the 3GPP confidentiality and integrity algorithms UEA2 & UIA2. Document 5: Design and evaluation report, version: 1.0 `http://www.3gpp.org/ftp/tsg_sa/WG3_Security/TSGS3_42_Bangalore/Docs/S3-06018%0.zip`.
6. Mitsuru Matsui. Linear Cryptanalysis Method for DES Cipher. In *Advances in Cryptology, Eurocrypt 1993, LNCS 765, Springer-Verlag* , pages 386–397, 1994.
7. Alexander Maximov and Thomas Johansson. Fast Computation of Large Distributions and Its Cryptographic Applications. In *Asiacrypt 2005, LNCS 3788, Springer-Verlag* , pages 313–332, 2005.
8. ECRYPT NoE. eSTREAM, the ECRYPT stream cipher project, 2005. `http://www.ecrypt.eu.org/stream/`.
9. Kaisa Nyberg. Correlation theorems in cryptanalysis. *Discrete Applied Mathematics*, pages 177–188, 111 2001.
10. Johan Wallén. Linear Approximations of Addition Modulo 2^n. In *Fast Software Encryption, FSE 2003, LNCS 2887, Springer-Verlag* , pages 261–273, 2003.
11. Dai Watanabe, Alex Biryukov, and Christophe De Cannière. A Distiguishing Attack of SNOW 2.0 with Linear Masking Method. In *Selected Areas in Cryptography, SAC 2003, LNCS 3006, Springer-Verlag* , pages 222–233, 2004.

A Linear Approximation of Addition Modulo 2^n

A.1 Notation

We identify the integers in $\{0, \ldots, 2^n - 1\}$ with the vectors in $\mathbf{F}_2^n$ by the natural correspondence that identifies the integer whose binary expansion is $\sum_{i=0}^{n-1} a_i 2^i$ with the vector $(a_{n-1}, \ldots, a_1, a_0)$. Given k n-bit integers $x^{(h)}$, $h = 1, \ldots, k$, the sum $(x^{(1)} + \cdots + x^{(k)}) \bmod 2^n$ carries over to a function from $(\mathbf{F}_2^n)^k$ to $\mathbf{F}_2^n$. Addition in $\mathbf{F}_2$ and $\mathbf{F}_2^n$ is always denoted by $\oplus$.

For vectors $x = (a_{n-1}, \ldots, a_0)$ and $y = (b_{n-1}, \ldots, b_0) \in \mathbf{F}_2^n$, let $x \cdot y$ denote the standard inner product $x \cdot y = a_{n-1}b_{n-1} \oplus \cdots \oplus a_0 b_0$. For k tuples of vectors in $\mathbf{F}_2^n$, $x^{(1)}, \ldots, x^{(k)}$ and $y^{(1)}, \ldots, y^{(k)}$, we set $(x^{(1)}, \ldots, x^{(k)}) \cdot (y^{(1)}, \ldots, y^{(k)}) = x^{(1)} \cdot y^{(1)} \oplus \cdots \oplus x^{(k)} \cdot y^{(k)}$. A linear approximation of the sum modulo 2^n with k inputs is an approximate relation of the form

$$u \cdot (x^{(1)} \boxplus \cdots \boxplus x^{(k)}) = (x^{(1)}, \ldots, x^{(k)}) \cdot (w^{(1)}, \ldots, w^{(k)})$$

where $u \in \mathbf{F}_2^n$ and $w^{(h)} \in \mathbf{F}_2^n$, $h = 1, \ldots, k$ are the mask vectors. The strength of the approximation is measured by the correlation

$$\begin{aligned} &\mathrm{cor}(u; w^{(1)}, \ldots, w^{(k)}) \\ &= 2\mathbf{Pr}[u \cdot (x^{(1)} \boxplus \ldots \boxplus x^{(k)}) = (x^{(1)}, \ldots, x^{(k)}) \cdot (w^{(1)}, \ldots, w^{(k)})] - 1, \end{aligned}$$

where the probability is taken over uniformly distributed $x^{(1)}, \ldots, x^{(k)}$.

A.2 Linear representation

We will derive a linear representation for the correlation of linear approximations of addition modulo 2^n. Towards this end, we write the linear approximation with mask vectors $u = (u_{n-1}, \ldots, u_0)$ and $w^{(1)}, \ldots, w^{(k)}$, where $w^{(h)} = (w^{(h)}_{n-1}, \ldots, w^{(h)}_0)$, as a word $z_{n-1} \ldots z_1, z_0$ over the alphabet $\{0, \ldots, 2^{k+1} - 1\}$, where $z_i = u_i 2^k + \sum_{h=1}^{k} w^{(h)}_i 2^{h-1}$. We will then show that there are 2^{k+1} $k \times k$ matrices over rationals, a row vector L and a column vector C such that

$$\mathrm{cor}(u; w^{(1)}, \ldots, w^{(k)}) = L A_{z_{n-1}} \ldots A_{z_1} A_{z_0} C,$$

for all n and all linear approximations $(u; w^{(1)}, \ldots, w^{(k)})$ of addition modulo 2^n of k n-bit integers. We say that the matrices L, A_r, $r = 0, \ldots, 2^{k+1} - 1$, and C form a linear representation of the correlation with dimension k.

For a vector $x \in \mathbf{F}_2^n$ (or integer $x \in \{0, \ldots, 2^n - 1\}$), we let $\mathrm{w_H}(x)$ denote the Hamming weight of x, that is, $\mathrm{w_H}(x)$ is a non-negative integer less than or equal to n, which is the number of non-zero components of x.

Theorem 1. *Let $k > 1$ be a fixed integer. Let L be the row vector of dimension k with all entries equal to 1, and let C be the column vector of dimension k with a single 1 in row 0 and zero otherwise. Let $A_0, \ldots, A_{2^{k+1}-1}$ be the $k \times k$ matrices*

$$\begin{aligned}(A_r)_{d,c} = 2^{-k}(&|\{x \in \mathbf{F}_2^k : u \cdot g(x,c) = x \cdot v, f(x,c) = d\}| - \\ &|\{x \in \mathbf{F}_2^k : u \cdot g(x,c) \neq x \cdot v, f(x,c) = d\}|),\end{aligned}$$

where

$$\begin{aligned}&r = u2^k + \sum_{h=1}^{k} v_h 2^{h-1},\ v = (v_1, \ldots, v_k),\ x = (x_1, \ldots, x_k),\\ &c, d \in \{0, \ldots, k-1\},\\ &f\colon \{0, \ldots, k-1\}^2 \to \{0, \ldots, k-1\}, f(x,c) = \lfloor (w_H(x) + c)/2 \rfloor,\\ &g\colon \{0, \ldots, k-1\}^2 \to \{0,1\}, g(x,c) = (w_H(x) + c) \bmod 2.\end{aligned}$$

Let $n \geq 1$ be an integer and let $(u; w^{(1)}, \ldots, w^{(k)})$ be a linear approximation of addition modulo 2^n with k inputs. Let $z = z_{n-1} \ldots z_1, z_0$ be the word associated with the approximation. We then have

$$\mathrm{cor}(u; w^{(1)}, \ldots, w^{(k)}) = L A_{z_{n-1}} \cdots A_{z_1} A_{z_0} C.$$

Note that the functions f and g are the carry and sum functions for the basic school-book method for adding k integers in binary.

Proof. We denote by $(x^{(1)}, \ldots, x^{(k)})$ the n-bit integers that are added modulo 2^n. We use the simple school-book method. We set the first carry bit $c_0 = 0$. Then the carries c_i and the sum bits s_i at each step $i = 0, \ldots, n-1$ are computed as follows

$$\begin{aligned} s_i &= g((x_i^{(1)}, \ldots, x_i^{(k)}), c_i), \\ c_{i+1} &= f((x_i^{(1)}, \ldots, x_i^{(k)}), c_i) \end{aligned}$$

We set $b_0 = 0$ and, for all $j = 1, \ldots, n$, let

$$b_j = \bigoplus_{i=0}^{j-1} (u_i s_i \oplus w_i^{(1)} x_i^{(1)} \oplus \cdots \oplus w_i^{(k)} x_i^{(k)}).$$

Let $P(z, j)$ be the column vector

$$P(z,j)_c = \mathbf{Pr}[b_j = 0, c_j = c] - \mathbf{Pr}[b_j = 1, c_j = c]$$

for $j = 0, \ldots, n$ and $c = 0, \ldots, k-1$. Let $M(z, i)$ be the $k \times k$ matrix

$$\begin{aligned} M(z,i)_{d,c} = \mathbf{Pr}[(u_i s_i \oplus w_i^{(1)} x_i^{(1)} \oplus \cdots \oplus w_i^{(k)} x_i^{(k)}) = 0 \text{ and } c_{i+1} = d \,|\, c_i = c] - \\ \mathbf{Pr}[(u_i s_i \oplus w_i^{(1)} x_i^{(1)} \oplus \cdots \oplus w_i^{(k)} x_i^{(k)}) = 1 \text{ and } c_{i+1} = d \,|\, c_i = c], \end{aligned}$$

for $i = 0, \ldots, n-1$. Then we have

$$\sum_{c=0}^{k-1} M(z,i)_{d,c} P(z,i)_c = P(z, i+1)_d,$$

and thus

$$P(z, i+1) = M(z,i) P(z,i).$$

Note that

$$\begin{aligned} P(z,0)_0 &= \mathbf{Pr}[b_0 = 0, c_0 = 0] - \mathbf{Pr}[b_0 = 1, c_0 = 0] = 1, \text{ and} \\ P(z,0)_c &= \mathbf{Pr}[b_0 = 0, c_0 = c] - \mathbf{Pr}[b_0 = 1, c_0 = c] = 0, \text{ for } c \neq 0. \end{aligned}$$

At the other end we have

$$\begin{aligned} LP(z,n) &= \sum_{c=0}^{k-1} (\mathbf{Pr}[b_n = 0, c_n = c] - \mathbf{Pr}[b_n = 1, c_n = c]) \\ &= \mathbf{Pr}[b_n = 0] - \mathbf{Pr}[b_n = 1] = \mathrm{cor}(u; w^{(1)}, \ldots, w^{(k)}) \end{aligned}$$

as desired. Since $A_{z_i} = M(z, i)$, it follows that

$$\mathrm{cor}(u; w^{(1)}, \ldots, w^{(k)}) = L A_{z_{n-1}} \cdots A_{z_1} A_{z_0} C.$$

□

The correlation of a linear approximation of addition modulo 2^n with k inputs can thus be computed by doing n multiplications of a $k \times k$ matrix and a column vector, and n additional additions. For a fixed k, this is a linear -time algorithm, and for small k efficient in practice. Note that the number of matrices to be stored in memory is 2^{k+1}. We remark that an analogous method can be used to compute the differential probability of addition modulo 2^n with k inputs.

Using Theorem 1 we get the following matrices for $k = 3$.

$$A_0 = \frac{1}{8}\begin{pmatrix} 4 & 1 & 0 \\ 4 & 6 & 4 \\ 0 & 1 & 4 \end{pmatrix},$$

$$A_1 = A_2 = A_4 = -A_8 = \frac{1}{8}\begin{pmatrix} 2 & 1 & 0 \\ -2 & 0 & 2 \\ 0 & -1 & -2 \end{pmatrix},$$

$$A_3 = A_5 = A_6 = -A_9 = -A_{10} = -A_{12} = \frac{1}{8}\begin{pmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{pmatrix},$$

$$A_7 = -A_{11} = -A_{13} = -A_{14} = \frac{1}{8}\begin{pmatrix} -2 & 1 & 0 \\ 2 & 0 & -2 \\ 0 & -1 & 2 \end{pmatrix}, \text{ and}$$

$$A_{15} = \frac{1}{8}\begin{pmatrix} 4 & -1 & 0 \\ 4 & -6 & 4 \\ 0 & -1 & 4 \end{pmatrix}.$$

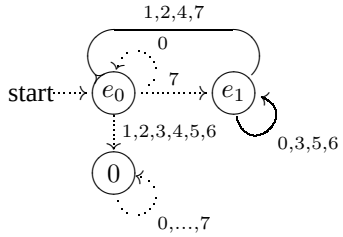
A.3 Searching for masks for a given correlation

In this section, we briefly describe the method used to search for all relevant masks for addition modulo 2^n with two inputs. Using Theorem 1, we get a linear representation L', A'_0, ..., A'_7, C of dimension 2 for the correlation of linear approximations of addition modulo 2^n with two inputs. The matrix A'_0 has the Jordan form $\mathrm{diag}(1, 1/2) = H_2 A'_0 H_2^{-1}$, where $H_2 = \left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$ is the 2×2 Hadamard matrix. We get a new linear representation by making the change of basis $L = R'H_2^{-1}$, $A_i = H_2 A_i H_2^{-1}$ and $C = H_2 C'$. This gives the matrices $L = \begin{pmatrix} 1 & 0 \end{pmatrix}$, $C = \begin{pmatrix} 1 & 1 \end{pmatrix}^t$,

$$A_0 = \frac{1}{2}\begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}, \quad A_1 = A_2 = -A_4 = \frac{1}{2}\begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix},$$
$$A_7 = \frac{1}{2}\begin{pmatrix} 0 & 2 \\ 1 & 0 \end{pmatrix}, \quad -A_3 = A_5 = A_6 = \frac{1}{2}\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

Let $e_0 = \begin{pmatrix} 1 & 0 \end{pmatrix}$ and $e_1 = \begin{pmatrix} 0 & 1 \end{pmatrix}$. Then $e_0 A_0 = e_0$, $e_0 A_7 = e_1$, $e_0 A_i = 0$ for $i \neq 0, 7$, $e_1 A_0 = e_1 A_5 = e_1 A_6 = \frac{1}{2} e_1$, $e_1 A_1 = e_1 A_2 = e_1 A_7 = \frac{1}{2} e_0$, $e_1 A_3 = -\frac{1}{2} e_1$ and $e_1 A_4 = -\frac{1}{2} e_0$. It follows that the computation of $L A_{w_{n-1}} \cdots A_{w_0} C$ by multiplication

from left to right can be described by the following automaton.



When reading w from left to right, if the automaton ends up in state 0, $LA_{w_{n-1}} \cdots A_{w_0} C = 0$. If the automaton ends up in state e_0 or e_1, $LA_{w_{n-1}} \cdots A_{w_0} C = \pm 2^{-k}$, where k is the number of transitions marked by a solid arrow, and the sign is determined by the number of occurrences of $\{3, 4\}$: $LA_{w_{n-1}} \cdots A_{w_0} C > 0$ if and only if the number of occurrences is even. For example, when $w = 73620_8$, we have the state transitions

$$(e_0) \overset{7}{\dashrightarrow} (e_1) \overset{3}{\longrightarrow} (e_1) \overset{6}{\longrightarrow} (e_1) \overset{2}{\longrightarrow} (e_0) \overset{0}{\dashrightarrow} (e_0)$$

and thus $LA_7 A_3 A_6 A_2 A_0 = -2^{-3}$. Clearly, $LA_{w_{n-1}} \cdots A_{w_0} C = 0$ if and only if w matches the regular expression

$$\big(0 + 7(0 + 3 + 5 + 6)^*(1 + 2 + 4 + 7)\big)^*(1 + 2 + 3 + 4 + 5 + 6)\Sigma^* \ ,$$

where $\Sigma = 0 + 1 + \cdots + 7$.

Let $S^0(n, k)$ and $S^1(n, k)$ denote the formal languages

$$S^0(n,k) = \{w \mid |w| = n, e_0 A_{w_{n-1}} \cdots A_{w_0} = \pm 2^{-k} e_0\} \quad \text{and}$$
$$S^1(n,k) = \{w \mid |w| = n, e_0 A_{w_{n-1}} \cdots A_{w_0} = \pm 2^{-k} e_1\}$$

for $n > 0$. Then $S^0(n, k) + \mathcal{S}^1(n, k)$ is the set of words of length $n > 0$ corresponding to linear approximations of addition with two inputs that have correlation $\pm 2^{-k}$. The languages are clearly given recursively by (juxtaposition denotes concatenation, and $+$ denotes union)

$$S^0(n,k) = S^0(n-1,k)0 + S^1(n-1,k-1)(1+2+4+7) \quad \text{and}$$
$$S^1(n,k) = S^0(n-1,k)7 + S^1(n-1,k-1)(0+3+5+6)$$

for all $0 \le k < n$. The base cases are $S^0(1, 0) = 0$ and $S^1(1, 0) = 7$. If $k < 0$ or $k \ge n$, $S^0(n, k) = S^1(n, k) = \emptyset$.

These recursive descriptions immediately give an efficient algorithm for finding all input and output masks for addition with a given correlation. Moreover, one or two of the three masks can optionally be fixed. Using generating functions, it is also straightforward to determine the distribution of the correlation coefficients—that is, count the number of input/output masks with a given correlation. These results where proved using different methods in [10].

Unfortunately, there does not seem to be any simple way to obtain the same results for addition with three inputs, since it seems impossible to obtain an equally simple linear representation with a change of basis.

B Other linear distinguishers—figures and equations

B.1 A three-round linear distinguisher for SNOW 2.0

A three-round linear distinguisher for SNOW 2.0 is depicted in Figure 4 and the corresponding linear approximate relation is given by equation (3).

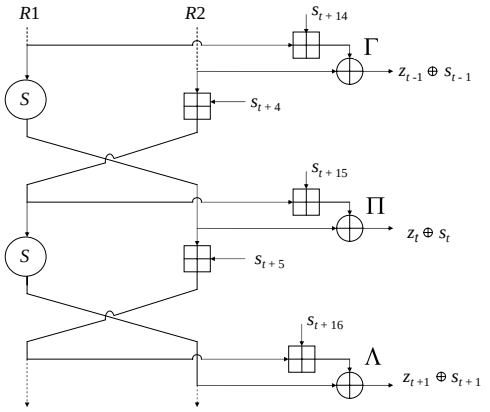


Fig. 4. Linear masking of SNOW 2.0 over three rounds

$$\begin{aligned}
&\Gamma \cdot (z_{t+15} \oplus z_{t+1}) \oplus \Gamma\alpha \cdot z_{t-1} \oplus \Gamma\alpha^{-1} \cdot z_{t+10} \oplus \\
&\Pi \cdot (z_{t+16} \oplus z_{t+2}) \oplus \Pi\alpha \cdot z_t \oplus \Pi\alpha^{-1} \cdot z_{t+11} \oplus \\
&\Lambda \cdot (z_{t+17} \oplus z_{t+3}) \oplus \Lambda\alpha \cdot z_{t+1} \oplus \Lambda\alpha^{-1} \cdot z_{t+12} = 0.
\end{aligned} \tag{3}$$

B.2 A two-and-half-round linear distinguisher for SNOW 3G with non-bijective $S2$

In this distinguisher it is assumed that the linear masking by Λ of the output from $S2$ is approximated by zero, see Figure 5. Then the linear approximate relation of this distinguisher is identical to (1).

B.3 A three-round linear distinguisher for SNOW 3G

The resulting linear approximate relation involving keystream terms z_i only, is the same as (3). This can be seen as follows. Let x denote the input to the first (in time) instance of $S1$, y denote the input to $S2$, and w the input to the second instance of $S1$. In addition to the mask values given in Figure 6 we denote by Δ and Ψ the input and output masks

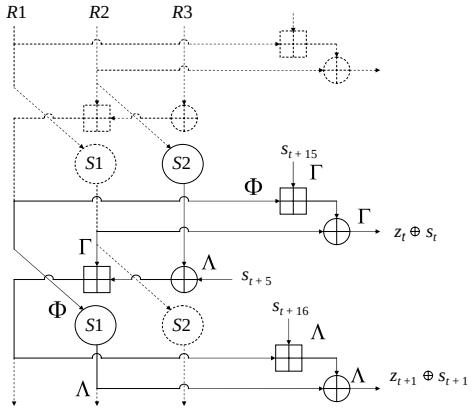


Fig. 5. Linear masking of SNOW 3G with non-bijective $S2$

for the first $S1$, Θ the input mask to the second $S1$, and finally, by Σ_1, Σ_2 and Σ_3 the three masks used to mask s_{t+14}, s_{t+15} and s_{t+16}, respectively. Then we have the following approximate relations:

$$
\begin{aligned}
&\Psi \cdot S1(x) = \Delta \cdot x \\
&\Phi \cdot S2(y) = \Gamma \cdot y \\
&\Lambda \cdot S1(w) = \Theta \cdot w \\
&\Gamma \cdot (s_{t+14} \boxplus x) = \Sigma_1 \cdot s_{t+14} \oplus \Delta \cdot x \\
&\Pi \cdot (s_{t+15} \boxplus w) = \Sigma_2 \cdot s_{t+15} \oplus \Theta \cdot w \\
&\Lambda \cdot (s_{t+16} \boxplus S1(x) \boxplus (s_{t+5} \oplus S2(y))) \\
&= \Sigma_3 \cdot s_{t+16} \oplus (\Psi \oplus \Pi) \cdot S1(x) \oplus \Phi \cdot (s_{t+5} \oplus S2(y)).
\end{aligned}
\tag{4}
$$

The three auxiliary variables x, y and w cancel due to the following three equalities:

$$
\begin{aligned}
&x \boxplus s_{t+14} = y \oplus z_{t-1} \oplus s_{t-1} \\
&z \boxplus s_{t+15} = S1(x) \oplus z_t \oplus s_t \\
&S1(x) \boxplus (S2(y) \oplus s_{t+5})) \boxplus s_{t+16} = S1(w) \oplus z_{t+1} \oplus s_{t+1}
\end{aligned}
$$

Then we have:

$$
\begin{aligned}
&\Gamma \cdot (z_{t-1} \oplus s_{t-1}) \oplus \Pi \cdot (z_t \oplus s_t) \oplus \Lambda \cdot (z_{t+1} \oplus s_{t+1}) \\
&\oplus \Sigma_1 \cdot s_{t+14} \oplus \Sigma_2 \cdot s_{t+15} \oplus \Sigma_3 \cdot s_{t+16} \oplus \Phi \cdot s_{t+5} = 0,
\end{aligned}
$$

or what is the same:

$$\Gamma \cdot z_{t-1} \oplus \Pi \cdot z_t \oplus \Lambda \cdot z_{t+1}$$

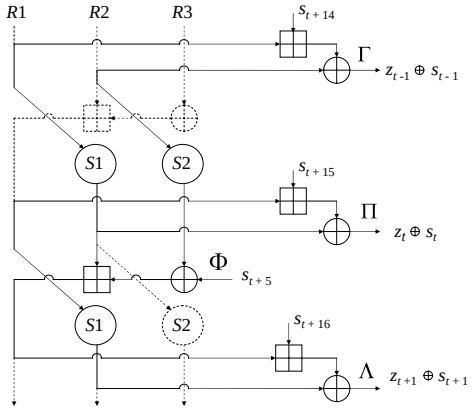


Fig. 6. Linear masking of SNOW 3G

$$= \Gamma \cdot s_{t-1} \oplus \Pi \cdot s_t \oplus \Lambda \cdot s_{t+1} \oplus \Phi \cdot s_{t+5} \oplus \Sigma_1 \cdot s_{t+14} \oplus \Sigma_2 \cdot s_{t+15} \oplus \Sigma_3 \cdot s_{t+16}.$$

This equation is used four times. First two times for $t = t + 2$ and $t = t + 16$. Then once with $t = t$ and with all z_i and s_i variables multiplied by α. Finally, the equation is used for $t = t + 11$ with all z_i and s_i variables multiplied by α^{-1}. Since the $\alpha s_t \oplus s_{t+2} \oplus \alpha^{-1} s_{t+11} \oplus s_{t+16} = 0$, for all t, the s_i variables cancel and we get:

$$\Gamma \cdot z_{t+1} \oplus \Pi \cdot z_{t+2} \oplus \Lambda \cdot z_{t+3} \oplus \Gamma \cdot z_{t+15} \oplus \Pi \cdot z_{t+16} \oplus \Lambda \cdot z_{t+17} \oplus \Gamma \cdot \alpha z_{t-1}$$
$$\oplus \Pi \cdot \alpha z_t \oplus \Lambda \cdot \alpha z_{t+1} \oplus \Gamma \cdot \alpha^{-1} z_{t+10} \oplus \Pi \cdot \alpha^{-1} z_{t+11} \oplus \Lambda \cdot \alpha^{-1} z_{t+12} = 0$$

which is the same as (3). When all s_i variables are multiplied by α the approximations over invidual functions take the following forms:

$$\Psi \cdot S1(x) = \Delta \cdot x$$
$$\Phi\alpha \cdot S2(y) = \Gamma\alpha \cdot y$$
$$\Lambda\alpha \cdot S1(w) = \Theta \cdot w$$
$$\Gamma\alpha \cdot (s_{t+14} \boxplus x) = \Sigma_1\alpha \cdot s_{t+14} \oplus \Delta \cdot x$$
$$\Pi\alpha \cdot (s_{t+15} \boxplus w) = \Sigma_2\alpha \cdot s_{t+15} \oplus \Theta \cdot w$$
$$\Lambda\alpha \cdot (s_{t+16} \boxplus S1(x) \boxplus (s_{t+5} \oplus S2(y)))$$
$$= \Sigma_3\alpha \cdot s_{t+16} \oplus (\Psi\alpha \oplus \Pi\alpha) \cdot S1(x) \oplus \Phi\alpha \cdot (s_{t+5} \oplus S2(y)).$$

To get approximations with multiplication by α^{-1} just replace α by α^{-1}. In both cases the masks Ψ, Δ and Θ can be chosen independently of the masks denoted using the same symbol for approximations (4).