Cryptanalysis of Sosemanuk and SNOW 2.0 Using Linear Masks

Jung-Keun Lee, Dong Hoon Lee, and Sangwoo Park

ETRI Network & Communication Security Division,
909 Jeonmin-dong, Yuseong-gu, Daejeon, Korea

Abstract. In this paper, we present a correlation attack on Sosemanuk with complexity less than 2^{150}. Sosemanuk is a software oriented stream cipher proposed by Berbain et al. to the eSTREAM call for stream cipher and has been selected in the final portfolio. Sosemanuk consists of a linear feedback shift register(LFSR) of ten 32-bit words and a finite state machine(FSM) of two 32-bit words. By combining linear approximation relations regarding the FSM update function, the FSM output function and the keystream output function, it is possible to derive linear approximation relations with correlation $-2^{-21.41}$ involving only the keystream words and the LFSR initial state. Using such linear approximation relations, we mount a correlation attack with complexity $2^{147.88}$ and success probability 99% to recover the initial internal state of 384 bits. We also mount a correlation attack on SNOW 2.0 with complexity $2^{204.38}$.

Keywords: stream cipher, Sosemanuk, SNOW 2.0, correlation attack, linear mask

1 Introduction

Sosemanuk[3] is a software oriented stream cipher proposed by Berbain et al. to the eSTREAM call for stream cipher and has been selected in the final portfolio. The merits of Sosemanuk has been recognized as its considerable security margin and moderate performance[2].

Sosemanuk is based on the stream cipher SNOW 2.0[11] and the block cipher Serpent[1]. Though SNOW 2.0 is a highly reputed stream cipher, it is vulnerable to linear distinguishing attacks using linear masks[14, 15]. To strengthen against linear distinguishing attacks, Sosemanuk applies the multiplication modulo 2^{32} with a bit rotation in the FSM update function and a Serpent S-box in bit slice mode in the keystream output function. As of now, there are no known attacks against Sosemanuk with complexity less than 2^{226}[5].

Linear masking has been used in the linear distinguishing attacks on word-based stream ciphers such as SNOW 1.0[9], SNOW 2.0, NLS[7], and Dragon[8]. Coppersmith et al.[9] presented a linear distinguishing attack on SNOW 1.0. They identified linear approximation relations of large correlation involving only the LFSR states and the keystream words. Then using simple bitwise recurrence relations between the LFSR state words, they were able to mount a linear

distinguishing attack on SNOW 1.0. Watanabe et al.[15] presented a linear distinguishing attack on SNOW 2.0 and then Nyberg and Wallén[14] refined the attack.

On the other hand, Berbain et al.[4] presented a correlation attack on Grain using linear approximation relations between the initial LFSR state and the keystream bits to recover the initial LFSR state. As to solving systems of linear approximation equations, similar technique was used in [6] and iterative decoding technique was used in [12].

In this paper, combining the linear masking method with the techniques in [4] using fast Walsh transform to recover the initial LFSR state of Grain, we mount a correlation attack on Sosemanuk. The time, data and memory complexity are all less than 2^{150}.

This paper is organized as follows. In Sect. 2, we present a description of Sosemanuk. In Sect. 3, we show how to get approximation relations between the initial LFSR state and the keystream words. In Sect. 4, we describe the attack using the approximation relations. In Sect. 5, we present simulation results. In Sect. 6, we present a correlation attack on SNOW 2.0. We conclude in Sect. 7.

2 Preliminaries

2.1 Notations and Definitions

We define the correlation of a function with respect to masks as follows. Let $f : (\mathrm{GF}(2)^n)^k \to \mathrm{GF}(2)^n$ be a function and let $\Gamma_0, \Gamma_1, \ldots, \Gamma_k$ be n-bit masks. Then the correlation of f with respect to the tuple $(\Gamma_0; \Gamma_1, \ldots \Gamma_k)$ of masks is defined as

$$\mathrm{c}_f(\Gamma_0; \Gamma_1, \ldots, \Gamma_k) := 2\,\mathrm{Prob}(\Gamma_0 \cdot f(x_1, \ldots, x_k) = \Gamma_1 \cdot x_1 \oplus \ldots \oplus \Gamma_k \cdot x_k) - 1,$$

where $\cdot$ represents the inner product which will be omitted henceforth. We also define the correlation of an approximation relation as

$$2\,\mathrm{Prob}(\text{the approximation holds}) - 1 \ .$$

The following notations will be used in the following sections.

- $\mathrm{wt}(x)$: the Hamming weight of a binary vector or a 32-bit word x
- $\boxplus$: addition modulo 2^{32}
- $\times$: multiplication modulo 2^{32}
- $[i_1, \ldots, i_m]$: the 32-bit linear mask $2^{i_1} + \ldots + 2^{i_m}$ ($i_1, \ldots, i_m$ are distinct integers in between 0 and 31.)
- $\mathrm{c}_+(\Gamma_0; \Gamma_1, \ldots, \Gamma_m)$: the correlation of $f(x_1, \ldots, x_m) = x_1 \boxplus \ldots \boxplus x_m$ with respect to the tuple $(\Gamma_0; \Gamma_1, \ldots \Gamma_k)$ of 32-bit masks
- $\mathrm{c2}_+(\Gamma) := \mathrm{c}_+(\Gamma; \Gamma, \Gamma)$ for 32-bit linear mask Γ
- $\mathrm{c3}_+(\Gamma) := \mathrm{c}_+(\Gamma; \Gamma, \Gamma, \Gamma)$ for 32-bit linear mask Γ
- $\mathrm{c}_T(\Gamma_0; \Gamma_1)$: the correlation of $\mathrm{Trans}(x)$ with respect to the tuple $(\Gamma_0;\ \Gamma_1)$ of 32-bit masks
- $\mathrm{c2}_T(\Gamma) = \mathrm{c}_T(\Gamma; \Gamma)$ for 32-bit linear mask Γ
- $x_{(j)}$: j-th least significant bit of a nibble, a byte or a 32-bit word x

2.2 Description of Sosemanuk

The structure of Sosemanuk[3] is depicted in Fig. 1. Sosemanuk consists of three main components: a 10-word linear feedback shift register, a 2-word finite state machine, and a nonlinear output function. Sosemanuk is initialized with the key of length in between 128 and 256 and the 128-bit initialization value. The output of the cipher is a sequence of 32-bit keystream words $(z_t)_{t\geq 1}$. The LFSR state at time t is denoted by $LR^t = (s_{t+1}, s_{t+2}, \ldots, s_{t+10})$.($t = 0$ designates the time after initialization.) The LFSR is updated using the recurrence relation

$$s_{t+10} = s_{t+9} \oplus \alpha^{-1} s_{t+3} \oplus \alpha s_t \text{ for all } t \geq 1,$$

where α is a zero of the primitive polynomial

$$P(X) = X^4 + \beta^{23} X^3 + \beta^{245} X^2 + \beta^{48} X + \beta^{239}$$

on $\mathrm{GF}(2^8)(X)$ and $\mathrm{GF}(2^8) = \mathrm{GF}(2)[\gamma]$, where γ is a zero of the primitive polynomial

$$Q(X) = X^8 + X^7 + X^5 + X^3 + 1$$

on $\mathrm{GF}(2)(X)$. The FSM state at time t is denoted by $(R1_t, R2_t)$. The FSM is updated as follows.

$$\begin{aligned} R1_t &= R2_{t-1} \boxplus (s_{t+1} \oplus \mathrm{lsb}(R1_{t-1}) s_{t+8}), \\ R2_t &= \mathrm{Trans}(R1_{t-1}) = (M \times R1_{t-1})^{<<<7}, \end{aligned}$$

where $M =$ `0x54655307`. The FSM has output

$$f_t = (s_{t+9} \boxplus R1_t) \oplus R2_t \ .$$

The keystream words are obtained as follows.

$$(z_{t+3}, z_{t+2}, z_{t+1}, z_t) = \mathit{Serpent}1(f_{t+3}, f_{t+2}, f_{t+1}, f_t) \oplus (s_{t+3}, s_{t+2}, s_{t+1}, s_t)$$
$$(t \equiv 1 (\mathrm{mod}\ 4))$$

where $\mathit{Serpent}1$ denotes the Serpent S-box S_2 applied in bit slice mode. Four words are output per 4 LFSR clockings.

3 Linear Approximations

In this section, we get linear approximation relations involving only the LFSR states and the keystream words with non-negligible correlation by approximating the FSM update functions, the FSM output functions, and the keystream output function using linear masks with non-negligible correlation.

Let $a_t = \mathrm{lsb}(R1_t)$. We consider the following approximations using 32-bit linear masks Γ by replacing all operations (modular additions and the Trans function) by XORs in the FSM update function and the FSM output function:

$$\begin{aligned} &\Gamma R1_{t+1} = \Gamma R2_t \oplus \Gamma(s_{t+2} \oplus a_t s_{t+9}), \\ &\Gamma R2_{t+1} = \Gamma R1_t, \\ &\Gamma f_t = \Gamma s_{t+9} \oplus \Gamma R1_t \oplus \Gamma R2_t, \\ &\Gamma f_{t+1} = \Gamma s_{t+10} \oplus \Gamma R1_{t+1} \oplus \Gamma R2_{t+1} \ . \end{aligned}$$

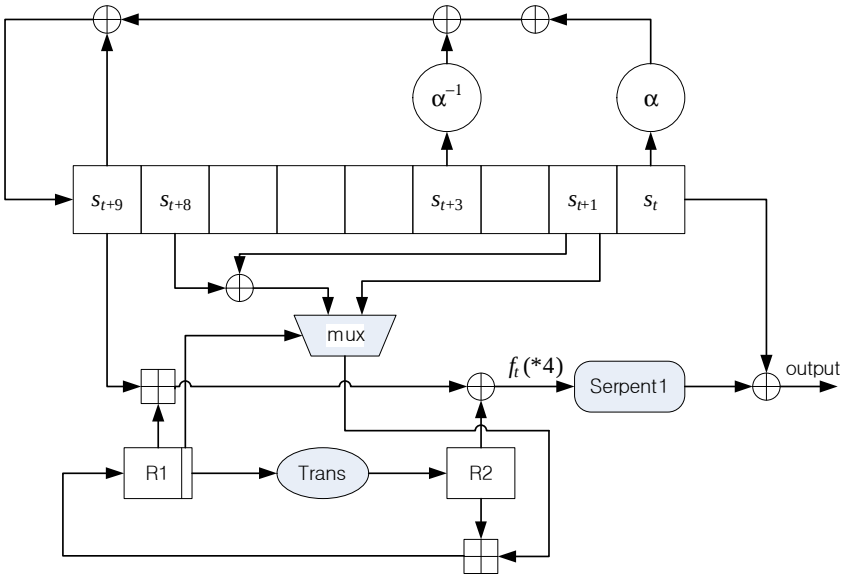

Fig. 1. The Structure of Sosemanuk

XORing the above relations and applying the Piling-Up Lemma, we have the approximation

$$\Gamma(f_t \oplus f_{t+1}) = \Gamma s_{t+2} \oplus a_t \Gamma s_{t+9} \oplus \Gamma s_{t+9} \oplus \Gamma s_{t+10} \tag{1}$$

with correlation $\mathrm{c2}_+(\Gamma)^3 \mathrm{c2}_T(\Gamma)$ *assuming that the four linear approximations are independent.*

However the way of computing the correlation as above is not accurate since the approximation relations have high dependencies. For example, approximations of two modular additions with correlations c_1, c_2 do not necessarily yield an approximation with correlation $c_1 c_2$. So we need to consider approximation relations which do not have obvious dependencies. We have the following equations regarding the internal states and keystream words:

$$\begin{aligned} f_t \oplus R2_t &= s_{t+9} \boxplus \mathrm{Trans}^{-1}(R2_{t+1}), \\ f_{t+1} \oplus R2_{t+1} &= s_{t+10} \boxplus (R2_t \boxplus (s_{t+2} \oplus a_t s_{t+9})) \ . \end{aligned}$$

We consider the following associated approximation relations

$$\begin{aligned} \Gamma f_t \oplus \Gamma R2_t &= \Gamma s_{t+9} \oplus \Gamma R2_{t+1}, \\ \Lambda f_{t+1} \oplus \Lambda R2_{t+1} &= \Lambda s_{t+10} \oplus \Lambda R2_t \oplus \Lambda s_{t+2} \oplus a_t \Lambda s_{t+9} \ . \end{aligned}$$

where Γ and Λ are linear masks as depicted in Fig. 2. The correlations of the above approximations are

$$\sum_{\Phi} \mathrm{c}_+(\Gamma; \Gamma, \Phi) \mathrm{c}_T(\Gamma; \Phi)$$

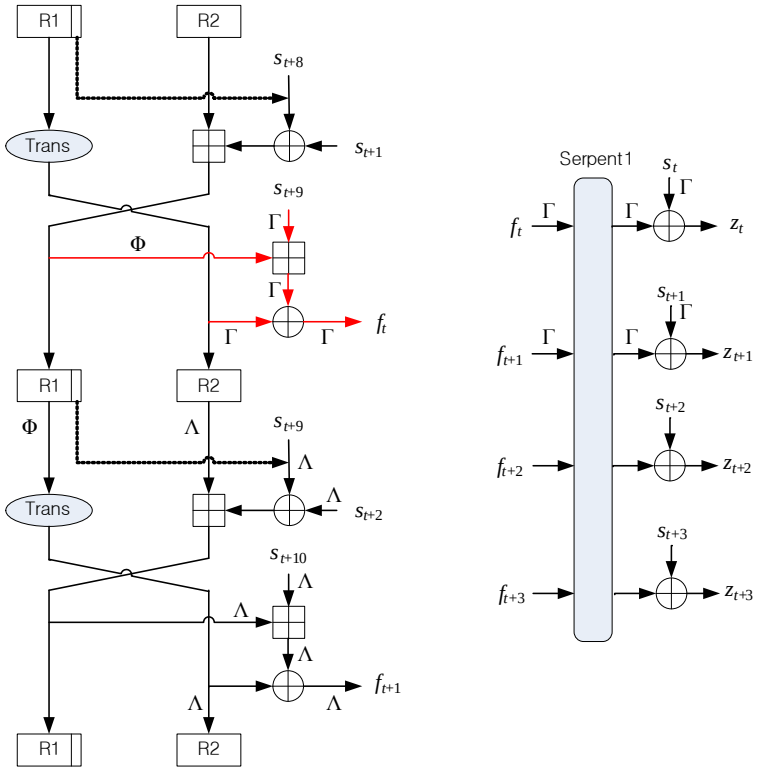

Fig. 2. Some Linear Masking of Sosemanuk

and $\mathrm{c3}_+(\Lambda)$, respectively.

Note that the first correlation is a composite correlation of the function $x_2 = \mathrm{Trans}^{-1}(x_3)$ and the function $y = x_1 \boxplus x_2$ with respect to $(\Gamma; \Gamma, \Gamma)$ which can be computed as a sum of partial correlations [13, Theorem 3][14]. So if we let $\Gamma = \Lambda$, we have the same approximation relation (1) with correlation

$$\mathrm{c3}_+(\Gamma) \sum_{\Phi} \mathrm{c}_+(\Gamma; \Gamma, \Phi)\mathrm{c}_T(\Gamma; \Phi) \ .$$

In order to remove terms involving f_t and f_{t+1} in (1), we will utilize a linear approximation relation regarding the keystream output function that comes from the third S-box S_2 of the block cipher Serpent in bit slice mode.

```
unsigned char S2[16] = {8,6,7,9,3,12,10,15,13,1,14,4,0,11,5,2}
```

S_2 has maximal linear correlation $\frac{1}{2}$. Regarding the function $y = S_2(x)$, we have 8 linear approximation relations with maximal correlation $\frac{1}{2}$ which is of the form

$$x_{(i)} + x_{(i+1)} + (\text{terms involving only } y) = 0 \ .$$

Each of such approximation relations gives linear approximation relations regarding the keystream output function. We will use the relation

$$x_{(0)} + x_{(1)} + y_{(0)} + y_{(3)} = 0$$

which induces the following relation for any $j = 0, \ldots, 31$,

$$(f_t)_{(j)} \oplus (f_{t+1})_{(j)} \oplus (z_t)_{(j)} \oplus (s_t)_{(j)} \oplus (z_{t+3})_{(j)} \oplus (s_{t+3})_{(j)} = 0,$$

with correlation $\frac{1}{2}$ when $t \equiv 1 (\mathrm{mod}\ 4)$. Thus if Γ is a linear mask, then

$$\Gamma(f_t \oplus f_{t+1}) \oplus \Gamma z_t \oplus \Gamma s_t \oplus \Gamma z_{t+3} \oplus \Gamma s_{t+3} = 0, \tag{2}$$

holds with correlation $(\frac{1}{2})^{\mathrm{wt}(\Gamma)}$ when $t \equiv 1 (\mathrm{mod}\ 4)$. Noting that

$$a_t \Gamma s_{t+9} \oplus \Gamma s_{t+9} = 0$$

holds with correlation $\frac{1}{2}$, we have linear approximation (3) involving only LFSR states and keystream words by XORing relations (1) and (2)

$$\Gamma s_t \oplus \Gamma s_{t+2} \oplus \Gamma s_{t+3} \oplus \Gamma s_{t+10} = \Gamma z_t \oplus \Gamma z_{t+3} \tag{3}$$

with correlation

$$C(\Gamma) := (\frac{1}{2})^{\mathrm{wt}(\Gamma)+1} \mathrm{c3}_+(\Gamma) \sum_{\Phi} \mathrm{c}_+(\Gamma; \Gamma, \Phi) \mathrm{c}_T(\Gamma; \Phi)$$

when $t \equiv 1 (\mathrm{mod}\ 4)$, assuming that the approximations are independent. Note that we don't see obvious dependencies between the approximations given above. We check the validity of our estimation by simulations described in Sect. 5.

3.1 Search for Linear Masks

We try to find Γ such that $|C(\Gamma)|$ is as large as possible. Taking into consideration the factor $(\frac{1}{2})^{\mathrm{wt}(\Gamma)}$, we confined the search to masks of weight less than or equal to 5. Furthermore, we have the following observation from many examples though we don't have a proof:

- If $\mathrm{c2}_T(\Gamma) = 0$, then $C(\Gamma) = 0$.

Based on this observation, we compute $C(\Gamma)$ for a given mask Γ in the following way:
If $\mathrm{c2}_T(\Gamma) \neq 0$, then

1. we compute $\mathrm{c3}_+(\Gamma)$ using [14, Theorem 1] regarding correlation of modular addition.
2. We compute $\sum_{\Phi} \mathrm{c}_+(\Gamma; \Gamma, \Phi) \mathrm{c}_T(\Gamma; \Phi)$ using [14, Theorem 1] and fast Walsh transform. Once Γ is fixed, we can compute $\mathrm{c}_+(\Gamma; \Gamma, \Phi)$ for any Φ using the description with finite automaton in [14]. It turns out that for each Γ, $\mathrm{c}_+(\Gamma; \Gamma, \Phi) = 0$ except for most Φ's. Using fast Walsh transform, for each fixed Γ, we can compute $\mathrm{c}_T(\Gamma; \Phi)$ for all Φ with time complexity 2^{37} and memory complexity 2^{32}.

Table 1. Correlations with respect to some linear masks of weight 4

Γ	$\log_2(\lvert c3_+(\Gamma)\rvert)$	$\log_2(\lvert\Sigma_\Phi\rvert)$	$-(\mathrm{wt}(\Gamma)+1)$	$\lvert C(\Gamma)\rvert$
$[25, 14, 13, 0]$	-3.17	-14.33	-5	$2^{-22.50}$
$[25, 24, 14, 0]$	-3.17	-13.24	-5	$2^{-21.41}$
$[25, 22, 18, 0]$	-4.55	-15.13	-5	$2^{-24.68}$

Then we obtain the following results:

- There does not exist a mask Γ of weight 1,2, or 3 such that $|C(\Gamma)| > 2^{-29}$.
- The only masks Γ of weight 2 such that $C(\Gamma) \neq 0$ are $[i, i+25]$ $(i = 0, \ldots, 6)$.
- There exist masks Γ of weight 4 such that $|C(\Gamma)| > 2^{-25}$. Some of them are listed in Table 1.

We also considered some masks Γ of the form $[i, i+25, j, k, l]$, but we could not find one such that $|C(\Gamma)| > 2^{-25}$. Thus the best linear mask we found out is $[25, 24, 14, 0]$, for which the correlation is $-2^{-21.41}$.

4 Correlation Attack on Sosemanuk

In this section, we describe a correlation attack against Sosemanuk recovering the initial internal state. Using the approximation relations (3) involving only LFSR state words and keystream words with non-negligible correlation obtained in the preceding section, we apply the techniques in [4] using fast Walsh transform to mount the attack.

Getting Approximation Relations between Initial LFSR State and Keystream Words. Let Γ be the linear mask $[25, 24, 14, 0]$, $\kappa = C(\Gamma) = -2^{-21.41}$, and $\epsilon = |\kappa/2| = 2^{-22.41}$ throughout this section. Starting with the approximation (3) with correlation κ, we can obtain arbitrarily many linear approximations with correlation κ involving the initial LFSR state $s_1, \cdots, s_{10}$ and the keystream words using the relation

$$\begin{aligned}&(\Gamma_0, \Gamma_1, \cdots, \Gamma_9) \cdot (s_{t+j}, s_{t+j+1}, \cdots, s_{t+j+9})\\ &= (\mathcal{G}^j(\Gamma_0, \Gamma_1, \cdots, \Gamma_9)) \cdot (s_t, s_{t+1}, \cdots, s_{t+9})\end{aligned}$$

for each $j > 0$, where $\mathcal{G}$ is the "dual" of the LFSR update transformation and is given by

$$\begin{aligned}&\mathcal{G}(\Gamma_0, \Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4, \Gamma_5, \Gamma_6, \Gamma_7, \Gamma_8, \Gamma_9)\\ &\quad = (\alpha^*\Gamma_9, \Gamma_0, \Gamma_1, \Gamma_2 \oplus (\alpha^{-1})^*\Gamma_9, \Gamma_3, \Gamma_4, \Gamma_5, \Gamma_6, \Gamma_7, \Gamma_8 \oplus \Gamma_9),\end{aligned}$$

where $\alpha^*\Gamma$ and $(\alpha^{-1})^*\Gamma$ are 32-bit linear masks such that $(\alpha^*\Gamma)(x) = \Gamma(\alpha x)$ and $((\alpha^{-1})^*\Gamma)(x) = \Gamma(\alpha^{-1}x)$ for each 32-bit x.

To be more explicit, the approximation relations (3) can be rewritten as

$$\begin{array}{l}(\Gamma \oplus \alpha^*\Gamma, 0, \Gamma, \Gamma \oplus (\alpha^{-1})^*\Gamma, 0, 0, 0, 0, 0, \Gamma) \cdot (s_1, \cdots, s_{10}) = \Gamma z_1 \oplus \Gamma z_4 \\ (\Gamma \oplus \alpha^*\Gamma, 0, \Gamma, \Gamma \oplus (\alpha^{-1})^*\Gamma, 0, 0, 0, 0, 0, \Gamma) \cdot (s_5, \cdots, s_{14}) = \Gamma z_5 \oplus \Gamma z_8 \\ (\Gamma \oplus \alpha^*\Gamma, 0, \Gamma, \Gamma \oplus (\alpha^{-1})^*\Gamma, 0, 0, 0, 0, 0, \Gamma) \cdot (s_9, \cdots, s_{18}) = \Gamma z_9 \oplus \Gamma z_{12} \\ \cdots,\end{array} \tag{4}$$

which are again equivalent to

$$\begin{array}{l}(\Gamma \oplus \alpha^*\Gamma, 0, \Gamma, \Gamma \oplus (\alpha^{-1})^*\Gamma, 0, 0, 0, 0, 0, \Gamma) \cdot (s_1, \cdots, s_{10}) = \Gamma z_1 \oplus \Gamma z_4 \\ \mathcal{F}(\Gamma \oplus \alpha^*\Gamma, 0, \Gamma, \Gamma \oplus (\alpha^{-1})^*\Gamma, 0, 0, 0, 0, 0, \Gamma) \cdot (s_1, \cdots, s_{10}) = \Gamma z_5 \oplus \Gamma z_8 \\ \mathcal{F}^2(\Gamma \oplus \alpha^*\Gamma, 0, \Gamma, \Gamma \oplus (\alpha^{-1})^*\Gamma, 0, 0, 0, 0, 0, \Gamma) \cdot (s_1, \cdots, s_{10}) = \Gamma z_9 \oplus \Gamma z_{12} \\ \cdots,\end{array} \tag{5}$$

where $\mathcal{F} = \mathcal{G}^4$. Thus the complexity of getting R relations between the initial LFSR state and the keystream words is comparable to the complexity of getting $128R$ bits of keystream.

Recovering Part of the Initial LFSR State. We apply the "Second LFSR Derivation Technique" in [4]. Let $n = 320$ be the size of the LFSR state in bits and $m < n$. Let $\epsilon' = 2\epsilon^2 = 2^{-43.82}$ and $N = (\frac{2\lambda}{3\epsilon'})^2$, where λ satisfies

$$\frac{1}{\sqrt{2\pi}} \int_{\lambda}^{\infty} e^{-\frac{t^2}{2}}\, dt = 2^{-m}.$$

Let $R = \sqrt{N 2^{n-m+1}}$. Let $u_1, \cdots, u_n$ be the bits of the LFSR initial state $s_1, \cdots, s_{10}$. Suppose we have R linear approximation relations of correlation κ involving u_i's. Let $i_1, \cdots, i_m$ be any integers such that $1 \le i_1 < \ldots < i_m \le n$. XORing pairs of those R equations, we get about $R(R-1)2^{m-n-1} \approx N$ approximation relations with correlation $2\epsilon'$ involving only $u_{i_1}, \ldots, u_{i_m}$ among u_i's. Let these relations be

$$a^j_{i_1} u_{i_1} + \cdots + a^j_{i_m} u_{i_m} = b^j. \ (j = 1, \ldots, N) \tag{6}$$

Let us define the function $\sigma : \mathrm{GF}(2)^m \to \mathbb{Z}$ by

$$\begin{array}{rl}\sigma(a_1, \cdots, a_m) = & |\{j \in \{1, \ldots, N\} : (a^j_{i_1}, \ldots, a^j_{i_m}) = (a_1, \ldots, a_m),\ b^j = 0\}| \\ & - |\{j \in \{1, \ldots, N\} : (a^j_{i_1}, \ldots, a^j_{i_m}) = (a_1, \ldots, a_m),\ b^j = 1\}|\end{array}$$

Let W be the fast Walsh transform defined by

$$W(f)(y_1, \ldots, y_m) = \sum_{x_1, \ldots, x_m \in \mathrm{GF}(2)} f(x_1, \ldots, x_m)(-1)^{y_1 x_1 + \ldots + y_m x_m}$$

for $f : \mathrm{GF}(2)^m \to \mathbb{Z}$. Note that, for each $(u_{i_1}, \cdots, u_{i_m})$, $W(\sigma)(u_{i_1}, \cdots, u_{i_m})$ is

$$\begin{array}{l}\text{the number of relations in (6) satisfied by} (u_{i_1}, \cdots, u_{i_m}) \\ - \text{ the number of relations in (6) not satisfied by} (u_{i_1}, \cdots, u_{i_m}).\end{array} \tag{7}$$

Table 2. Complexity of the Attack

	with Precomputation			without Precomputation		
	time(unit)	memory(bit)	data(bit)	time(unit)	memory(bit)	data(bit)
Precomputation	$2^{147.47}$	$2^{148.34}$				
Online computation	$2^{144.66}$	$2^{144.55}$	$2^{145.50}$	$2^{147.88}$	$2^{147.10}$	$2^{145.50}$

For the right value of $(u_{i_1}, \cdots, u_{i_m})$, above number follows the normal distribution $\mathrm{N}(2N\epsilon', N(1-4\epsilon'^2))$. So, using $N(1-4\epsilon'^2) \approx N$, for the right value of $(u_{i_1}, \cdots, u_{i_m})$,

$$\mathrm{Prob}\left(W(\sigma)(u_{i_1}, \cdots, u_{i_m}) < \frac{3}{2}N\epsilon'\right) = \frac{1}{\sqrt{2\pi}} \int_{\frac{\lambda}{3}}^{\infty} e^{-\frac{t^2}{2}}\, dt \ .$$

But for random $(u_{i_1}, \cdots, u_{i_m})$, (7) follows the distribution $\mathrm{N}(0, N)$. So for random $(u_{i_1}, \cdots, u_{i_m})$,

$$\mathrm{Prob}\left(W(\sigma)(u_{i_1}, \cdots, u_{i_m}) > \frac{3}{2}N\epsilon'\right) = \frac{1}{\sqrt{2\pi}} \int_{\lambda}^{\infty} e^{-\frac{t^2}{2}}\, dt = 2^{-m} \ .$$

Thus, when we use the threshold value $\frac{3}{2}N\epsilon'$ for determining whether a partial LFSR state candidate $(u_{i_1}, \cdots, u_{i_m})$ is the right one, we have non-detection probability less than $\frac{1}{\sqrt{2\pi}} \int_{\frac{\lambda}{3}}^{\infty} e^{-\frac{t^2}{2}}\, dt$ and false alarm rate 2^{-m}.

Complexity of the Attack. The attack can be performed in two ways. One way is to precompute the coefficients $(a^j_{i_1}, \cdots, a^j_{i_m})$ and then perform all other computations in online phase. The other is to perform all the computations online. Complexity of both ways are described below and summarized in Table 2.

Attack with Precomputation. To recover partial bits $u_{i_1}, \cdots, u_{i_m}$ of the initial initial state, in the precomputation phase, we get the coefficients of the left hand sides of the R approximation relations (5) between the LFSR initial states and the keystream words. Store the $(320 + \lceil \log_2(R) \rceil)$-bit values (U_i, i) $(i = 0, \cdots, R-1)$ in a list, where

$$U_i := \mathcal{F}^i(\Gamma \oplus \alpha^* \Gamma, 0, \Gamma, \Gamma \oplus (\alpha^{-1})^* \Gamma, 0, 0, 0, 0, 0, \Gamma)$$

for each i. Then sort the list according to the components in $\{1, \cdots, m\} - \{i_1, \cdots, i_m\}$. For each pair (i, k) such that the components of U_i and U_k in $\{1, \cdots, m\} - \{i_1, \cdots, i_m\}$ coincides, compute $X_{i,k} :=$ ($U_i \oplus U_k$ restricted to i_1-th, $\cdots$, i_m-th components), and store $(X_{i,k}, i, k)$ in a list. The list has about N entries of size $m + 2\lceil \log_2(R) \rceil$. In the online phase, set the function $\sigma : \mathrm{GF}(2)^m \rightarrow$

Table 3. Complexity of basic operations

operations	time complexity
XOR of two k-bit words	k
Comparison of two k-bit words	k
Sorting a list with r k-bit entries	$kr\log_2(r)$
Walsh transform for 2^m k-bit integers	$km2^m$

$\mathbb{Z}$ as zero. Let $w_i = \Gamma z_{4i+1} + \Gamma z_{4i+4}$ for each $i = 0, \cdots, R-1$. For each $(X_{i,k}, i, k)$ in the list, compute the value $w_i + w_k$ and update σ. (The update rule is that $\sigma(X_{i,k})$ increases by 1 if $w_i + w_k = 0$ and decreases by 1 otherwise.) Perform the fast Walsh transform to σ and check if there is some $(u_{i_1}, \cdots, u_{i_m})$ such that $W(\sigma)(u_{i_1}, \cdots, u_{i_m}) > \frac{3}{2}N\epsilon'$. The complexity of the above attack to recover m bits of the initial LFSR state is as follows. The complexity of the above attack to recover m bits of the initial LFSR state is as follows. We assume the complexity of the basic operations as in Table 3. The precomputation phase has time complexity of about $128R + R\log_2(R)(320 + \lceil\log_2(R)\rceil) + (N+R)(320 + \lceil\log_2(R)\rceil)$ and memory requirement of $R(320 + \lceil\log_2(R)\rceil) + N(m + 2\lceil\log_2(R)\rceil)$ bits if we apply a sorting algorithm of small memory requirement. The online phase takes $2^m\lceil\log_2(N)\rceil$-bits of memory and time complexity of $8N + m2^m\lceil\log_2(N)\rceil$. The data complexity of the online phase is 2^7R bits. Let $m = 138$. Then $\lambda \approx 13.6$(by e.g. Lemma 1 in the Appendix), $N = 2^{94.00}$ and $R = 2^{138.50}$. For recovery of the whole n bits of the LFSR initial state, we recover $(u_1, \cdots, u_m)$ and $(u_m, \cdots, u_{2m-1})$ using above-mentioned methods. Then restore the remaining 45 bits of the initial LFSR state and 64 initial FSM bits simultaneously using exhaustive search. The precomputation phase takes time complexity of $128R + 2(R\log_2(R)(320 + \lceil\log_2(R)\rceil) + (N+R)(320 + \lceil\log_2(R)\rceil)) = 2^{155.47}$. (The number in the table is $2^{147.47}$ regarding 1 time unit as the time needed to generate 256 bits of keystream which is not greater than the time cost of one trial in the exhaustive search .)The required memory is $2R(320 + \lceil\log_2(R)\rceil) + N(m + 2\lceil\log_2(R)\rceil) = 2^{148.34}$ bits. The online phase has time complexity of $2(8N + m2^m\lceil\log_2(N)\rceil) = 2^{152.66}$, memory requirement of $2^m\lceil\log_2(N)\rceil = 2^{144.55}$ bits, and data complexity of $2^7R = 2^{145.50}$ bits. The non-detection probability is less than $\frac{2}{\sqrt{2\pi}}\int_{\frac{\lambda}{3}}^{\infty} e^{-\frac{t^2}{2}}\,dt \leq 0.01$. We mention that the increased complexity due to sorting was not considered in [4].

Attack without Precomputation. To recover partial bits $u_{i_1}, \cdots, u_{i_m}$ of the initial LFSR state, we first get all the coefficients of the R approximation relations using the keystreams. Store the $(320+1)$-bit values (U_i, w_i) $(i = 0, \cdots, R-1)$. Then sort the list according to the components in $\{1, \cdots, m\} - \{i_1, \cdots, i_m\}$. Set the function σ as zero. For each pair (i, k) such that the components of U_i and U_k in $\{1, \cdots, m\} - \{i_1, \cdots, i_m\}$ coincides, compute $X_{i,k}$ and update the function σ using $(X_{i,k}, w_i + w_k)$. Perform the fast Walsh transform to σ and check if there is some $(u_{i_1}, \cdots, u_{i_m})$ such that $W(\sigma)(u_{i_1}, \cdots, u_{i_m}) > \frac{3}{2}N\epsilon'$. The

time complexity is about $128R + R\log_2(R)(n+1) + N(n+1) + m2^m\lceil\log_2(N)\rceil$ and memory requirement is about $\lceil\log_2(N)\rceil 2^m + (320+1)R$ bits. The data complexity is 2^7R bits. Let $m = 138$. For recovery of the whole n bits of the LFSR initial state, we recover $(u_1, \cdots, u_m)$ and $(u_m, \cdots, u_{2m-1})$ using above-mentioned methods. Then restore the remaining 45 bits of the initial LFSR state and 64 initial FSM bits simultaneously using exhaustive search. The time complexity is $2(128R+R\log_2(R)(n+1)+N(n+1)+m2^m\lceil\log_2(N)\rceil)+129\cdot 2^{129} = 2^{155.88}$. The memory requirement is $\lceil\log_2(N)\rceil 2^m + (320+1)R = 2^{147.10}$ bits, and the data complexity is $2^7R = 2^{145.50}$ bits.

Improving the Attack. We can reduce the data complexity without increasing the time complexity. For the Serpent S-box S_2, we have 8 linear approximations with correlation $\frac{1}{2}$ which is of the form

$$x_{(i)} + x_{(i+1)} + (\text{terms involving only } y) = 0 \ .$$

Using these approximations, we can get 8 linear approximation relations involving the LFSR initial state and keystream words with correlation κ. Thus we can reduce the data complexity at least by the factor of 2^3. We can also reduce the memory requirement of the attack using the "Improved Hybrid Method"[4] without increasing time complexity or data complexity much.

5 Simulations and Results

5.1 Simulations for a Reduced Cipher

We validate our claims by simulating a reduced version of Sosemanuk keystream generator defined as follows. It consists of an LFSR of five bytes and an FSM of two bytes. The LFSR state at time t is $(s_t, s_{t+1}, \ldots, s_{t+5})$. The LFSR is updated using the relation

$$s_{t+5} = s_{t+4} \oplus \beta^{-1} s_{t+3} \oplus \beta s_t,$$

where β is a zero of $x^8 + x^7 + x^5 + x^3 + 1$ in

$$GF(2^8) = GF(2)(\beta) = GF(2)[x]/ < x^8 + x^7 + x^5 + x^3 + 1 >$$

The FSM state at time t is denoted by $(R1_t, R2_t)$. The FSM is updated as follows.

$$R1_t = R2_{t-1} + (s_{t+1} \oplus \text{lsb}(R1_{t-1})s_{t+3})(\text{mod } 2^8)$$
$$R2_t = \text{Trans}(R1_{t-1}) = ((M \times R1_{t-1})(\text{mod } 2^8))^{<<<3}$$

where, $M =$ `0x59`. The FSM has output

$$f_t = (s_{t+4} + R1_t)(\text{mod } 2^8) \oplus R2_t.$$

The keystream bytes are obtained as follows.

$$(z_{t+3}, z_{t+2}, z_{t+1}, z_t) = Serpent1(f_{t+3}, f_{t+2}, f_{t+1}, f_t) \oplus (s_{t+3}, s_{t+2}, s_{t+1}, s_t)$$
$$(t \equiv 1(\text{mod } 4))$$

Table 4. Correlations with respect to linear masks of weight 2

Γ	$\log_2(\lvert c3_+(\Gamma)\rvert)$	$\log_2(\lvert\Sigma_\Phi\rvert)$	$-(\mathrm{wt}(\Gamma)+1)$	correlation
$[5,0]$	-1.59	-1.91	-3	$-2^{-6.50}$
$[6,1]$	-10	-3	-3	-2^{-16}
$[7,2]$	-3.57	-3.36	-3	$-2^{-9.93}$

Then we get a linear approximation relation

$$\Gamma s_t \oplus \Gamma s_{t+2} \oplus \Gamma s_{t+3} \oplus \Gamma s_{t+5} = \Gamma z_t \oplus \Gamma z_{t+3} \ \ (t \equiv 1 (\mathrm{mod}\ 4))$$

with correlation

$$(\frac{1}{2})^{\mathrm{wt}(\Gamma)+1} \mathrm{c3}_+(\Gamma) \sum_{\Phi} \mathrm{c}_+(\Gamma;\Gamma,\Phi)\mathrm{c}_T(\Gamma;\Phi)$$

when $t \equiv 1 (\mathrm{mod}\ 4)$, for each 8-bit mask Γ. In the simulation, we generate 2^{30} bytes of keystream and observe the actual correlation of the linear approximation regarding the LFSR states and the keystream bytes for various initial internal states. The observed actual correlation is about $-2^{-6.12}$ when $\Gamma = [5,0]$ and about $-2^{-10.31}$ when $\Gamma = [7,2]$ regardless of the initial internal state. Using the observed correlation for $\Gamma = [5,0]$, we are able to recover the initial internal state using the method explained in Sect. 4. The parameters are $n = 40$, $m = 24$, $\lambda = 2.83$, $N = 2^{30.31}$ and $R = 2^{23.66}$. We get R approximation relations regarding the n-bit initial LFSR state and the keystream words. Then we get about N approximations regarding the latter m bits of the initial LFSR state. Applying the fast Walsh transform to an array with 2^m entries, we can recover the m bits correctly most of the time. We performed the experiments to recover the latter 24 bits of the initial LFSR state for 100 initial initial internal states as follows.

- LFSR initial states: $(i, i+1, i+2, i+3, i+4)$ $(i = 0, \cdots, 99)$
- FSM initial state: (0,0) (fixed)

With the threshold $\frac{3}{2}N2^{-13.24} = 206382$, we were able to get the right 24-bit value in each case except when $i = 26$. In each case 0–4 false alarms occurred with average 1.18. A few minutes was spent on a Pentium IV 3.4GHz CPU with 1GB RAM for each case. This experimental results corroborate our assertions.

5.2 Simulations with Long Keystreams for Full Sosemanuk

To check if the correlation of relations (3) is correct in another way, we generate long keystreams for Sosemanuk for some initial internal states. We consider the following 2 LFSR initial states and 8 FSM initial states.

- LFSR initial states

Table 5. Simulation Result for Long Keystreams

LFSR	FSM	z-value	correlation*	LFSR	FSM	z-value	correlation*
A	0	-0.29	$-2^{-21.28}$	B	0	1.93	$-2^{-22.89}$
	1	-2.13	$-2^{-20.64}$		1	-0.65	$-2^{-21.13}$
	2	0.69	$-2^{-21.79}$		2	-0.15	$-2^{-21.34}$
	3	0.35	$-2^{-21.59}$		3	1.09	$-2^{-22.06}$
	4	0.54	$-2^{-21.70}$		4	-0.95	$-2^{-21.02}$
	5	-0.35	$-2^{-21.25}$		5	-0.16	$-2^{-21.34}$
	6	-0.48	$-2^{-21.20}$		6	0.99	$-2^{-21.99}$
	7	-0.62	$-2^{-21.14}$		7	1.73	$-2^{-22.64}$

*:observed correlation

  - A: (`0x9000, 0x8000,` $\cdots$ `, 0x1000, 0x0000`)
  - B: (`0x9111, 0x8000,` $\cdots$ `, 0x1000, 0x0111`) (the same as A except for the first and the last word)
- FSM initial states: (`0x0000, 0x0000`), $\cdots$, (`0x7000, 0x7000`)

For each of the 16 initial states, we generate Sosemanuk keystreams of 2^{53} bits and count how many of the 2^{46} induced relations (3) are satisfied for the mask $\Gamma = [25, 24, 14, 0]$ and compute the observed correlation. The results are as in Table 5. In the table, "z-value" represents

$$\frac{(\text{the number of the satisfied among the } 2^{46} \text{ relations}) - (2^{45} + 2^{45}C(\Gamma))}{2^{22}},$$

which is the normalized deviation in the assumed normal distribution. In total, the observed correlation using the 2^{50} relations is $-2^{-21.45}$, which is very close to $C(\Gamma)$. This result also corroborates our assertions.

6 Correlation Attack on SNOW 2.0

SNOW 2.0[11] consists of an LFSR consisting of 16 words and an FSM of 2 words. In [14], it was shown that there exists a linear approximation relation of the LFSR bits and keystream bits with bias $2^{-15.496}$ or correlation $\pm 2^{-14.496}$[14, Table 2]. One of such approximation relations is

$$\Lambda s_t + \Lambda s_{t+1} + \Lambda s_{t+5} + \Lambda s_{t+15} + \Lambda s_{t+16} = \Lambda z_t + \Lambda z_{t+1},$$

where $\Lambda = [0, 15, 16]$. Applying the "Second LFSR derivation technique" again with parameter $n = 512$ and $\epsilon = 2^{-15.496}$, we can mount a correlation attack on SNOW 2.0 without precomputation as follows.

Let $m = 192$. Using the same notation as in Sect. 4, $\lambda \approx 16.1$, $N = 2^{66.54}$ and $R = 2^{193.77}$. The time complexity of the attack for recovering m bits is $32R + R\log_2(R)(n+1) + N(n+1) + m2^m\lceil\log_2(N)\rceil$. (The factor 32 comes

from the fact that 32 bits of keystreams are needed per one approximation relation.) Memory requirement is about $\lceil \log_2(N) \rceil 2^m + (512+1)R$ bits. The data complexity is $2^5 R$ bits. For recovery of the whole initial LFSR state, recover partial 192 bits of LFSR three times and then recover the initial FSM state by exhaustive search. The total time complexity is $3(32R + R\log_2(R)(n+1) + N(n+1) + m2^m\lceil \log_2(N) \rceil) = 2^{212.38}$. The memory complexity is about $\lceil \log_2(N) \rceil 2^m + (512+1)R = 2^{202.83}$ bits. The data complexity is $2^5 R = 2^{198.77}$ bits. Since the initialization of SNOW 2.0 is a reversible process, we can recover the key from the initial state.

7 Conclusion

We described an attack recovering the initial internal state with time complexity $2^{147.88}$, memory complexity $2^{147.10}$ bits, and data complexity $2^{145.50}$ bits. Though the attack does not threaten the claimed 128-bit security of Sosemanuk, it indicates that using keys longer than 150 bits for Sosemanuk does not guarantee the security level of the key size. The main reason Sosemanuk is vulnerable to the attack described in this paper is that the LFSR state is too small in the presence of a relatively large correlation between the LFSR state and the keystream words. Similar attack of complexity $2^{204.38}$ is valid against SNOW 2.0.

References

1. R. Anderson, E. Biham, and L. Knudsen. Serpent: A Proposal for the Advanced Encryption Standard. Available from `http://www.cl.cam.ac.uk/~rja14/serpent.html`.
2. S. Babbage et al. The eSTREAM Portfolio. Available from `http://www.ecrypt.eu.org/stream/portfolio.pdf`, April 15,2008.
3. C. Berbain, O. Billet, A. Canteaut, N. Courtois, H. Gilbert, L. Goubin, A. Gouget, L. Granboulan, C. Lauradoux, M. Minier, T. Pornin, and H. Sibert. SOSEMANUK, a fast software-oriented stream cipher. eSTREAM Report 2005/027 (2005)
4. C. Berbain, H. Gilbert, A. Maximov. Cryptanalysis of Grain. In *Fast Software Encryption*(FSE 2006), LNCS 4047, pp. 15–29, Springer-Verlag, 2006.
5. D. Bernstein. Which eSTREAM ciphers have been broken? eSTREAM Report 2008/010 (2008)
6. V. Chepyzhov, T. Johansson, and B. Smeets. A Simple Algorithm for Fast Correlation Attacks on Stream Ciphers. In *Fast Software Encryption*(FSE 2000), LNCS 1978, pp. 181–195, Springer-Verlag, 2001.
7. J. Cho and J. Pieprzyk. Crossword Puzzle Attack on NLS. In *Selected Areas in Cryptography*(SAC 2006), LNCS 4356, pp. 249–265, Springer-Verlag, 2007.
8. J. Cho, *An Improved Estimate of the Correlation of Distinguisher for Dragon*, In Workshop Record of *The State of the Art of Stream Ciphers*(SASC 2008), pp. 11-20.
9. D. Coppersmith, S. Halevi, and C. Jutla. Cryptanalysis of stream ciphers with linear masking. In *Advances in Cryptology - Crypto 2002*, LNCS 2442, pp. 515–532, Springer-Verlag, 2002.

10. P. Ekdahl and T. Johansson. SNOW - a new stream cipher. Available from `http://www.it.ith.se/cryptology/snow/`.
11. P. Ekdahl and T. Johansson. A new version of the stream cipher SNOW. In *Selected Areas in Cryptography*(SAC 2002), LNCS 1233, pp. 37–46, Springer-Verlag, 2002.
12. J. Golic, V. Bagini, and G. Morgari. Linear Cryptanalysis of Bluetooth Stream Cipher. In *Advances in Cryptology - Eurocrypt 2002*, LNCS 2332, pp. 238–255, Springer-Verlag, 2002.
13. K. Nyberg. Correlation theorems in cryptanalysis. *Discrete Applied Mathematics*, pp. 177–188, Volume 111, 2001.
14. K. Nyberg and J. Wallén. Improved Linear Distinguishers for SNOW 2.0. In *Fast Software Encryption*(FSE 2006), LNCS 4047, pp. 144–162, Springer-Verlag, 2006.
15. D. Watanabe, A. Biryukov, and C. De Canniere. A Distinguishing Attack of SNOW 2.0 with Linear Masking Method. In *Selected Areas in Cryptography*(SAC 2003), LNCS 3006, pp. 222–233, Springer-Verlag, 2004.

A An Approximation of the Cumulative Normal Distribution Function

Lemma 1. *For any* $0 < a < 1$*, we have*

$$\frac{a}{\lambda}e^{-\frac{\lambda^2}{2}} \leq \int_{\lambda}^{\infty} e^{-\frac{t^2}{2}}\, dt \leq \frac{1}{\lambda}e^{-\frac{\lambda^2}{2}}$$

for any $\lambda \geq 1$ *such that* $a \leq \frac{\lambda^2}{\lambda^2+1}$.

Proof. Let

$$F(x) = \int_{x}^{\infty} e^{-\frac{t^2}{2}}\, dt - \frac{1}{x}e^{-\frac{x^2}{2}} \quad (x > 0) \ .$$

Then $F'(x) = \frac{1}{x}e^{-\frac{x^2}{2}} > 0$ and $\lim_{x\to\infty} F(x) = 0$. Hence $F(x) < 0$ for all $x > 0$. Let

$$G(x) = \int_{x}^{\infty} e^{-\frac{t^2}{2}}\, dt - \frac{a}{x}e^{-\frac{x^2}{2}} \quad (x > 0) \ .$$

Then $G'(x) = (a-1)e^{-\frac{x^2}{2}} + \frac{a}{x^2}e^{-\frac{x^2}{2}}$ so that $G'(x) < 0$ if $a < \frac{x^2}{x^2+1}$.
Since $\lim_{x\to\infty} G(x) = 0$, $G(x) > 0$ when $a < \frac{x^2}{x^2+1}$. □