A Correlation Attack on Full SNOW-V and SNOW-Vi

Zhen Shi, Chenhui Jin, Jiyan Zhang, Ting Cui, and Lin Ding

Information Engineering University, Zhengzhou 450000, China
shizhenieu@126.com, jinchenhui@126.com

Abstract. In this paper, a method for searching correlations between the binary stream of LFSR and the keystream of SNOW-V and SNOW-Vi is presented based on the techniques of composite function. With the aid of the linear relationship between the four taps of LFSR inputting to FSM at three consecutive clocks, we present an automatic search model based on the SAT/SMT technique and search out a binary linear approximation with a correlation $2^{-49.54}$. Applying such approximation, we provide a correlation attack on SNOW-V with an expected time complexity $2^{248.81}$, a memory complexity 2^{240} and 2^{240} keystream words generated by the same key and IV. For SNOW-Vi, we provide a binary linear approximation with the same correlation and mount a correlation attack with the same complexity as that of SNOW-V. The results indicate that neither of SNOW-V and SNOW-Vi can guarantee the 256-bit security level if the design constraint that the maximum of keystream length for a single pair of key and IV is less than 2^{64} is ignored.

Key words: SNOW-V; SNOW-Vi; Cryptanalysis, Linear Approximation; Automatic Search.

1 Introduction

SNOW-V is a new member of the SNOW family stream ciphers following SNOW 1.0, SNOW 2.0 and SNOW 3G. Proposed by Ekdahl and Johansson in 2000 and 2002 respectively, SNOW 1.0 [5] and SNOW 2.0 [6] consist of two main components: the LFSR (linear feedback shift register) part and the FSM (finite state machine) part. SNOW 3G [17] is a word-oriented stream cipher used as the core of 3GPP (3G Partnership Project) Confidentiality and Integrity Algorithms UEA2 & UIA2 for UMTS and LTE. As an improved version of SNOW 2.0, SNOW 3G introduces the third 32-bit register in the FSM and a second 32-bit nonlinear transformation to update the register, thus enhances the security under the 128-bit key. With the development of communication and computing technology, there is an urgent need for increasing the security level to 256-bit key length in 5G application environment. In 2018, SNOW-V [7] was announced to satisfy the 256-bit security level requirement for 5G from 3GPP. Lately, Ekdahl et al. proposed SNOW-Vi [8], which is a new member of SNOW series stream cipher. Compared with SNOW 3G, the structure of SNOW-V and SNOW-Vi

keeps the same, while a couple of LFSRs are used to replace the original one, and the size of registers increases from 32 bits to 128 bits so that the size of internal state raises significantly. This makes SNOW-V and SNOW-Vi difficult to be analyzed.

Resistance against the correlation attack [12, 3] is a standard design criterion for LFSR-based stream ciphers. The basic idea for correlation attack is to approximate the nonlinear parts with linear expressions, and derive statistical characteristics related to keystream words at different clock instances and the LFSR parts, and then recover the initial states of LFSR. Fast correlation attacks on SNOW series stream cipher have always been a hot topic of cryptanalysis. For SNOW 1.0, Coppersmith et al. found a linear approximation with the correlation $2^{-8.3}$ [4] and proposed a fast correlation attack with a complexity 2^{100}. In [18], a distinguishing attack on SNOW 2.0 with complexity around 2^{230} was presented, and then an enhanced version with a reduced complexity 2^{174} was provided in [13]. By approximating the functions with the form of $(x+y)\oplus x\oplus f(y)$, Gong et al. searched out a binary approximation with a correlation $2^{-33.82}$ and mounted a fast correlation attack on SNOW 3G with a complexity $2^{232.33}$ [10]. Later, they took the idea to the extreme, found a binary linear approximation with a correlation $2^{-18.67}$ of SNOW-V$_{\sigma 0}$ [11], which results in a fast correlation attack with time complexity $2^{251.93}$, memory complexity 2^{244} and $2^{103.83}$ keystream words. On the other hand, Zhang et al. gave a fast correlation attack on SNOW 2.0 by building a byte-wise approximation of the FSM part in an assigned finite field [21]. Adopting this idea, Yang et al. found a byte-wise approximation in $GF(2^8)$ with SEI (Squared Euclidean Imbalance) being $2^{-40.97}$ [19], and mounted a fast correlation attack with a complexity $2^{176.56}$. For full SNOW-V and SNOW-Vi, there is no prior correlation attack beyond their design documents.

In this paper, we focus on the search of binary linear approximations with high correlation of SNOW-V and SNOW-Vi. As the FSM transformation is mainly composed of S-boxes, linear transformations and additions modulo 2^{32}, we simplify the linear approximation of FSM transformation to approximation trails of a composite function so that we can take advantage of automatic search technique which allows a wide range of search. Meanwhile, in the previous fast correlation attacks, masks of outputs and LFSR taps are usually the same values, in this paper we consider a different and more general case that the masks of outputs and LFSR taps are different, which can further expand the search range. With the aid of the linear relationship between the four taps of LFSR inputting into FSM at three consecutive clocks, we present an automatic search model based on SAT/SMT technique and search out a binary linear approximation with a correlation $2^{-49.54}$, which results in a correlation attack with an expected time complexity $2^{248.81}$, a memory complexity 2^{240}, requiring 2^{240} keystream words, which can recover the internal state of SNOW-V for the clock producing the first keystream word. For SNOW-Vi, we provide a linear approximation with the same correlation and mount a correlation attack with the same complexity as on SNOW-V. The results of this paper show that SNOW-V and SNOW-Vi can be attacked with complexity less than the key exhaustion, if the

limitation that at most 2^{64} keystream words can be generated by a single key-IV pair is ignored. As far as we know, these results are the best correlation attacks for SNOW-V and SNOW-Vi up to now. It needs to emphasize that our cryptanalysis results don't means that SNOW-V and SNOW-Vi are not safe under their design limit for the maximum length of keystream with a single pair of key and IV.

The rest of this paper is organized as follows: Section 2 lists some notations and gives a brief introduction of SNOW-V and SNOW-Vi. Section 3 proposes the framework of our linear approximation of SNOW-V. Section 4 describes the automatic search models used in this paper in detail. Section 5 and 6 show the correlation attacks on SNOW-V and SNOW-Vi respectively. We conclude this paper in Section 7.

2 Preliminaries

2.1 Notations and definitions

Henceforth, we fix some notations for convenience.

- The binary field is denoted by $GF(2)$, and its n-dimensional extension field is denoted by $GF(2^n)$.
- The bitwise XOR is denoted by $\oplus$, and the addition and minus modulo 2^{32} is denoted by $\boxplus$ and $\boxminus$ respectively.
- Given a binary variable x, $\bar{x}$ denotes $x \oplus 1$.
- $wt(x)$ denotes the Hamming weight of Boolean vector x.
- We denote binary vectors in this paper as $x = (x_{n-1}, x_{n-2}, ..., x_0)$, in which x_{n-1} is the most significant bit and x_0 is the least significant bit.
- Given two binary vectors $a = (a_{n-1}, a_{n-2}, ..., a_0), b = (b_{n-1}, b_{n-2}, ..., b_0)$, the cascading operation is defined as $a||b = (a_{n-1}, a_{n-2}, ..., a_0, b_{n-1}, b_{n-2}, ..., b_0)$.
- Given two binary vectors $a = (a_{n-1}, a_{n-2}, ..., a_0), b = (b_{n-1}, b_{n-2}, ..., b_0)$, the inner product is defined as $a \cdot b = \bigoplus_{i=0}^{n-1} a_i b_i$.
- Let x be an element of $GF(2^k)$ and $y = (y_{m-1}, y_{m-2}, ..., y_0)$ be an m-dimensional vector on the same field, the product of x and y is defined as $x * y = (xy_{m-1}, xy_{m-2}, ..., xy_0)$, in which the product xy_i is taken over $GF(2^k)$.
- The correlation of a binary random variable x is defined as

$$\rho(x) = \Pr(x = 0) - \Pr(x = 1).$$

- The correlation of a Boolean function $f : F_{2^n} \to F_2$ is defined as

$$\rho(f) = \Pr(f(x) = 0) - \Pr(f(x) = 1).$$

- The correlation of $f(x, y) = x \boxplus y$ with the 128-bit input mask α, β and 128-bit output mask γ is denoted by $\rho_A(\gamma \leftarrow \alpha, \beta)$.

– The correlation of $f(x) = AES^R(x, 0)$ with the 128-bit input mask α and 128-bit output mask β is denoted by $\rho_E(\beta \leftarrow \alpha)$.
– We use the corresponding bold letter to denote the matrix of a linear transformation, e.g. $P(x) = \mathbf{P}x$ for a linear transformation P and a column vector x.

2.2 Description of SNOW-V and SNOW-Vi

SNOW-V

SNOW-V has greatly expanded the internal state of the original structure of SNOW 2.0 and SNOW 3G. The LFSR part of SNOW-V is a circular structure consisting of two LFSRs, and the size of each register in FSM part increases to 128 bits. The overall schematic of SNOW-V algorithm is shown in Fig.1.

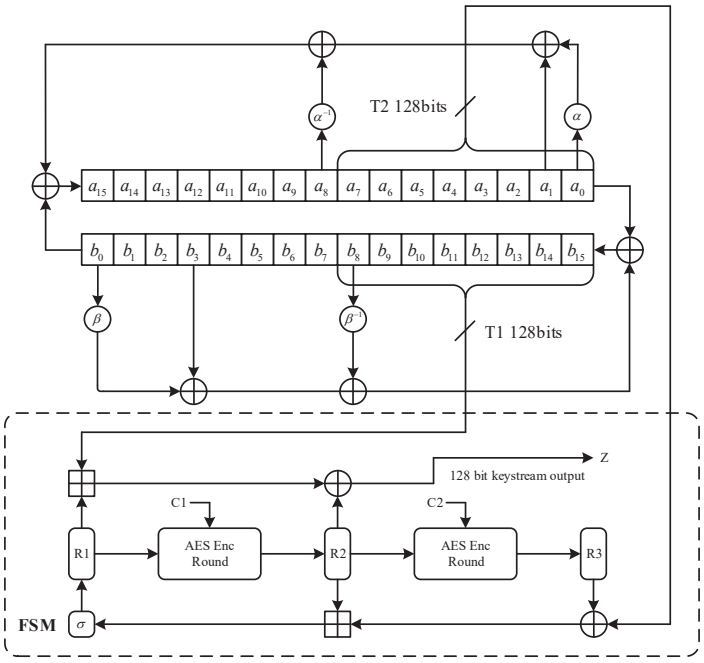

Fig. 1. The keystream generation phase of the SNOW-V stream cipher

Each cell in LFSR part represents an element in $GF(2^{16})$. The elements of LFSR-A are generated by the polynomial

$$g^A(x) = x^{16} + x^{15} + x^{12} + x^{11} + x^8 + x^3 + x^2 + x + 1 \in GF(2)[x],$$

while the elements of LFSR-B are generated by

$$g^B(x) = x^{16} + x^{15} + x^{14} + x^{11} + x^8 + x^6 + x^5 + x + 1 \in GF(2)[x].$$

The LFSR part is updated by

$$a^{(t+16)} = b^{(t)} + \alpha a^{(t)} + a^{(t+1)} + \alpha^{-1} a^{(t+8)} \bmod g^A(\alpha),$$
$$b^{(t+16)} = a^{(t)} + \beta b^{(t)} + b^{(t+3)} + \beta^{-1} b^{(t+8)} \bmod g^B(\beta),$$

in which α is a root of $g^A(x)$ and β is a root of $g^B(x)$. Two taps T_1 and T_2 at time t are respectively given as

$$T_1^{(t)} = (b_{15}^{(8t)}, b_{14}^{(8t)}, ..., b_8^{(8t)}), T_2^{(t)} = (a_7^{(8t)}, a_6^{(8t)}, ..., a_0^{(8t)}).$$

R_1, R_2, R_3 are three 128-bit registers of FSM part, updated by

$$R_1^{(t+1)} = \sigma(R_2^{(t)} \boxplus (R_3^{(t)} \oplus T_2^{(t)})),$$
$$R_2^{(t+1)} = AES^R(R_1^{(t)}, C1),$$
$$R_3^{(t+1)} = AES^R(R_2^{(t)}, C2).$$

$AES^R(input, key)$ denotes the AES encryption round function, $C1$ and $C2$ are zero. σ is a byte-oriented permutation:

$$\sigma = [0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15],$$

and $\sigma^{-1} = \sigma$. The 128 bits keystream output at clock t is given by:

$$z^{(t)} = (R_1^{(t)} \boxplus T_1^{(t)}) \oplus R_2^{(t)}.$$

For more details of SNOW-V, please refer to [7].

SNOW-Vi

SNOW-Vi is an extreme performance variant of SNOW-V, and eliminates the linear relationship between ${T_1}^{(t-1)}, {T_1}^{(t)}, T_1^{(t+1)}, T_2^{(t)}$. SNOW-Vi is consistent with SNOW-V except replacing the polynomial $g^A(x)$ and $g^B(x)$ with

$$g^A(x) = x^{16} + x^{14} + x^{11} + x^9 + x^6 + x^5 + x^3 + x^2 + 1 \in F_2[x](0x4a6d),$$
$$g^B(x) = x^{16} + x^{15} + x^{14} + x^{11} + x^{10} + x^7 + x^2 + x + 1 \in F_2[x](0xcc87),$$

and the LFSR sequences generating expressions with

$$a^{(t+16)} = b^{(t)} + \alpha a^{(t)} + a^{(t+7)} \bmod g^A(\alpha),$$
$$b^{(t+16)} = a^{(t)} + \beta b^{(t)} + b^{(t+8)} \bmod g^B(\beta).$$

Two taps T_1 and T_2 at time t are given respectively as

$$T_1^{(t)} = (b_{15}^{(8t)}, b_{14}^{(8t)}, \cdots, b_8^{(8t)}), T_2^{(t)} = (a_{15}^{(8t)}, a_{14}^{(8t)}, \cdots, a_8^{(8t)}).$$

For more details of SNOW-Vi, we refer to [8].

3 Linear approximation of SNOW-V

SNOW-V employs two LFSRs making up a circular structure. There is a straightforward observation [20] that the four taps at three consecutive clocks satisfy

$$T_1^{(t+1)} = T_2^{(t)} \oplus \beta * {T_1}^{(t-1)} \oplus \beta^{-1} * {T_1}^{(t)} \oplus ({T_1}^{(t-1)} >> 48) \oplus ({T_1}^{(t)} << 80),$$

we also confirm it experimentally. For convenience, we denote it by

$$L(T_1^{(t-1)}, T_1^{(t)}) = T_1^{(t+1)} \oplus T_2^{(t)}.$$

From now, we omit the superscript of $R_1^{(t)}, R_2^{(t)}, R_3^{(t)}$, and simply them as R_1, R_2, R_3. The keystream outputs in three consecutive clocks can be expressed by

$$\begin{aligned} z_{t-1} &= (T_1^{(t-1)} \boxplus E^{-1}(R_2)) \oplus E^{-1}(R_3), \\ z_t &= (T_1^{(t)} \boxplus R_1) \oplus R_2, \\ z_{t+1} &= (T_1^{(t+1)} \boxplus \sigma(R_2 \boxplus (R_3 \oplus T_2^{(t)}))) \oplus E(R_1). \end{aligned}$$

Let $\alpha, \beta, \gamma, l, m, n, h$ be 128-bit masks, we can straightforward observe that the following equation will show a nonzero correlation ρ when the masks take certain values:

$$\begin{aligned} & (\alpha, \beta, \gamma, l, m, n, h) \cdot (z_{t-1}, z_t, z_{t+1}, T_1^{(t-1)}, T_1^{(t)}, T_1^{(t+1)}, T_2^{(t)}) \\ = {} & \alpha \cdot (E^{-1}(R_2) \boxplus T_1^{(t-1)}) \oplus \beta \cdot R_2 \oplus \gamma \cdot (\sigma(R_2 \boxplus (R_3 \oplus T_2^{(t)})) \boxplus T_1^{(t+1)}) \\ & \oplus \alpha \cdot E^{-1}(R_3) \oplus \beta \cdot (R_1 \boxplus T_1^{(t)}) \oplus \gamma \cdot E(R_1) \\ & \oplus l \cdot T_1^{(t-1)} \oplus m \cdot T_1^{(t)} \oplus n \cdot T_1^{(t+1)} \oplus h \cdot T_2^{(t)} \\ \stackrel{\rho}{=} {} & 0. \end{aligned}$$

In order to simplify the linear approximation analysis, we divide the process of the approximation into several sub-steps by introducing 6 functions

$$\begin{aligned} f_1(x, y, z, u, v, w) &= (x \boxminus v, y, z, u, L(z, u) \oplus v, w), \\ f_2(x, y, z, u, v, w) &= ((\sigma^{-1}(x) \boxminus y) \oplus v, y, z, u, v, w), \\ f_3(x, y, z, u, v, w) &= (E^{-1}(x), E^{-1}(y), z, u, v, w), \\ f_4(x, y, z, u, v, w) &= (x, (y \boxplus z), u, v, w), \\ f_5(x, y, z, u, v) &= (x, y, z, u, E^{-1}(v)), \\ f_6(x, y, z, u, v) &= (x, y, u, (z \boxplus v)). \end{aligned}$$

It is clear that the composition function

$$F(x, y, z, u, v, w) := (f_6 \circ f_5 \circ f_4 \circ f_3 \circ f_2 \circ f_1)(x, y, z, u, v, w)$$

has 6-word input and 4-word output.

Consider the binary linear approximation of F:

$$(\gamma, \beta, l, m, n, \gamma) \xrightarrow{F} (\alpha, \alpha, h, \beta).$$

Let

$$(x, y, z, u, v, w) = (\sigma(R_2 \boxplus (R_3 \oplus T_2^{(t)})) \boxplus T_1^{(t+1)}, R_2, T_1^{(t-1)}, T_1^{(t)}, T_1^{(t+1)}, E(R_1)).$$

Then x, y, z, u, v, w are uniform distributed words and independent with each other. Recall $L(T_1^{(t-1)}, T_1^{(t)}) = T_1^{(t+1)} \oplus T_2^{(t)}$, we have

$$f_1(x, y, z, u, v, w) = (\sigma(R_2 \boxplus (R_3 \oplus T_2^{(t)})), R_2, T_1^{(t-1)}, T_1^{(t)}, T_2^{(t)}, E(R_1)),$$

$$f_2 \circ f_1(x, y, z, u, v, w) = (R_3, R_2, T_1^{(t-1)}, T_1^{(t)}, T_2^{(t)}, E(R_1)),$$

$$f_3 \circ f_2 \circ f_1(x, y, z, u, v, w) = (E^{-1}(R_3), E^{-1}(R_2), T_1^{(t-1)}, T_1^{(t)}, T_2^{(t)}, E(R_1)),$$

$$f_4 \circ f_3 \circ f_2 \circ f_1(x, y, z, u, v, w) = (E^{-1}(R_3), E^{-1}(R_2) \boxplus T_1^{(t-1)}, T_1^{(t)}, T_2^{(t)}, E(R_1)),$$

$$f_5 \circ f_4 \circ f_3 \circ f_2 \circ f_1(x, y, z, u, v, w) = (E^{-1}(R_3), E^{-1}(R_2) \boxplus T_1^{(t-1)}, T_1^{(t)}, T_2^{(t)}, R_1),$$

$$f_6 \circ f_5 \circ f_4 \circ f_3 \circ f_2 \circ f_1(x, y, z, u, v, w) = (E^{-1}(R_3), E^{-1}(R_2) \boxplus T_1^{(t-1)}, T_2^{(t)}, T_1^{(t)} \boxplus R_1).$$

Thus, the equation of the linear approximation $(\gamma, \beta, l, m, n, \gamma) \xrightarrow{F} (\alpha, \alpha, h, \beta)$ is

$$\begin{aligned}&(\gamma, \beta, l, m, n, \gamma) \cdot (\sigma(R_2 \boxplus (R_3 \oplus T_2^{(t)})) \boxplus T_1^{(t+1)}, R_2, T_1^{(t-1)}, T_1^{(t)}, T_1^{(t+1)}, E(R_1)) \\ \oplus\ &(\alpha, \alpha, h, \beta) \cdot (E^{-1}(R_3), E^{-1}(R_2) \boxplus T_1^{(t-1)}, T_2^{(t)}, T_1^{(t)} \boxplus R_1) \overset{\rho_F}{=} 0.\end{aligned}$$

It is obvious that the correlation ρ_F of the above approximation is equal to ρ. If $\rho \neq 0$, we will get a distinguisher for distinguishing attack when

$$l \cdot T_1^{(t-1)} \oplus m \cdot T_1^{(t)} \oplus n \cdot T_1^{(t+1)} \oplus h \cdot T_2^{(t)} = 0,$$

and a distinguisher for correlation attack when

$$l \cdot T_1^{(t-1)} \oplus m \cdot T_1^{(t)} \oplus n \cdot T_1^{(t+1)} \oplus h \cdot T_2^{(t)} \neq 0.$$

In this paper, we focus on the search of correlation attack distinguishers.

Now we analyze the masks in the process of the linear approximation on F above in the case that the input and output masks are fixed as $(\gamma, \beta, l, m, n, \gamma)$ and $(\alpha, \alpha, h, \beta)$ respectively. We denote the mask of the j-th output of f_i as ξ_j^i and the correlation of f_i as ρ_i. Then the linear approximation equation of f_1 is

$$\begin{aligned}&\gamma \cdot x \oplus \beta \cdot y \oplus l \cdot z \oplus m \cdot u \oplus n \cdot v \oplus \gamma \cdot w, \\ \overset{\rho_1}{=}\ &\xi_1^1 \cdot (x \boxminus v) \oplus \xi_2^1 \cdot y \oplus \xi_3^1 \cdot z \oplus \xi_4^1 \cdot u \oplus \xi_5^1 \cdot L(z, u) \oplus \xi_5^1 \cdot v \oplus \xi_6^1 \cdot w,\end{aligned} \tag{1}$$

which is equivalent to

$$\begin{aligned}&(\beta \oplus \xi_2^1) \cdot y \oplus (\gamma \oplus \xi_6^1) \cdot w \oplus [\xi_5^1 \cdot L(z, u) \oplus (l \oplus \xi_3^1) \cdot z \oplus (\xi_4^1 \oplus m) \cdot u] \\ \oplus\ &[\xi_1^1 \cdot (x \boxminus v) \oplus \gamma \cdot x \oplus (n \oplus \xi_5^1) \cdot v] \overset{\rho_1}{=} 0.\end{aligned}$$

With the assumption that $\rho_1 \neq 0$, we have $\xi_2^1 = \beta, \xi_6^1 = \gamma$. Denoting $\xi_1^1 = a, \xi_3^1 = e, \xi_4^1 = f, \xi_5^1 = d$, we have $d\mathbf{L} = (e \oplus l)||(f \oplus m)$ by

$$d \cdot L(z, u) = d\mathbf{L} \begin{pmatrix} z \\ u \end{pmatrix} = (e \oplus l)z \oplus (f \oplus m)u,$$

and (1) is equivalent to $\gamma \cdot x \overset{\rho_1}{=} a \cdot (x \boxminus v) \oplus (n \oplus d) \cdot v$, which is the linear approximation $\gamma \leftarrow a, n \oplus d$ of the addition modulo 2^{32}. Thus the correlation of (1) is $\rho_1 = \rho_A(\gamma \leftarrow a, n \oplus d)$.

For f_2, we have

$$\begin{aligned} & a \cdot x \oplus \beta \cdot y \oplus e \cdot z \oplus f \cdot u \oplus d \cdot v \oplus \gamma \cdot w \\ = & \xi_1^2 \cdot (\sigma^{-1}(x) \boxminus y) \oplus \xi_1^2 \cdot v \oplus \xi_2^2 \cdot y \oplus \xi_3^2 \cdot z \oplus \xi_4^2 \cdot u \oplus \xi_5^2 \cdot v \oplus \xi_6^2 \cdot w, \end{aligned} \tag{2}$$

scilicet

$$\begin{aligned} & [\xi_1^2 \cdot (\sigma^{-1}(x) \boxminus y) \oplus a \cdot x \oplus (\beta \oplus \xi_2^2) \cdot y] \oplus (e \oplus \xi_3^2) \cdot z \oplus (\xi_4^2 \oplus f) \cdot u \\ \oplus & (\xi_1^2 \oplus \xi_5^2 \oplus d) \cdot v \oplus (\xi_6^2 \oplus \gamma) \cdot w \overset{\rho_2}{=} 0. \end{aligned}$$

By $\rho_2 \neq 0$ we know that $\xi_3^2 = e, \xi_4^2 = f, \xi_6^2 = \gamma, \xi_1^2 = d \oplus \xi_5^2$. Denoting $\xi_2^2 = b$, then (2) is equivalent to $\xi_1^2 \cdot (\boldsymbol{\sigma}^{-1}(x) \boxminus y) \oplus a \cdot x \oplus (\beta \oplus \xi_2^2) \cdot y = 0$. Let $X = \boldsymbol{\sigma}^{-1}(x)$, then the above equation can be converted to

$$a \cdot \sigma(X) = (\boldsymbol{\sigma}^T a) X \overset{\rho_2}{=} (\beta \oplus b) \cdot y \oplus \xi_1^2 \cdot (X \boxminus y),$$

which is the linear approximation $\boldsymbol{\sigma}^T a \leftarrow \beta \oplus b, d \oplus \xi_5^2$ of the addition modulo 2^{32}, hence $\rho_2 = \rho_A(\boldsymbol{\sigma}^T a \leftarrow \beta \oplus b, d \oplus \xi_5^2)$.

For f_3, the following equation holds

$$\begin{aligned} & (d \oplus \xi_5^2) \cdot x \oplus b \cdot y \oplus e \cdot z \oplus f \cdot u \oplus \xi_5^2 \cdot v \oplus \gamma \cdot w \\ \overset{\rho_3}{=} & \xi_1^3 \cdot E^{-1}(x) \oplus \xi_2^3 \cdot E^{-1}(y) \oplus \xi_3^3 \cdot z \oplus \xi_4^3 \cdot u \oplus \xi_5^3 \cdot v \oplus \xi_6^3 \cdot w. \end{aligned} \tag{3}$$

It is equivalent to

$$\begin{aligned} & [\xi_1^3 \cdot E^{-1}(x) \oplus (d \oplus \xi_5^2) \cdot x] \oplus [\xi_2^3 \cdot E^{-1}(y) \oplus b \cdot y] \oplus (\xi_3^3 \oplus e) \cdot z \oplus (\xi_4^3 \oplus f) \cdot u \\ \oplus & (\xi_5^3 \oplus \xi_5^2) \cdot v \oplus (\xi_6^3 \oplus \gamma) \cdot w \overset{\rho_3}{=} 0. \end{aligned}$$

By $\rho_3 \neq 0$ we know that $\xi_3^3 = e, \xi_4^3 = f, \xi_5^3 = \xi_5^2, \xi_6^3 = \gamma$. Let $\xi_2^3 = c$, then (3) is equivalent to $[\xi_1^3 \cdot E^{-1}(x) \oplus (d \oplus \xi_5^2) \cdot x] \oplus [\xi_2^3 \cdot E^{-1}(y) \oplus b \cdot y] \overset{\rho_3}{=} 0$, which is the two linear approximations $d \oplus \xi_5^2 \overset{AES}{\leftarrow} \xi_1^3$ and $b \overset{AES}{\leftarrow} c$ of AES round function, so we have $\rho_3 = \rho_E(d \oplus \xi_5^2 \leftarrow \xi_1^3) \rho_E(b \leftarrow c)$.

For f_4, we have

$$\xi_1^3 \cdot x \oplus c \cdot y \oplus e \cdot z \oplus f \cdot u \oplus \xi_5^2 \cdot v \oplus \gamma \cdot w \overset{\rho_4}{=} \xi_1^4 \cdot x \oplus \xi_2^4 \cdot (y \boxplus z) \oplus \xi_3^4 \cdot u \oplus \xi_4^4 \cdot v \oplus \xi_5^4 \cdot w, \tag{4}$$

which is equivalent to

$$(\xi_1^3 \oplus \xi_1^4) \cdot x \oplus [\xi_2^4 \cdot (y \boxplus z) \oplus c \cdot y \oplus e \cdot z] \oplus (f \oplus \xi_3^4) \cdot u \oplus (\xi_5^2 \oplus \xi_4^4) \cdot v \oplus (\gamma \oplus \xi_5^4) \cdot w \overset{\rho_4}{=} 0.$$

By $\rho_4 \neq 0$ we know that $\xi_1^4 = \xi_1^3, \xi_3^4 = f, \xi_4^4 = \xi_5^2, \xi_5^4 = \gamma$, and we can rewrite the above equation as $\xi_2^4 \cdot (y \boxplus z) \overset{\rho_4}{=} c \cdot y \oplus e \cdot z$, which is the approximation $\xi_2^4 \leftarrow c, e$ of addition modulo 2^{32}. Obviously $\rho_4 = \rho_A(\xi_2^4 \leftarrow c, e)$.

For f_5, the approximation equation is

$$\xi_1^3 \cdot x \oplus \xi_2^4 \cdot y \oplus f \cdot z \oplus \xi_5^2 \cdot u \oplus \gamma \cdot v \overset{\rho_5}{=} \xi_1^5 \cdot x \oplus \xi_2^5 \cdot y \oplus \xi_3^5 \cdot z \oplus \xi_4^5 \cdot u \oplus \xi_5^5 \cdot E^{-1}(v), \tag{5}$$

scilicet

$$(\xi_1^3 \oplus \xi_1^5) \cdot x \oplus (\xi_2^4 \oplus \xi_2^5) \cdot y \oplus (f \oplus \xi_3^5) \cdot z \oplus (\xi_5^2 \oplus \xi_4^5) \cdot u \oplus [\xi_5^5 \cdot E^{-1}(v) \oplus \gamma \cdot v] \stackrel{\rho_5}{=} 0.$$

By $\rho_5 \neq 0$ we know that $\xi_1^5 = \xi_1^3, \xi_2^5 = \xi_2^4, \xi_3^5 = f, \xi_4^5 = \xi_5^2$. Denoting $\xi_5^5 = q$, then (5) can be reduced to $q \cdot E^{-1}(v) \oplus \gamma \cdot v \stackrel{\rho_5}{=} 0$, which is the linear approximation $\gamma \overset{AES}{\leftarrow} q$ of AES round function, so $\rho_5 = \rho_E(\gamma \leftarrow q)$.

For f_6, we have

$$\xi_1^3 \cdot x \oplus \xi_2^4 \cdot y \oplus f \cdot z \oplus \xi_5^2 \cdot u \oplus q \cdot v \stackrel{\rho_6}{=} \alpha \cdot x \oplus \alpha \cdot y \oplus h \cdot u \oplus \beta \cdot (z \boxplus v). \quad (6)$$

By $\rho_6 \neq 0$ we know that $\xi_1^3 = \xi_2^4 = \alpha, \xi_5^2 = h$, and (6) can be simplified to $\beta \cdot (z \boxplus v) \oplus f \cdot z \oplus q \cdot v \stackrel{\rho_6}{=} 0$, which is the linear approximation $\beta \leftarrow f, q$ of addition modulo 2^{32}. Thus $\rho_6 = \rho_A(\beta \leftarrow f, q)$.

Thus, the linear approximation trail of F above can be described as

$$(\gamma, \beta, l, m, n, \gamma) \xrightarrow[d\mathbf{L}=(e\oplus l)||(f\oplus m), \rho_A(\gamma \leftarrow a, n\oplus d)]{f_1} (a, \beta, e, f, d, \gamma) \xrightarrow[\rho_A(\boldsymbol{\sigma}^T a \leftarrow b\oplus\beta, d\oplus h)]{f_2}$$
$$(d \oplus h, b, e, f, h, \gamma) \xrightarrow[\rho_E(d\oplus h \leftarrow \alpha)\rho_E(b \leftarrow c)]{f_3} (\alpha, c, e, f, h, \gamma) \xrightarrow[\rho_A(\alpha \leftarrow e, c)]{f_4} (\alpha, \alpha, f, h, \gamma)$$
$$\xrightarrow[\rho_E(\gamma \leftarrow q)]{f_5} (\alpha, \alpha, f, h, q) \xrightarrow[\rho_A(\beta \leftarrow f, q)]{f_6} (\alpha, \alpha, h, \beta),$$

and its correlation can be computed as

$$\begin{aligned}\rho(a, b, c, d, q) =& \rho_A(\gamma \leftarrow a, n \oplus d)\rho_A(\boldsymbol{\sigma}^T a \leftarrow b \oplus \beta, d \oplus h)\rho_E(d \oplus h \leftarrow \alpha) \\ & \rho_E(b \leftarrow c)\rho_A(\alpha \leftarrow e, c)\rho_E(\gamma \leftarrow q)\rho_A(\beta \leftarrow f, q),\end{aligned}$$

with the constraint $d\mathbf{L} = (e \oplus l)||(f \oplus m)$, and a, b, c, d, q are free masks.

4 Automatic search of linear approximation trails of SNOW-V

STP is an SMT solver which encodes the constraints with CVC, SMT-LIB1 and SMT-LIB2 languages [9]. Since STP solver can model XOR operations much easier than MILP and SAT solvers, we construct STP-based automatic search program for linear approximation trails of SNOW-V. STP solver will return a solution that meets the conditions if there is one. The model of the linear approximation above contains three substitution layers and four layers of addition modulo 2^{32} operations as the nonlinear part. Here we characterize the linear approximation in the way available for STP solver. For convenience, signs of correlation values are temporarily ignored in the process of characterization.

8-bit S-box. We denote the correlation of an S-box with the input mask $x = (x_7, x_6, ..., x_0)$ and output mask $y = (y_7, y_6, ..., y_0)$ as $c(x, y)$. Accurately

characterizing the correlation of an S-box will make the program too large for the STP solver, hence we take the method used in [1] to character the activity of the S-boxes. For every S-box in the process of linear approximation, we introduce a new Boolean function such that

$$f(x,y) = \begin{cases} 1, & if|c(x,y)| \neq 0; \\ 0, & if|c(x,y)| = 0. \end{cases}$$

Since the expressions longer than 256 characters are not supported by STP solver, $f(x,y)$ needs to be converted into a series of shorter constrains that are fully satisfied. By inputting the truth tables, the software LogicFriday can directly give the product-of-sum representation of a Boolean function. For example, the Boolean function with 3 input bits and 1 output bit $h(a_0,a_1,a_2) = a_0a_1a_2 \oplus a_0a_1 \oplus a_2$ has the product-of-sum representation $h(a_0,a_1,a_2) = (a_0|a_1|a_2)$ & $(a_0|\bar{a}_1|a_2)$ & $(\bar{a}_0|a_1|a_2)$. Thus, the Boolean function $h(a_0,a_1,a_2)$ has essential conditions

$$\begin{cases} a_0|a_1|a_2 = 0 \Rightarrow h(a_0,a_1,a_2) = 0, \\ a_0|\tilde{a}_1|a_2 = 0 \Rightarrow h(a_0,a_1,a_2) = 0, \\ \tilde{a}_0|a_1|a_2 = 0 \Rightarrow h(a_0,a_1,a_2) = 0. \end{cases}$$

In the same way, $f(x,y)$ can be converted into a series of logical conditions. With the additional constraint introduced: $f(x,y) = x_0|x_1|\cdots|x_7|y_0|y_1|\cdots|y_7$, $f(x,y)$ is the linear activity of the S-box, i.e. $f(x,y) = 1$ if and only if the S-box is linear active.

Addition modulo 2^{32}. Johan Wallén proposed an recursive method to compute the correlation with given input and output masks efficiently [16]. Then the result was improved by Schulte-Geers by proving that modulo addition is CCZ-equivalent to a vectorial quadratic Boolean function [14]. Denoting u as the output mask, v, w as the input masks, x_i as the i-th bit of Boolean vector x, the constraints to obtain a valid linear approximation shall be expressed as [14]:

$$\begin{aligned} &z_{n-1} = 0, \\ &z_j = z_{j+1} \oplus u_{j+1} \oplus v_{j+1} \oplus w_{j+1} (0 \leq j < n-1), \\ &z_i \geq u_i \oplus v_i (0 \leq i < n), \\ &z_i \geq u_i \oplus w_i, \end{aligned}$$

in which z is a dummy variable. The correlation of the linear approximation is not zero if and only if there exists a z satisfying the constraints, and is given by $cor(u,v,w) = (-1)^{(u \oplus v)\cdot(u \oplus w)} 2^{-wt(z)}$ when it is not zero.

Objective function. As the maximum of the absolute values of correlations of an S-box is 2^{-3}, we take $3\sum_{i=1}^{48} f^i(x,y) + \sum_{j=1}^{16} wt(z^{(j)})$ instead of the accurate correlation of a linear trail, where $f^i(x,y)$ denotes the activity of the i-th S-box and $z^{(j)}$ denotes the dummy variable of j-th modulo addition operation. This objective function may not search out the optimal linear approximation trail, but can

find trails with relatively high correlation. Meanwhile, we can give an untight upper bound on the correlation of linear trails: the precise correlation of a linear trail shall less than 2^{-k} when the trail satisfies $3\sum_{i=1}^{48} f^i(x,y) + \sum_{j=1}^{16} wt(z^{(j)}) < k$.

The precise correlation and its sign. After STP solver returns a linear approximation trail that satisfies all constraints, we verify the trail and evaluate its accurate correlation with the real correlations of S-boxes. The sign of correlation is also determined in the process of verification.

Finding more trails. According to the properties of Walsh spectrum of composite functions, the correlation of a binary linear approximation of SNOW-V can be computed as

$$c(\alpha,\beta,\gamma,l,m,n,h) = \sum_{a,b,c,d,q} \rho(a,b,c,d,q).$$

Assuming that the trail $(\alpha_0,\beta_0,\gamma_0,l_0,m_0,n_0,h_0,a_0,b_0,c_0,d_0,q_0)$ has been found, we can keep searching for other new solutions by introducing the additional constraints:

$$\alpha=\alpha_0,\beta=\beta_0,\gamma=\gamma_0,l=l_0,m=m_0,n=n_0,h=h_0,$$
$$(a\oplus a_0)|(b\oplus b_0)|(c\oplus c_0)|(d\oplus d_0)|(q\oplus q_0)\neq 0.$$

Different solutions can be generated one by one in this way, and the binary correlation gradually approaches its real value by summing up the correlations of linear trails. We build the automatic search program for the linear approximation above. With the constraint that $3\sum_{i=1}^{48} f^i(x,y) + \sum_{j=1}^{16} wt(z^{(j)}) < k$, STP solver returns False when $k=44$ and returns a solution when $k=45$. Thus we know that 2^{-44} is an upper bound on correlation of linear trails, but may not be tight. The best result we have found is

$$\alpha = l = c = 0x1, \texttt{0}, \texttt{0}, \texttt{0}$$
$$\beta = m = 0x80, \texttt{0}, \texttt{0}, \texttt{0}$$
$$\gamma = h = b = 0x81ec5a80, \texttt{0}, \texttt{0}, \texttt{0}$$
$$n = 0x81ec5a00, \texttt{0}, \texttt{0}, \texttt{0}$$
$$a = 0xc1000000, \texttt{0}, \texttt{0}, \texttt{0}$$
$$d = \texttt{0}, \texttt{0}, \texttt{0}, \texttt{0}.$$

with the correlation $2^{-49.83}$ (The symblo '0' denotes 32-bit 0). Once the input and output mask $(\alpha,\beta,\gamma,l,m,n,h)$ been fixed, we gradually add constraints and search out all the linear trails with the constraint $3\sum_{i=1}^{48} f^i(x,y) + \sum_{j=1}^{16} wt(z^{(j)}) < 50$ and sum the correlations up. The overall correlation reaches $2^{-49.54}$. Trails we have found are shown in Appendix.

5 A correlation attack on SNOW-V

In this section, we present a correlation attack on SNOW-V based on the linear approximation with the correlation $2^{-49.54}$ given in Section 4.

5.1 General description of the presented correlation attack on SNOW-V

We call the state of LFSR that produce the first keystream word as the initial state of LFSR. Our aim is to recover the initial state of LFSR. By the result above, we have

$$\alpha \cdot z_{t-1} \oplus \beta \cdot z_t \oplus \gamma \cdot z_{t+1} \oplus l \cdot T_1^{(t-1)} \oplus m \cdot T_1^{(t)} \oplus n \cdot T_1^{(t+1)} \oplus h \cdot T_2^{(t)} \overset{2^{-49.54}}{=} 0.$$

We assume $u = (u_{511}, u_{510}, \cdots, u_0)^T$ and $\hat{u} = (\hat{u}_{511}, \hat{u}_{510}, \cdots, \hat{u}_0)^T$ as the initial state and guessed initial state respectively. Since the output of LFSR at clock t can always be expressed as a linear combination of the initial state, i.e. there always exists a $\Gamma_t \in \{0,1\}^{512}$ such that $\Gamma_t \cdot u = l \cdot T_1^{(t-1)} \oplus m \cdot T_1^{(t)} \oplus n \cdot T_1^{(t+1)} \oplus h \cdot T_2^{(t)}$, we can construct a distinguisher with the form

$$\begin{aligned}\phi_t(\hat{u}) =& \alpha \cdot z_{t-1} \oplus \beta \cdot z_t \oplus \gamma \cdot z_{t+1} \oplus \Gamma_t \cdot \hat{u} \\ =& \alpha \cdot z_{t-1} \oplus \beta \cdot z_t \oplus \gamma \cdot z_{t+1} \\ & \oplus l \cdot T_1^{(t-1)} \oplus m \cdot T_1^{(t)} \oplus n \cdot T_1^{(t+1)} \oplus h \cdot T_2^{(t)} \oplus \Gamma_t \cdot (u \oplus \hat{u}).\end{aligned}$$

$\phi_t(\hat{u})$ will show the correlation $\rho = 2^{-49.54}$ when $\hat{u} = u$, otherwise the distribution of $\phi_t(\hat{u})$ is uniform. With this analysis, we recover the initial LFSR state in two steps:

Preprocessing stage: Let the most significant B bits of binary vector $x = (x_{511}, x_{510}, \cdots, x_0)^T$ be $x^h = (x_{511}, x_{510}, \cdots, x_{512-B})^T$, the least significant 512-B bits be $x^l = (x_{511-B}, x_{510-B}, \cdots, x_0)^T$ and the number of keystream words produced by a pair of key and IV be N. For $1 \le i_1, i_2 \le N$ it follows that

$$(\Gamma_{i_1} \oplus \Gamma_{i_2}) \cdot u = (\Gamma_{i_1}^h \oplus \Gamma_{i_2}^h) \cdot u^h \oplus (\Gamma_{i_1}^l \oplus \Gamma_{i_2}^l) \cdot u^l.$$

If $\Gamma_{i_1}^l = \Gamma_{i_2}^l$, the equation above is converted into $(\Gamma_{i_1} \oplus \Gamma_{i_2}) \cdot u = (\Gamma_{i_1}^h \oplus \Gamma_{i_2}^h) \cdot u^h$. As the event $\phi_{i_1}(u) = 0$ is independent of the event $\phi_{i_2}(u) = 0$ and $p(\phi_{i_1}(u) = 0) = p(\phi_{i_2}(u) = 0) = \frac{1}{2} + \frac{1}{2}\rho$, we have $p(\phi_{i_1}(u) \oplus \phi_{i_2}(u) = 0) = \frac{1}{2} + \frac{1}{2}\rho^2$. Therefore we can get a parity check equation of B bits of the initial state u

$$\alpha \cdot (z_{i_1-1} \oplus z_{i_2-1}) \oplus \beta \cdot (z_{i_1} \oplus z_{i_2}) \oplus \gamma \cdot (z_{i_1+1} \oplus z_{i_2+1}) \oplus (\Gamma_{i_1}^h \oplus \Gamma_{i_2}^h) \cdot u^h \overset{\rho^2}{=} 0,$$

if $\Gamma_{i_1}^l = \Gamma_{i_2}^l$ holds. Since the probability $p(\Gamma_{i_1}^l = \Gamma_{i_2}^l) = 2^{-512-B}$, the expected number of parity check equations with $\Gamma_{i_1}^l = \Gamma_{i_2}^l$ among C_N^2 pairs of Γ_i is

$M = C_N^2 2^{-(512-B)} \approx 2^{-(513-B)} N^2$. Therefore, we can find $2^{-(513-B)} N^2$ parity check equations in preprocessing stage on average.

Processing stage: Among the M parity check equations we denote the j-th equation as $(\alpha, \beta, \gamma) \cdot Z_j \oplus \Gamma_j^h \cdot u^h = 0$, where $Z_j = (z_{i_1-1} \oplus z_{i_2-1}, z_{i_1} \oplus z_{i_2}, z_{i_1+1} \oplus z_{i_2+1})$ and $\Gamma_j^h = (\Gamma_{i_1}^h \oplus \Gamma_{i_2}^h)$. For each guess of the B bits $\hat{u}^h \in \{0,1\}^B$ of the initial state u, we evaluate the parity checks, get

$$T(\hat{u}^h) = \sum_{j=1}^{M} (-1)^{(\alpha,\beta,\gamma)\cdot Z_j \oplus \Gamma_j^h \cdot \hat{u}^h},$$

and predict the $\hat{u}$ that maximizes $T(\hat{u}^h)$ as the correct one. For the remaining $512-B$ bits, the above process can be repeated when the first B bits are known. Thus, all the initial 512 bits of the LFSR can be recovered.

5.2 Success probability and complexity

For linear attacks, the relationship between the probability of success and the number of check equations is given in [15]:

Definition 1 [15]. If a B-bit key is attacked and the right key is ranked r-th among all 2^B candidates, $a = B - \log_2 r$ is called the advantage provided by the attack.

In this paper we refer to the advantage defined by Definition 1 as *gain*.

Lemma 1 [15]. Let p_s be the probability that a linear attack on a B-bit subkey, with a linear approximation of probability $p = \frac{1}{2} + \frac{1}{2}\rho$ and M known parity check equations, delivers an a-bit or higher gain. Under the assuming that the linear approximation's probability to hold is independent for each guessed key and its probability is equal to 1/2 for all wrong keys, we have for sufficiently large B and M that

$$p_s = \Phi\left(2\sqrt{M}\left|p - \frac{1}{2}\right| - \Phi^{-1}(1 - 2^{-a-1})\right),$$

where $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{x^2}{2}} dx$ is the distribution function of the standard normal distribution.

Corollary 1 With the assumptions of Lemma 1,

$$M = \frac{1}{\rho^2}\left(\Phi^{-1}(p_s) + \Phi^{-1}(1 - 2^{-a-1})\right)^2$$

parity check equations are needed in a linear attack to accomplish an a-bit gain with a success probability of p_s.

By the results of [2], we use the formula $\Phi^{-1}(1-\lambda) \overset{\lambda \to 0^+}{\approx} \sqrt{-2\ln\lambda}$ to approximate $\Phi^{-1}(1 - 2^{-a-1})$. Hence, we have $M \approx \frac{1}{\rho^2}\left(\Phi^{-1}(p_s) + \sqrt{2(a+1)\ln 2}\right)^2$ for sufficiently large a.

The complexity can be evaluated as follows. In the preprocessing stage, we evaluate and store each $\Gamma_i \in \{0,1\}^{512}$ for $1 \le i \le N$. Then we sort Γ_i according to the value of Γ_i^l such that $\Gamma_{i_1}^l = \Gamma_{i_2}^l$ holds for any i_1 and i_2 in the same set. Thus we can construct a series of parity check equations which is only related to the most significant B bits of initial state. The time complexity of preprocessing stage is $O(N) + O(N\log_2 N)$, and memory complexity is $O(N)$.

In the processing stage, $T(\hat{u}^h)$ is calculated for each guessed $\hat{u}^h \in \{0,1\}^B$ by evaluating M parity check equations. When $B > \log M$, denoting Γ_j^{h1} and $\hat{u}^{h1}$ as the most significant $\lceil \log M \rceil$bits, Γ_j^{h2} and $\hat{u}^{h2}$ as the least significant $B - \lceil \log M \rceil$ bits of Γ_j^h and $\hat{u}^h$ respectively, we can accelerate the process using fast Walsh transformation by

$$\begin{aligned}
T(\hat{u}^h) &= \sum_{j=1}^{M} (-1)^{(\alpha,\beta,\gamma)\cdot Z_j \oplus \Gamma_j^h \cdot \hat{u}^h} \\
&= \sum_{\zeta \in \{0,1\}^{\lceil \log M \rceil}} \sum_{j, \Gamma_j^{h1} = \zeta} (-1)^{(\alpha,\beta,\gamma)\cdot Z_j} \cdot (-1)^{\Gamma_j^{h1}\cdot \hat{u}^{h1}} \cdot (-1)^{\Gamma_j^{h2}\cdot \hat{u}^{h2}} \\
&= \sum_{\zeta \in \{0,1\}^{\lceil \log M \rceil}} \sum_{j, \Gamma_j^{h1} = \zeta} (-1)^{(\alpha,\beta,\gamma)\cdot Z_j} \cdot (-1)^{\Gamma_j^{h2}\cdot \hat{u}^{h2}} \cdot (-1)^{\zeta \cdot \hat{u}^{h1}} \\
&= \sum_{\zeta \in \{0,1\}^{\lceil \log M \rceil}} (-1)^{\zeta \cdot \hat{u}^{h1}} \sum_{j, \Gamma_j^{h} = \zeta} (-1)^{(\alpha,\beta,\gamma)\cdot Z_j} \cdot (-1)^{\Gamma_j^{h2}\cdot \hat{u}^{h2}} \\
&= \sum_{\zeta \in \{0,1\}^{\lceil \log M \rceil}} (-1)^{\zeta \cdot \hat{u}^{h1}} g_{\hat{u}^{h2}}(\zeta),
\end{aligned}$$

where $g(\zeta) = \sum_{j, \Gamma_j^{h1} = \zeta} (-1)^{(\alpha,\beta,\gamma)\cdot Z_j \oplus \Gamma_j^{h2}\cdot \hat{u}^{h2}}$.

For each guessed $\hat{u}^{h2} \in \{0,1\}^{B - \lceil \log M \rceil}$ and $\zeta \in \{0,1\}^{\lceil \log M \rceil}$, we compute $g_{\hat{u}^{h2}}(\zeta)$ and get $T(\hat{u}^h) = \sum_{\zeta \in \{0,1\}^{\lceil \log M \rceil}} (-1)^{\zeta \cdot \hat{u}^{h1}} g_{\hat{u}^{h2}}(\zeta)$ by calculating the Walsh transform of $g_{\hat{u}^{h2}}(\zeta)$. This process can be done with time complexity

$$2^{B - \lceil \log M \rceil}(M + \lceil \log M \rceil 2^{\lceil \log M \rceil}) \approx 2^B(1 + \lceil \log M \rceil)$$

and memory complexity $O(2^B)$. By Corollary 1, we have

$$M = \frac{1}{\rho^2}\left(\Phi^{-1}(p_s) + \Phi^{-1}(1 - 2^{-B-1})\right)^2$$

when the correct u^h is predicted as the top ranked, i.e. $a = B$. Therefore, we can work out M with fixed p_s and B, then compute N by $M \approx 2^{-(513-B)}N^2$. Finally, the values which minimize the time complexity $N(\log N + 1) + 2^B(1 + \lceil \log M \rceil)$ shall be taken to determine the total complexity.

We test different choices for p_s and B and find that $M \approx 2^{207}$ and $N \approx 2^{240}$ under $p_s = 0.999211$ and $B = 240$, which makes the total complexity lowest.

The time complexity of the preprocessing stage is $2^{247.91}$, memory complexity is 2^{240}. In the processing stage the time complexity is $2^{247.7}$, memory complexity is 2^{240}. Thus, the attack can be done with the total time complexity $2^{248.81}$, memory complexity 2^{240} and 2^{240} keystream words given. It is easy to see that with the recovery of states of LFSR in encryption stage, one can recovery the three memories R_1, R_2 and R_3 in encryption stage with a time complexity less than 2^{128}. Thus the problem is remained that how to recovery the original key effectively if one has recovered the internal states in encryption stage.

6 Correlation attack on SNOW-Vi

6.1 Linear approximation of SNOW-Vi

In March of 2021, Ekdahl et al. proposed SNOW-Vi. Besides the field and update transformation of the LFSR, the tap $T_2^{(t)} = (a_7^{(8t)}, a_6^{(8t)}, \cdots, a_0^{(8t)})$ of SNOW-V was changed to $T_2^{(t)} = (a_{15}^{(8t)}, a_{14}^{(8t)}, \cdots, a_8^{(8t)})$ as well. We have experimentally confirmed that, regarding every bit in four taps $T_1^{(t-1)}, T_1^{(t)}, T_1^{(t+1)}$ and $T_2^{(t)}$ as the coefficient vector of the initial state, the 512×512 binary matrix composed of these vectors is full rank, i.e. $T_1^{(t-1)}, T_1^{(t)}, T_1^{(t+1)}$ and $T_2^{(t)}$ do not have linear relationship any more. Thus, we modify the six functions to be

$$
\begin{aligned}
&f_1(x, y, z, t, u, v, w) = ((x \boxminus u), y, z, t, v, w),\\
&f_2(x, y, z, u, v, w) = ((\sigma^{-1}(x) \boxminus y) \oplus v, y, z, u, w),\\
&f_3(x, y, z, u, v) = (E^{-1}(x), E^{-1}(y), z, u, v),\\
&f_4(x, y, z, u, v) = (x, (y \boxplus z), u, v),\\
&f_5(x, y, z, u) = (x, y, z, E^{-1}(u)),\\
&f_6(x, y, z, u) = (x, y, (z \boxplus u)).
\end{aligned}
$$

The composite function becomes

$$F(x, y, z, t, u, v, w) = (f_6 \circ f_5 \circ f_4 \circ f_3 \circ f_2 \circ f_1)(x, y, z, t, u, v, w),$$

with 7 input words and 3 output words. Using the same method and symbols as in Section 3, we consider the linear approximation $(\gamma, \beta, l, m, n, h, \gamma) \xrightarrow{F} (\alpha, \alpha, \beta)$. Taking 7 independent and uniform distributed words as the input variables:

$$
\begin{aligned}
&(x, y, z, t, u, v, w)\\
&= (\sigma(R_2 \boxplus (R_3 \oplus T_2^{(t)})) \boxplus T_1^{(t+1)}, R_2, T_1^{(t-1)}, T_1^{(t)}, T_1^{(t+1)}, T_2^{(t)}, E(R_1)),
\end{aligned}
$$

we have

$$f_1(x, y, z, t, u, v, w) = (\sigma(R_2 \boxplus (R_3 \oplus T_2^{(t)})), R_2, T_1^{(t-1)}, T_1^{(t)}, T_2^{(t)}, E(R_1)),$$

$$f_2 \circ f_1(x, y, z, t, u, v, w) = (R_3, R_2, T_1^{(t-1)}, T_1^{(t)}, E(R_1)),$$

$$f_3 \circ f_2 \circ f_1(x, y, z, t, u, v, w) = (E^{-1}(R_3), E^{-1}(R_2), T_1^{(t-1)}, T_1^{(t)}, E(R_1)),$$

$$f_4 \circ f_3 \circ f_2 \circ f_1(x,y,z,t,u,v,w) = (E^{-1}(R_3), E^{-1}(R_2) \boxplus T_1^{(t-1)}, T_1^{(t)}, E(R_1)),$$
$$f_5 \circ f_4 \circ f_3 \circ f_2 \circ f_1(x,y,z,t,u,v,w) = (E^{-1}(R_3), E^{-1}(R_2) \boxplus T_1^{(t-1)}, T_1^{(t)}, R_1),$$
$$f_6 \circ f_5 \circ f_4 \circ f_3 \circ f_2 \circ f_1(x,y,z,t,u,v,w) = (E^{-1}(R_3), E^{-1}(R_2) \boxplus T_1^{(t-1)}, T_1^{(t)} \boxplus R_1).$$

Then the equation of the linear approximation $(\gamma,\beta,l,m,n,h,\gamma) \xrightarrow{F} (\alpha,\alpha,\beta)$ is

$$\begin{aligned}
&(\gamma,\beta,l,m,n,h,\gamma)\\
&\cdot(\sigma(R_2 \boxplus (R_3 \oplus T_2^{(t)})) \boxplus T_1^{(t+1)}, R_2, T_1^{(t-1)}, T_1^{(t)}, T_1^{(t+1)}, T_2^{(t)}, E(R_1))\\
&\oplus(\alpha,\alpha,\beta) \cdot (E^{-1}(R_3), E^{-1}(R_2) \boxplus T_1^{(t-1)}, T_1^{(t)} \boxplus R_1)\\
=\ &\alpha \cdot z_{t-1} \oplus \beta \cdot z_t \oplus \gamma \cdot z_{t+1} \oplus l \cdot T_1^{(t-1)} \oplus m \cdot T_1^{(t)} \oplus n \cdot T_1^{(t+1)} \oplus h \cdot T_2^{(t)}\\
\overset{\rho}{=}\ &0.
\end{aligned}$$

The linear approximation above can be expressed as

$$\begin{aligned}
&(\gamma,\beta,l,m,n,h,\gamma) \xrightarrow[\rho_A(\gamma \leftarrow a,n)]{f_1} (a,\beta,l,m,h,\gamma) \xrightarrow[\rho_A(\boldsymbol{\sigma}^T a \leftarrow b \oplus \beta, h)]{f_2} (h,b,l,m,\gamma)\\
&\xrightarrow[\rho_E(h \leftarrow \alpha)\rho_E(b \leftarrow c)]{f_3} (\alpha,c,l,m,\gamma) \xrightarrow[\rho_A(\alpha \leftarrow l,c)]{f_4} (\alpha,\alpha,m,\gamma) \xrightarrow[\rho_E(\gamma \leftarrow q)]{f_5} (\alpha,\alpha,m,q)\\
&\xrightarrow[\rho_A(\beta \leftarrow m,q)]{f_6} (\alpha,\alpha,\beta),
\end{aligned}$$

and the correlation of a linear trail can be computed as

$$\begin{aligned}
\rho(a,b,c,q) =&\rho_A(\gamma \leftarrow a,n)\rho_A(\boldsymbol{\sigma}^T a \leftarrow b \oplus \beta, h)\rho_E(h \leftarrow \alpha)\rho_E(b \leftarrow c)\\
&\rho_A(\alpha \leftarrow l,c)\rho_E(\gamma \leftarrow q)\rho_A(\beta \leftarrow m,q),
\end{aligned}$$

where a,b,c,q are the free masks.

6.2 Compared with the linear approximation of SNOW-V

The correlation of a linear trail of SNOW-V is

$$\begin{aligned}
\rho(a,b,c,d,q) =&\rho_A(\gamma \leftarrow a, n \oplus d)\rho_A(\boldsymbol{\sigma}^T a \leftarrow b \oplus \beta, d \oplus h)\rho_E(d \oplus h \leftarrow \alpha)\rho_E(b \leftarrow c)\\
&\rho_A(\alpha \leftarrow e,c)\rho_E(\gamma \leftarrow q)\rho_A(\beta \leftarrow f,q).
\end{aligned}$$

Since $d = 0$, $e = l, f = m$ holds for the 6 linear trails we have searched out, for this type of trails, the expression can be reduced to

$$\begin{aligned}
\rho(a,b,c,0,q) =&\rho_A(\gamma \leftarrow a,n)\rho_A(\boldsymbol{\sigma}^T a \leftarrow b \oplus \beta, h)\rho_E(h \leftarrow \alpha)\rho_E(b \leftarrow c)\\
&\rho_A(\alpha \leftarrow l,c)\rho_E(\gamma \leftarrow q)\rho_A(\beta \leftarrow m,q),
\end{aligned}$$

which is the same as the correlation $\rho(a,b,c,q)$ of SNOW-Vi. Hence, we have the straightforward observation.

Proposition 1 For any trail of the linear approximation process of SNOW-Vi above, there exists a linear trail of SNOW-V with the same mask

$$(\alpha,\beta,\gamma,l,m,n,h,a,b,c,q)$$

and the same correlation.

Proposition 1 indicates that the set consists of all linear trails of SNOW-Vi is a subset of the set of linear trails of SNOW-V, so the results of SNOW-V in this paper are also appropriate for SNOW-Vi. Therefore, we can confirm that there is no linear approximation trail with correlation higher than 2^{-44}, and we could approximate SNOW-Vi with the same correlation of SNOW-V under

$$\begin{aligned}
&\alpha = l = 0x1, 0, 0, 0\\
&\beta = m = 0x80, 0, 0, 0\\
&\gamma = h = 0x81ec5a80, 0, 0, 0\\
&n = 0x81ec5a00, 0, 0, 0
\end{aligned}$$

with the 6 trails in Appendix. Similarly, the correlation attack presented in Section 5 with time complexity $2^{248.81}$, memory complexity 2^{240} and 2^{240} words given is effective for SNOW-Vi as well.

7 Conclusion

In this paper, we study the linear approximation of the nonlinear functions of SNOW-V and SNOW-Vi by the composite function techniques. By the Walsh spectrum theorem of composite function, we propose a method for searching linear trails with high correlation of SNOW-V and SNOW-Vi in a wide range. As the automatic search technique is available for this framework, the search efficiency has been improved greatly. For SNOW-V, we search out a binary linear approximation with correlation $2^{-49.54}$, which improves greatly the results in the design document. Using the linear approximation we launch a correlation attack with time complexity $2^{248.81}$, memory complexity 2^{240} and 2^{240} words given. For SNOW-Vi, the binary linear approximation is also valid, and the correlation attack on SNOW-V is effective for SNOW-Vi as well. The results of this paper show that SNOW-V and SNOW-Vi can be attacked with complexity less than key exhaustion, with the design constraint that the maximum of keystream length with a single pair of key and IV is 2^{64} ignored. Due to the solver limitation, the linear trails we have found may not be optimal. Therefore, in terms of provable security, we only give a untight upper bound 2^{-44} of the correlation of linear trails. It remains to be further verified whether SNOW-V and SNOW-Vi still has linear trails or binary approximation with greater correlation, or multi-bit distribution with larger SEI.

References

1. Abdelkhalek, A., Sasaki, Y., Todo, Y., Tolba, M., Youssef, A.M.: MILP modeling for (large) S-boxes to optimize probability of differential characteristics. IACR Transactions on Symmetric Cryptology pp. 99–129 (2017).

2. Blondeau, C., Gérard, B., Tillich, J.P.: Accurate estimates of the data complexity and success probability for various cryptanalyses. Designs, Codes and Cryptography 59(1), 3–34 (2011).
3. Chose, P., Joux, A., Mitton, M.: Fast correlation attacks: An algorithmic point of view. In: EUROCRYPT 2002. pp. 209–221.
4. Coppersmith, D., Halevi, S., Jutla, C.: Cryptanalysis of stream ciphers with linear masking. Advances in Cryptology–CRYPTO 2002 pp.515–532.
5. Ekdahl, P., Johansson, T.: SNOW - A new stream cipher. In: Proceedings of First Open NESSIE Workshop, KU-Leuven. pp. 167–168. Citeseer (2000).
6. Ekdahl, P., Johansson, T.: A new version of the stream cipher SNOW. In: International Workshop on Selected Areas in Cryptography 2002. pp. 47–61.
7. Ekdahl, P., Johansson, T., Maximov, A., Yang, J.: A new SNOW stream cipher called SNOW-V. IACR Transactions on Symmetric Cryptology pp. 1–42 (2019)
8. Ekdahl, P., Maximov, A., Johansson, T., Yang, J.: SNOW-Vi: an extreme performance variant of SNOW-V for lower grade CPUs. In: Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks. pp. 261–272 (2021)
9. Ganesh, V., Hansen, T., Soos, M., Liew, D., Govostes, R.: STP constraint solver (2011)
10. Gong, X., Zhang, B.: Fast computation of linear approximation over certain composition functions and applications to SNOW 2.0 and SNOW 3G. Designs, Codes and Cryptography 88(11), 2407–2431 (2020)
11. Gong, X., Zhang, B.: Resistance of SNOW-V against fast correlation attacks. IACR Transactions on Symmetric Cryptology pp. 378–410 (2021)
12. Meier, W., Staffelbach, O.: Fast correlation attacks on certain stream ciphers. Journal of cryptology 1(3), 159–176 (1989)
13. Nyberg, K., Wallén, J.: Improved linear distinguishers for SNOW 2.0. In: International Workshop on Fast Software Encryption. pp. 144–162. Springer (2006)
14. Schulte-Geers, E.: On CCZ-equivalence of addition mod 2^n. Designs, Codes and Cryptography 66(1), 111–127 (2013)
15. Selçuk, A.A.: On probability of success in linear and differential cryptanalysis. Journal of Cryptology 21(1), 131–147 (2008)
16. Wallén, J.: Linear approximations of addition modulo 2^n. In: International Workshop on Fast Software Encryption 2003. pp. 261–273.
17. UEA2&UIA, I.: Specification of the 3GPP confidentiality and integrity algorithms UEA2& UIA2 (2006)
18. Watanabe, D., Biryukov, A., De Canniére, C.: A distinguishing attack of SNOW 2.0 with linear masking method. In: International Workshop on Selected Areas in Cryptography 2003. pp. 222–233.
19. Yang, J., Johansson, T., Maximov, A.: Vectorized linear approximations for attacks on SNOW 3G. IACR Transactions on Symmetric Cryptology pp. 249–271 (2019)
20. Yang, J., Johansson, T., Maximov, A.:)Improved guess-and-determine and distinguishing attacks on SNOW-V. IACR Cryptol. ePrint Arch. 2021,544 (2021)
21. Zhang, B., Xu, C., Meier, W.: Fast correlation attacks over extension fields, large-unit linear approximation and cryptanalysis of SNOW 2.0. In: Gennaro, R., Robshaw, M. (eds.) Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015,

Appendix.

Six linear trails of SNOW-V with $\alpha = l = 0x1, 0, 0, 0, \beta = m = 0x80, 0, 0, 0, \gamma = h = 0x81ec5a80, 0, 0, 0, n = 0x81ec5a00, 0, 0, 0$

a	b	c	d	q	$-\log_2\|\rho\|$	Sign
0xc1000000,0,0,0	0x81ec5a80,0,0,0	0x1,0,0,0	0,0,0,0	0xa0,0,0,0	49.830	+
0xc1000000,0,0,0	0x81ec5a80,0,0,0	0x1,0,0,0	0,0,0,0	0xc0,0,0,0	51.830	+
0xa1000000,0,0,0	0x81ec5a80,0,0,0	0x1,0,0,0	0,0,0,0	0xa0,0,0,0	51.830	+
0xc1000000,0,0,0	0x81ec5a80,0,0,0	0x1,0,0,0	0,0,0,0	0x90,0,0,0	52.245	-
0xc1000000,0,0,0	0x81ec5a80,0,0,0	0x1,0,0,0	0,0,0,0	0x88,0,0,0	52.508	+
0xa1000000,0,0,0	0x81ec5a80,0,0,0	0x1,0,0,0	0,0,0,0	0xc0,0,0,0	53.830	-