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

Zhen Shi(✉), Chenhui Jin, Jiyan Zhang, Ting Cui, Lin Ding, and Yu Jin

PLA SSF Information Engineering University, Zhengzhou 450000, China
shizhenieu@126.com, jinchenhui@126.com, xdzhangjiyan@126.com, cuiting_1209@126.com, dinglin_cipher@163.com, jinyu0801@foxmail.com

Abstract. In this paper, a method for searching correlations between the binary stream of Linear Feedback Shift Register (LFSR) and the keystream of SNOW-V and SNOW-Vi is presented based on the technique of approximation to composite functions. With the aid of the linear relationship between the four taps of LFSR input into Finite State Machine (FSM) at three consecutive clocks, we present an automatic search model based on the SAT/SMT technique and search out a series of linear approximation trails with high correlation. By exhausting the intermediate masks, we find a binary linear approximation with a correlation $-2^{-47.76}$. Using such approximation, we propose a correlation attack on SNOW-V with an expected time complexity $2^{246.53}$, a memory complexity $2^{238.77}$ and $2^{237.5}$ keystream words generated by the same key and Initial Vector (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. To the best of our knowledge, this is the first known attack on full SNOW-V and SNOW-Vi, which is better than the exhaustive key search with respect to time complexity. The results indicate that neither SNOW-V nor SNOW-Vi can guarantee the 256-bit security level if we ignore the design constraint that the maximum length of keystream for a single pair of key and IV is less than 2^{64}.

Keywords: SNOW-V · SNOW-Vi · Cryptanalysis · Linear Approximation · Automatic Search · Correlation Attack

1 Introduction

SNOW-V is a new member of the SNOW family stream ciphers following SNOW 1.0 [1], SNOW 2.0 [2] and SNOW 3G [3]. SNOW 3G, which is used as the core of 3G Partnership Project (3GPP) Confidentiality and Integrity Algorithms UEA2 & UIA2 for UMTS and LTE, enhances the security under the 128-bit key. In 2018, SNOW-V, which was proposed for a standard encryption scheme for 5G mobile communication system by Ekdahl et al. [4], was announced to satisfy the 256-bit security level requirement from 3GPP with a 256-bit key and 128-bit Initial Vector (IV). Lately, Ekdahl et al. [5] proposed SNOW-Vi as an extreme performance variant of SNOW-V. Compared with SNOW 3G, the structure of SNOW-V and SNOW-Vi keeps the same, except that a couple of Linear Feedback Shift Registers (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 rises significantly. This makes SNOW-V and SNOW-Vi difficult to analyze.

Up to now, several security evaluations on various versions of SNOW-V and SNOW-Vi have been published. Jiao et al. [7] proposed a byte-based guess and determine attack on SNOW-V with a time complexity of 2^{406}, using only seven keystream words. At ToSC 2021, Gong and Zhang [8] performed a linear cryptanalysis of SNOW-V and proposed correlation attacks against three reduced variants. For the reduced variant SNOW-V$_{\boxplus_{32},\boxplus_8}$, they mounted a fast correlation attack with a time complexity $2^{377.01}$, a memory complexity 2^{363}, and $2^{253.73}$ keystream outputs. Yang et al. [9] proposed a guess and determine attack against the full SNOW-V with a time complexity of 2^{378}, and a distinguishing attack against the reduced variant SNOW-V$_{\oplus}$ with a complexity 2^{303}. For the initialization phase, Hoki et al. [10] investigated the security of the initialization of SNOW-V at ACISP 2021, using Mixed-integer Linear Programming (MILP) model to efficiently search for integral and differential characteristics. The resulting distinguishing or key recovery attacks are applicable to SNOW-V with reduced initialization rounds of five, out of the original 16 rounds. However, none of these cryptanalysis efforts result in a valid attack against SNOW-V and SNOW-Vi, which is faster than the exhaustive key search.

Our contributions. In this paper, we focus on the security levels of SNOW-V and SNOW-Vi against correlation attack.

- The search for high correlation binary approximations is quite a challenge in this cryptanalysis. We introduce a newly constructed composite function, which helps to equivalently transform the linear approximation of the Finite State Machine (FSM) part into that of the composition of several simple functions. In this way, the correlation of a linear approximation trail can be computed by multiplying those of sub-functions with no need to consider the dependence between sub-functions, and the correlations of linear approximations based on three consecutive outputs of SNOW-V can be evaluated by the linear approximation trails of the composite function.
- A series of automatic search models have been widely used to search linear trails with high correlations of block ciphers, but just a few for stream ciphers [11] as far as we know. As we have already converted the approximations of FSM into those of a composite function, we can launch a wide range of search for linear trails by taking advantage of automatic search techniques, and approximate the accurate correlations with the correlations of linear trails.
- With the aid of the linear relationship between the four taps of LFSR input into FSM at three consecutive clocks, we present an automatic search model based on SAT/SMT technique and search out a series of linear approximation

trails with high correlations. By exhausting the intermediate masks, we find an accurate binary linear approximation with a correlation $-2^{-47.76}$.

- Using the approximation, we mount a correlation attack with an expected time complexity $2^{246.53}$, a memory complexity $2^{238.77}$, and $2^{237.5}$ keystream words, which can recover the internal state of SNOW-V at 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 that on SNOW-V.

As far as we know, it is the first attack with the time complexity less than that of exhaustive attack on SNOW-V and SNOW-Vi (see Table 1).

Table 1. Summary of the attacks on SNOW-V and SNOW-Vi

Version	Technique	Round	Time	Data	References
SNOW-V$_\oplus$	Distinguishing attack	full	2^{303}	2^{303}	[9]
SNOW-V$_{\sigma_0}$	Correlation attack	full	$2^{251.93}$	$2^{103.83}$	[8]
SNOW-V$_{\boxplus_{32},\boxplus_8}$	Correlation attack	full	$2^{377.01}$	$2^{253.73}$	[8]
SNOW-V	Differential attack	4	$2^{153.97}$	$2^{26.96}$	[10]
	Guess and Determine	full	2^{512}	7	[6]
	Guess and Determine	full	2^{406}	7	[7]
	Guess and Determine	full	2^{378}	8	[9]
	Correlation attack	full	$2^{246.53}$	$2^{237.5}$	Section 6
SNOW-Vi	Differential attack	4	$2^{233.99}$	$2^{7.94}$	[10]
	Correlation attack	full	$2^{246.53}$	$2^{237.5}$	Section 7

Organization. Section 2 lists some notations and briefly introduces 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. Based on the results of automatic search, Section 5 evaluates the complete correlation. Section 6 and 7 show the correlation attacks on SNOW-V and SNOW-Vi, respectively. Section 8 concludes this paper.

2 Preliminaries

2.1 Notations and definitions

Henceforth, we fix some notations for convenience.

$GF(2)$	the binary field
$GF(2^n)$	the n-dimensional extension field of $GF(2)$
$\oplus$	the bitwise XOR operation
$\boxplus$	the parallel of four additions modulo 2^{32}

$\boxminus$	the parallel of four subtractions modulo 2^{32}
$\bar{x}$	the NOT operation for a given bit x
$wt(x)$	the Hamming weight of a Boolean vector x

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 $GF(2^{16})$ 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 $x * y = (xy_{m-1}, xy_{m-2}, ..., xy_0)$, where the product xy_i is operated over $GF(2^{16})$.

The correlation of a binary random variable x is defined as

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

and the absolute correlation of the binary random variable x is defined as $|\rho(x)|$.

For a vectorized Boolean function $f : GF(2^m) \to GF(2^n)$, the correlation with the input mask α and the output mask β is calculated as

$$\rho(\alpha \to \beta) = \frac{1}{2^m} \sum_{x \in GF(2^m)} (-1)^{\alpha \cdot x \oplus \beta \cdot f(x)},$$

and we express the approximation process by $\alpha \xrightarrow[\rho(\alpha\to\beta)]{f} \beta$. Especially, the correlation of $f(x, y) = x \boxplus y$ with the input mask α, β and output mask γ is denoted by $\rho_A(\alpha, \beta \to \gamma)$, and the correlation of $f(x) = AES^R(x, 0)$ with input mask α and output mask β is denoted by $\rho_E(\alpha \to \beta)$.

We use the corresponding bold letter to denote the matrix of a linear transformation, e.g., $P(x) = \boldsymbol{P}x$ for a linear transformation P and a column vector x.

2.2 Description of SNOW-V and SNOW-Vi

SNOW-V. Like previous SNOW stream ciphers, SNOW-V consists of LFSR part and FSM part. 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.

The two LFSRs are named LFSR-A and LFSR-B, both with a length of 16 and a cell size of 16 bits. The 32 cells are denoted as $a_{15}, ..., a_0$ and $b_{15}, ..., b_0$ respectively. 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

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

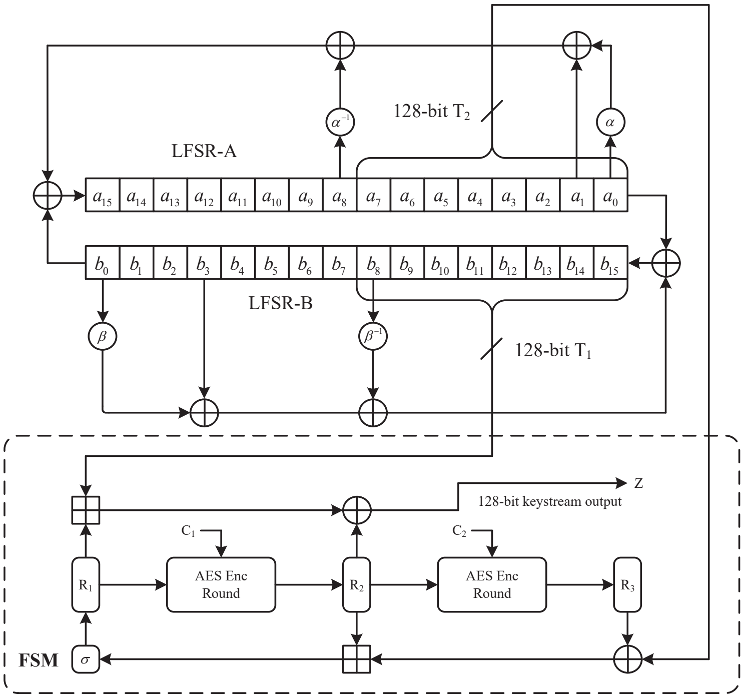
128-bit T₂
α⁻¹
α
LFSR-A
a₁₅ a₁₄ a₁₃ a₁₂ a₁₁ a₁₀ a₉ a₈ a₇ a₆ a₅ a₄ a₃ a₂ a₁ a₀
b₀ b₁ b₂ b₃ b₄ b₅ b₆ b₇ b₈ b₉ b₁₀ b₁₁ b₁₂ b₁₃ b₁₄ b₁₅
LFSR-B
β
β⁻¹
128-bit T₁
Z
128-bit keystream output
C₁
C₂
R₁
AES Enc Round
R₂
AES Enc Round
R₃
FSM
σ

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

The LFSR part is updated by

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

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 clock t are given respectively by

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

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

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

$AES^R(input, key)$ denotes the AES encryption round function, C_1 and C_2 are zeros. 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 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 [4].

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 that the two fields F_2^A and F_2^B have the generating polynomials:

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

and the LFSR part updates as follows:

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

Two taps T_1 and T_2 at clock 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, please refer to [5].

3 Linear approximation of SNOW-V

Our motivation is to find biased binary approximations of SNOW-V which only relate to the output words and LFSR states. For most stream ciphers, how to evaluate the correlations of linear approximations based on consecutive outputs is a difficult problem. In this section, we convert the linear approximations based on three consecutive outputs of SNOW-V into those of a composite function equivalently. Thus we can evaluate the correlations by the properties of Walsh spectrum, and it is much clearer to investigate the linear approximations from the point of view of composite functions.

SNOW-V employs two LFSRs making up a circular structure. There is a straightforward observation [9] that the four taps at three consecutive clocks satisfy

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

and we also confirm it experimentally. The above equation can be rewritten as

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

where L is a linear mapping recording the linear relationship above. We omit the superscript of $R_1^{(t)}, R_2^{(t)}, R_3^{(t)}$, and simplify 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 observe that the following equation will show a nonzero correlation ρ when the masks take certain values:

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

In order to make the linear approximation process more explicit and precise, we divide it 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 composite 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. In the following theorem, we equivalently convert the approximation of FSM part of SNOW-V into that of the composite function F.

Theorem 1. Assume that $R_1, R_2, R_3, T_1^{(t-1)}, T_1^{(t)}, T_1^{(t+1)}$ are independent and uniform distributed. For the binary linear approximation of F

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

which is under the masks defined by the same $\alpha, \beta, \gamma, l, m, n, h$ as in Equation (1), we have $\rho = \rho_F$.

Proof. Set

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

As $R_1, R_2, R_3, T_1^{(t-1)}, T_1^{(t)}, T_1^{(t+1)}$ are independent and uniform distributed, x, y, z, u, v, w are independent and uniform distributed as well. Recall that

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

we have

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

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

$$\begin{aligned}&(\gamma, \beta, l, m, n, \gamma) \cdot (\sigma(R_2 \boxplus (R_3 \oplus T_2^{(t)})) \boxplus T_1^{(t+1)}, R_2, T_1^{(t-1)}, T_1^{(t)}, T_1^{(t+1)}, E(R_1)) \\ &\oplus (\alpha, \alpha, h, \beta) \cdot (E^{-1}(R_3), E^{-1}(R_2) \boxplus T_1^{(t-1)}, T_2^{(t)}, T_1^{(t)} \boxplus R_1) \overset{\rho_F}{=} 0.\end{aligned} \tag{2}$$

As the linear approximation equations (1) and (2) are exactly the same, it is obvious that the correlation ρ_F of the above approximation is equal to ρ. □

By Theorem 1, we convert the problem of computing the correlation of Equation (1) into that of searching for linear approximations of F equivalently. In fact, Theorem 1 also indicates a provable result of the correlations of binary approximations based on three consecutive outputs of SNOW-V. The binary linear approximations of F defined in Theorem 1 by the parameters $\alpha, \beta, \gamma, l, m, n, h$ correspond one-to-one to those of Equation (1), which have the same correlation. Thus, using the properties of Walsh spectrum of composite functions, we can evaluate the approximations of SNOW-V by measuring the linear trails instead of computing the correlations of approximations directly.

Notice that different choices of l, m, n, h lead to different forms of Equation (1), i.e., different distinguishers when $\rho \neq 0$. From the linear relation

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

we know that

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

when $n\boldsymbol{L} = h\boldsymbol{L} = (l||m)$ holds, in which $||$ represents the cascading operation. Then Equation (1) shall become

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

which contains only the output words z_{t-1}, z_t, z_{t+1}, and indicates a distinguisher for distinguishing attack; otherwise, when

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

we will get a distinguisher for correlation attack which can be used to recover the initial state of the LFSR.

In this paper, we focus on the search of distinguishers for correlation attack. Denoting the intermediate masks as a, b, c, d, e, f, q respectively, the linear approximation trail of F above can be described as

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

and its correlation can be evaluated as

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

with the constraint $d\boldsymbol{L} = (e\oplus l)||(f\oplus m)$. This form of approximation enables us to find linear trails with the assistance of automatic search technique. The detailed reasoning process of intermediate masks is given in Appendix A, and there is no need to consider the influence of e and f in later analysis, because they can be generated linearly by d, l, m. According to the properties of Walsh spectrum of composite functions, the accurate correlation of a binary linear approximation with the input and output masks defined by parameters $\alpha, \beta, \gamma, l, m, n, h$ of F should be computed by

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

which means we can get the accurate correlation by exhausting the intermediate masks a, b, c, d, q.

4 Automatic search of linear approximation trails of SNOW-V

In this section, we model the problem of searching linear approximation trails as STP-based automatic search programs. STP is an SMT solver which encodes the constraints with CVC, SMT-LIB1 and SMT-LIB2 languages [12]. STP has been used to analyze block ciphers [14], but for stream ciphers, there is no precedent as far as we know. Since STP solver can model XOR operations easily, 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, and determined in the verification process.

8-bit S-box. We denote $c(x, y)$ the correlation of an S-box with the input mask $x = (x_7, x_6, ..., x_0)$ and output mask $y = (y_7, y_6, ..., y_0)$. Since the nonzero absolute correlations of the S-box except 1 has 8 values, we split the linear correlation table into multiple Boolean functions like in [13]. Here we construct 8 Boolean functions:

$$f_k(x, y) = \begin{cases} 1, & \text{if} \quad |c(x, y)| = 4k/256; \\ 0, & \text{if} \quad |c(x, y)| \neq 4k/256. \end{cases} \quad k = 1, 2, ..., 8$$

As 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, $f_k(x, y)$ can be converted into a series of logical conditions. With the constraint

$$f_1(x, y)|f_2(x, y)|...|f_8(x, y) = x_0|x_1|...|x_7|y_0|y_1|...|y_7$$

added, we have the observation that $f_k(x, y) = 1$ if and only if $|c(x, y)| = 4k/256$. STP solver does not support the floating-point data type, so we replace the absolute correlation of the i-th S-box $|c^{(i)}(x, y)|$ with [14]

$$S^{(i)} = -\left\lfloor 10^t \log_2 |c^{(i)}(x, y)| \right\rfloor = \sum_{k=1}^{8} \left\lfloor 10^t f_k^{(i)}(x, y) \log_2(256/4k) \right\rfloor,$$

in which t is the precision parameter. Thus we get the absolute correlation of an S-box being accurate to t decimal places.

Addition modulo 2^{32}. Wallén [15] proposed a recursive method to efficiently compute the correlation with given input and output masks. Then the result was improved by Schulte-Geers [16]. Denoting the output mask as u, the input masks as v, w, the i-th bit of Boolean vector x as x_i, the constraints to obtain a valid linear approximation shall be expressed as

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

in which z is a dummy variable. The correlation of the linear approximation is not zero if and only if z satisfies the constraints, and is given by $\rho_A(v, w \to u) = (-1)^{(u \oplus v)\cdot(u \oplus w)} 2^{-wt(z)}$ when it is not zero. In order to keep consistent with the accuracy of the correlation of S-boxes, we replace the absolute correlation of the j-th modular addition $2^{-wt(z^{(j)})}$ with $Z^{(j)} = 10^t wt(z^{(j)})$ as well.

Objective function. As there are 48 S-boxes and 16 modular additions taking part in the linear approximation, a trail can be evaluated by $\sum\limits_{i=1}^{48} S^{(i)} + \sum\limits_{j=1}^{16} Z^{(j)}$. A solution returned by the STP solver satisfying the constraint

$$\sum_{i=1}^{48} S^{(i)} + \sum_{j=1}^{16} Z^{(j)} < l$$

represents a trail with the absolute correlation higher than $2^{-10^{-t}l}$.

The sign. After STP solver returns a linear approximation trail that satisfies all constraints, we verify the trail and determine its sign.

Finding more trails. As mentioned in Section 3, 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:

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

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. Labeling different trails with different subscripts, the best result we have found is

$$\begin{aligned} &\alpha_1 = l_1 = c_1 = 0xc, \mathsf{0}, \mathsf{0}, \mathsf{0} \\ &\beta_1 = m_1 = 0x80, \mathsf{0}, \mathsf{0}, \mathsf{0} \\ &\gamma_1 = h_1 = b_1 = 0x81ec5a80, \mathsf{0}, \mathsf{0}, \mathsf{0} \\ &n_1 = 0x81ec5a00, \mathsf{0}, \mathsf{0}, \mathsf{0} \\ &a_1 = 0xc1000000, \mathsf{0}, \mathsf{0}, \mathsf{0} \\ &q_1 = 0xa0, \mathsf{0}, \mathsf{0}, \mathsf{0} \\ &d_1 = \mathsf{0}, \mathsf{0}, \mathsf{0}, \mathsf{0}. \end{aligned}$$

with the correlation 2^{-48} (The symbol '0' denotes 32-bit 0, and the leftmost 32-bit word is the most significant word hereafter), meanwhile we also focus on another trail

$$\begin{aligned}
&\alpha_2 = l_2 = c_2 = 0xd, \mathtt{0}, \mathtt{0}, \mathtt{0}\\
&\beta_2 = m_2 = 0x40, \mathtt{0}, \mathtt{0}, \mathtt{0}\\
&\gamma_2 = h_2 = b_2 = 0x81ec5a80, \mathtt{0}, \mathtt{0}, \mathtt{0}\\
&n_2 = 0x81ec5a00, \mathtt{0}, \mathtt{0}, \mathtt{0}\\
&a_2 = 0xc1000000, \mathtt{0}, \mathtt{0}, \mathtt{0}\\
&q_2 = 0x60, \mathtt{0}, \mathtt{0}, \mathtt{0}\\
&d_2 = \mathtt{0}, \mathtt{0}, \mathtt{0}, \mathtt{0}.
\end{aligned}$$

with the correlation $-2^{-49.063}$.

From the expression of $\rho(a, b, c, d, q)$ in Section 3, we can see that the common features of both trails are that there is only one active S-box in each substitution layer, and only one active 32-bit word in each layer of addition modulo 2^{32}. Besides, we can see that masks of the two trails are similar. In fact, all of the linear approximation trails we found with high correlations have masks of the same form above. Although these trails can approximate the corresponding linear approximations to some extent, we still want to evaluate the accurate correlations.

5 Evaluating the accurate correlations for a special type of binary linear approximations

Based on the trails we have searched out, we try to get accurate values of some high correlations. In this section, indeterminate nonzero 8-bit bytes in 128-bit masks are denoted as $*$. Since all the trails we have searched out with absolute correlation higher than 2^{-50} are of the form

$$\begin{aligned}
&\alpha = l = 0x000000*, \mathtt{0}, \mathtt{0}, \mathtt{0}\\
&\beta = m = 0x000000*, \mathtt{0}, \mathtt{0}, \mathtt{0}\\
&\gamma = h = 0x81ec5a80, \mathtt{0}, \mathtt{0}, \mathtt{0}\\
&n = 0x81ec5a00, \mathtt{0}, \mathtt{0}, \mathtt{0},
\end{aligned}$$

and the accurate correlation of a binary linear approximation should be

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

Thus, we evaluate the accurate correlations of linear trails with parameters

$$\begin{aligned}
&\alpha = l = 0x000000X, \mathtt{0}, \mathtt{0}, \mathtt{0}\\
&\beta = m = 0x000000Y, \mathtt{0}, \mathtt{0}, \mathtt{0}\\
&\gamma = h = 0x81ec5a80, \mathtt{0}, \mathtt{0}, \mathtt{0}\\
&n = 0x81ec5a00, \mathtt{0}, \mathtt{0}, \mathtt{0},
\end{aligned}$$

by exhausting the intermediate masks a, b, c, d, q for given 8-bit words X and Y.

Mask $\boldsymbol{c}$ and $\boldsymbol{q}$. Due to the properties of the linear approximation of modular addition, the most significant nonzero bit of an input mask must be in the same position as that of the output mask. Therefore, there is a straightforward observation that c and e have the form $(0x000000{*}, \mathtt{0}, \mathtt{0}, \mathtt{0})$, i.e., c and e are zeros except for their 12-th bytes, with the assumption $\alpha = (0x000000X, \mathtt{0}, \mathtt{0}, \mathtt{0})$ and $\rho_A(c, e \to \alpha) \neq 0$. In a similar way, we can deduce $q = (0x000000{*}, \mathtt{0}, \mathtt{0}, \mathtt{0})$ by $\beta = (0x000000Y, \mathtt{0}, \mathtt{0}, \mathtt{0})$ and $\rho_A(f, q \to \beta) \neq 0$, and so as f. Hence we only need to exhaust at most 255 values of c and q respectively.

Mask $\boldsymbol{d}$. As l, m, e, f are zeros except for their 12-th bytes, from the linear relation $d\boldsymbol{L} = (e \oplus l)||(f \oplus m)$ we can get

$$d\boldsymbol{L} = (0x000000{*}, \mathtt{0}, \mathtt{0}, \mathtt{0}, 0x000000{*}, \mathtt{0}, \mathtt{0}, \mathtt{0}).$$

This system of linear equations can be confirmed that the unique solution of d is $(\mathtt{0}, \mathtt{0}, \mathtt{0}, \mathtt{0})$. The detailed proof is shown in Appendix B.

Mask $\boldsymbol{b}$ and $\boldsymbol{a}$. By $\rho_E(c \to b) \neq 0$ and $c = (0x000000{*}, \mathtt{0}, \mathtt{0}, \mathtt{0})$, we know that $b\boldsymbol{P} = (0x000000{*}, \mathtt{0}, \mathtt{0}, \mathtt{0})$, where $\boldsymbol{P}$ is the binary matrix of the linear transformation of AES round function. So there are 255 values for b to traverse as well. a is constrained by both $\rho_A(a, n \oplus d \to \gamma) \neq 0$ and $\rho_A(b \oplus \beta, d \oplus h \to \boldsymbol{\sigma}^T a) \neq 0$. The first constraint means that the least significant three 32-bit words of a are zeros while the second indicates that the least significant three 32-bit words of $\boldsymbol{\sigma}^T a$ are zeros. Since the 15-th byte is the unique fixed point of σ among the 4 most significant bytes, we have $a = (0x * 000000, \mathtt{0}, \mathtt{0}, \mathtt{0})$.

Thus, we only need to exhaust 4 bytes to get all the trails with nonzero correlation and reach the accurate correlation of the linear approximation by summing them up when $\alpha, \beta, \gamma, l, m, n, h$ are chosen as above. We could also traverse X and Y to find the optimal approximation of this type. The time complexity is far less than 2^{48}, for most of the cases shall break halfway. Based on the two trails shown in Section 4, we compute the correlations and get

$$c(\alpha_1, \beta_1, \gamma_1, l_1, m_1, n_1, h_1) = 2^{-48.06}, c(\alpha_2, \beta_2, \gamma_2, l_2, m_2, n_2, h_2) = -2^{-47.76}.$$

The second one is the optimal approximation of this type, and we can see an interesting aggregation effect: the absolute correlation of the first approximation is lower than the second while the first trail has a higher absolute correlation. For both approximations, there are more trails with negative correlations in the exhaustion process. So the correlation of the first trail 2^{-48} is offset by a large number of negative correlations, resulting in a lower sum $2^{-48.06}$, while the absolute correlation of the second trail $2^{-49.06}$ increases to $2^{-47.76}$ by summing the correlations up.

The involved codes can be found at: `https://github.com/acaofsas/SNOW-V`.

6 A correlation attack on SNOW-V

Based on the generic method of fast correlation attack given in [17], we present a correlation attack on SNOW-V using the linear approximation with the correlation $c = -2^{-47.76}$ given in Section 5, and give a more detailed analysis of success probability and complexity.

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

We call the state of LFSR that produces 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

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

We assume that $u = (u_{511}, u_{510}, \cdots, u_0)$ and $\hat{u} = (\hat{u}_{511}, \hat{u}_{510}, \cdots, \hat{u}_0)$ are 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 $c = -2^{-47.76}$ when $\hat{u} = u$, otherwise $\phi_t(\hat{u})$ is uniform distributed. With this analysis, we recover the initial LFSR state in two steps.

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

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

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 $p(\phi_{i_1}(u) = 0) = p(\phi_{i_2}(u) = 0) = \frac{1}{2} + \frac{1}{2}c$, we have $p(\phi_{i_1}(u) \oplus \phi_{i_2}(u) = 0) = \frac{1}{2} + \frac{1}{2}c^2$, i.e., $\rho = c^2$. Thus we can get a parity check equation of B bits of the initial state u

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

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

$M = C_N^2 2^{-(512-B)} \approx 2^{-(513-B)} N^2$. Thus 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 $(\alpha, \beta, \gamma) \cdot Z_j \oplus \delta_j \cdot u^h = 0$, 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 $\delta_j = (\Gamma_{i_1}^h \oplus \Gamma_{i_2}^h)$. For each guessed B bits $\hat{u}^h \in \{0,1\}^B$ of the initial state u, we evaluate the parity checks, then get

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

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.

6.2 Success probability and complexity

For linear attacks, we recall the relationship between the probability of success and the number of check equations below:

Definition 1. [18] *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. [18] *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. 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 $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2}} dt$ is the distribution function of the standard normal distribution.

Corollary 1. [18] *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 [19], we use the formula $\varPhi^{-1}(1-\lambda) \overset{\lambda\to 0^+}{\approx} \sqrt{-2\ln\lambda}$ to approximate $\varPhi^{-1}(1-2^{-a-1})$. Hence, we have $M \approx \frac{1}{\rho^2}\left(\varPhi^{-1}(p_s) + \sqrt{2(a+1)\ln 2}\right)^2$ for sufficiently large a.

The complexity can be evaluated as follows. In the preprocessing stage, we evaluate and store each $\varGamma_i \in \{0,1\}^{512}$ for $1 \leq i \leq N$. Then we sort $\varGamma_i$ according to the value of $\varGamma_i^l$ such that $\varGamma_{i_1}^l = \varGamma_{i_2}^l$ 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 computed for each guessed $\hat{u}^h \in \{0,1\}^B$ by evaluating M parity check equations. When $B > \log M$, denoting the most significant $\lceil \log M \rceil$ bits of δ_j and $\hat{u}^h$ as δ_j^{h1} and $\hat{u}^{h1}$, the least significant $B - \lceil \log M \rceil$ bits as δ_j^{h2} and $\hat{u}^{h2}$ respectively, we can accelerate the process using fast Walsh transformation by

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

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

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

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

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 $M \approx 2^{200}$ and $N \approx 2^{237.5}$ under $p_s = 0.999992$ and $B = 238$, which makes the total complexity lowest. The time complexity of the preprocessing stage is $2^{245.40}$, the memory complexity is $2^{237.5}$. In the processing stage the time complexity is $2^{245.65}$, the memory complexity is 2^{238}. Thus, the attack can be done with the total time complexity $2^{246.53}$, memory complexity $2^{238.77}$ and $2^{237.5}$ keystream words given.

After the first 238 bits in the initial state of the LFSR be recovered, one can recover the rest 274 bits of LFSR in the same way. Denoting the most significant 238 bits of binary vector x as $x^{(238)} = (x_{511}, x_{510}, ..., x_{274})$, the following B' bits as $x^{h'} = (x_{273}, ..., x_{274-B'})$ and the least significant $274 - B'$ bits as $x^{l'} = (x_{273-B'}, ..., x_0)$ respectively, for $1 < i_1, i_2 < N'$ we have

$$(\varGamma_{i_1} \oplus \varGamma_{i_2}) \cdot u = (\varGamma_{i_1}^{(238)} \oplus \varGamma_{i_2}^{(238)}) \cdot u^{(238)} \oplus (\varGamma_{i_1}^{(h')} \oplus \varGamma_{i_2}^{(h')}) \cdot u^{(h')} \oplus (\varGamma_{i_1}^{(l')} \oplus \varGamma_{i_2}^{(l')}) \cdot u^{(l')}.$$

As the most significant 238 bits are known, we can get $M' \approx 2^{-(275-B)} N'^2$ parity check equations when $\varGamma_{i_1}^{l'} = \varGamma_{i_2}^{l'}$ holds. By Corollary 1, when $p_s = 0.999992$ and $B' = 199$, we have $M' \approx 2^{200}$ and $N' \approx 2^{138}$, which indicates a much lower time complexity of $2^{206.66}$, and lower data and memory complexities than that of the recovery of the first 238 bits. The remaining 75 bits can be exhausted directly.

It is easy to see that with the recovery of the LFSR state in encryption stage, one can recover the three memories R_1, R_2 and R_3 in encryption stage with a time complexity not more than 2^{128}. In fact, as $z^{(t)} = (R_1^{(t)} \boxplus T_1^{(t)}) \oplus R_2^{(t)}$, the state $R_2^{(t)}$, $R_1^{(t+1)}$, $R_2^{(t+1)}$ and $R_3^{(t+1)}$ shall be recovered as soon as $R_1^{(t)}$ is guessed. Thus the initial states are recovered, i.e., we can predict the keystream word at any clock. There remains an open problem of how to recover the original key of SNOW-V effectively if one has recovered the initial states in encryption stage, which is worth being explored in the future.

7 A correlation attack on SNOW-Vi

Using the same method, in this section we launch a correlation attack on SNOW-Vi in a similar way.

7.1 Linear approximation of SNOW-Vi

In WiSec 2021, Ekdahl et al. [5] 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

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

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

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

and the correlation of a linear trail can be computed as

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

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

7.2 Compared with the linear approximation of SNOW-V

The correlation of a linear trail of SNOW-V is

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

Since $d = 0$, $e = l, f = m$ holds for the type of linear trails in Section 5, the expression can be reduced to

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

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, the linear trail of SNOW-V determined by the same parameters

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

with $d = 0$ has the same correlation as that of SNOW-Vi, i.e.,

$$\rho(a,b,c,0,q) = \rho'(a,b,c,q).$$

Proposition 1 indicates that the linear approximation trails of SNOW-Vi correspond one-to-one to the trails with $d = 0$ of SNOW-V, and the set consisting of all linear trails of SNOW-Vi is a subset of that of SNOW-V. Therefore, the results of SNOW-V in this paper are also appropriate for SNOW-Vi. We could approximate SNOW-Vi with the same correlation $-2^{-47.76}$ of SNOW-V under

$$\begin{aligned}&\alpha = l = 0xd, \mathtt{0}, \mathtt{0}, \mathtt{0}\\ &\beta = m = 0x40, \mathtt{0}, \mathtt{0}, \mathtt{0}\\ &\gamma = h = 0x81ec5a80, \mathtt{0}, \mathtt{0}, \mathtt{0}\\ &n = 0x81ec5a00, \mathtt{0}, \mathtt{0}, \mathtt{0}.\end{aligned}$$

Similarly, the correlation attack presented in Section 5 with time complexity $2^{246.53}$, memory complexity $2^{238.77}$ and $2^{237.5}$ words given is effective for SNOW-Vi as well.

8 Conclusion

In this paper, we study the linear approximation of the nonlinear functions of SNOW-V and SNOW-Vi by the composite function technique. 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 significantly improved. Based on the results searched out, we manage to evaluate the accurate correlation of a special type of binary linear approximations. For SNOW-V, we find a binary linear approximation with correlation $-2^{-47.76}$, substantially improving the results in the design document. Using the linear approximation we launch a correlation attack with a time complexity $2^{246.53}$, a memory complexity $2^{238.77}$ and $2^{237.5}$ keystream 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 when we ignore the design constraint that the maximum of keystream length with a single pair of key and IV is 2^{64}.

Acknowledgements. We thank the anonymous reviewers for their valuable suggestions on how to improve this paper. Also, we thank Alexander Maximov for helpful discussion and verification of our results. This work is supported by National Natural Science Foundation of China (Grant No. 61772547).

References

1. Ekdahl, P., Johansson, T.: SNOW - a new stream cipher. In: Proceedings of First Open NESSIE Workshop, KU-Leuven. pp. 167–168 (2000)
2. Ekdahl, P., Johansson, T.: A new version of the stream cipher SNOW. In: Nyberg, K., Heys, H.M. (eds.) SAC 2002. LNCS, vol. 2595, pp. 47–61. Springer, Heidelberg (2002)
3. ETSI/SAGE.: Specification of the 3GPP confidentiality and integrity algorithms UEA2 & UIA2 (2006)
4. Ekdahl, P., Johansson, T., Maximov, A., Yang, J.: A new SNOW stream cipher called SNOW-V. IACR Trans. Symmetric Cryptol. **2019**(3), 1–42 (2019)
5. Ekdahl, P., Maximov, A., Johansson, T., Yang, J.: SNOW-Vi: an extreme performance variant of SNOW-V for lower grade CPUs. In: Pöpper, C., Vanhoef, M., Batina, L., Mayrhofer, R. (eds.) WiSec 2021. pp. 261–272. ACM, New York (2021)
6. Cid, C., Dodd, M., Murphy, S.: A security evaluation of the SNOW-V stream cipher. 4 June 2020. Quaternion Security Ltd. https://www.3gpp.org/ftp/tsg_sa/WG3_Security/TSGS3_101e/Docs/S3-202852.zip.
7. Jiao, L., Li, Y., Hao, Y.: A guess-and-determine attack on SNOW-V stream cipher. Comput. J. **63**(12), 1789–1812 (2020)
8. Gong, X., Zhang, B.: Resistance of SNOW-V against fast correlation attacks. IACR Trans. Symmetric Cryptol. **2021**(1), 378–410 (2021)
9. Yang, J., Johansson, T., Maximov, A. (2021). Improved guess-and-determine and distinguishing attacks on SNOW-V. IACR Trans. Symmetric Cryptol. **2021**(3), 54–83 (2021)
10. Hoki, J., Isobe, T., Ito, R., Liu, F., Sakamoto, K.: Distinguishing and key recovery attacks on the reduced-round SNOW-V and SNOW-Vi. Cryptology ePrint Archive, Report 2021/546 (2021)
11. Shi, D., Sun, S., Sasaki, Y., Li, C., Hu, L.: Correlation of quadratic boolean functions: Cryptanalysis of all versions of full MORUS. In: Boldyreva, A., Micciancio, D. (eds.) Advances in Cryptology - CRYPTO 2019 - 39th Annual International

Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II. Lecture Notes in Computer Science, vol. 11693, pp. 180–209. Springer (2019).

12. Ganesh, V., Hansen, T., Soos, M., Liew, D., Govostes, R.: STP constraint solver (2014). https://github.com/stp/stp.
13. Abdelkhalek, A., Sasaki, Y., Todo, Y., Tolba, M., Youssef, A.M.: MILP modeling for (large) S-boxes to optimize probability of differential characteristics. IACR Trans. Symmetric Cryptol. **2017**(4), 99–129 (2017)
14. Liu, Y., Liang, H., Li, M., Huang, L., Hu, K., Yang, C., Wang, M.: STP models of optimal differential and linear trail for S-box based ciphers. Sci. China Inf. Sci. **64**(5) (2021)
15. Wallén, J.: Linear approximations of addition modulo 2^n. In: Johansson, T. (ed.) FSE 2003. LNCS, vol. 2887, pp. 261–273. Springer, Heidelberg (2003)
16. Schulte-Geers, E.: On CCZ-equivalence of addition mod 2^n. Des. Codes Cryptogr. **66**, 111–127 (2013)
17. 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.) CRYPTO 2015. LNCS, vol. 9215, pp. 643–662. Springer, Heidelberg (2015)
18. Selçuk, A.A.: On probability of success in linear and differential cryptanalysis. J. Cryptol. **21**(1), 131–147 (2008)
19. Blondeau, C., Gérard, B., Tillich, J.P.: Accurate estimates of the data complexity and success probability for various cryptanalyses. Des. Codes Cryptogr. **59**, 3–34 (2011)

A Detailed reasoning process of intermediate masks

Based on the linear approximation of F in Section 3, we analyze the intermediate masks 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 ξ_j^i the mask of the j-th output of f_i and ρ_i the correlation of f_i. Then the linear approximation equation of f_1 is

$$\begin{aligned}&\gamma \cdot x \oplus \beta \cdot y \oplus l \cdot z \oplus m \cdot u \oplus n \cdot v \oplus \gamma \cdot w,\\ &\overset{\rho_1}{=} \xi_1^1 \cdot (x \boxminus v) \oplus \xi_2^1 \cdot y \oplus \xi_3^1 \cdot z \oplus \xi_4^1 \cdot u \oplus \xi_5^1 \cdot L(z,u) \oplus \xi_5^1 \cdot v \oplus \xi_6^1 \cdot w,\end{aligned} \tag{3}$$

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 $d\boldsymbol{L} = (e \oplus l)||(f \oplus m)$ by

$$d \cdot L(z,u) = d\boldsymbol{L}(z||u)^T = (e \oplus l) \cdot z \oplus (f \oplus m) \cdot u,$$

and (3) 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 $a, n \oplus d \to \gamma$ of the addition modulo 2^{32}. Thus the correlation of (3) is $\rho_1 = \rho_A(a, n \oplus d \to \gamma)$.

For f_2, we have

$$\begin{aligned}&a \cdot x \oplus \beta \cdot y \oplus e \cdot z \oplus f \cdot u \oplus d \cdot v \oplus \gamma \cdot w \\ =\, &\xi_1^2 \cdot (\sigma^{-1}(x) \boxminus y) \oplus \xi_1^2 \cdot v \oplus \xi_2^2 \cdot y \oplus \xi_3^2 \cdot z \oplus \xi_4^2 \cdot u \oplus \xi_5^2 \cdot v \oplus \xi_6^2 \cdot w,\end{aligned} \tag{4}$$

which is equivalent to

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

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 (4) 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

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

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

For f_3, the following equation holds

$$\begin{aligned}&(d \oplus \xi_5^2) \cdot x \oplus b \cdot y \oplus e \cdot z \oplus f \cdot u \oplus \xi_5^2 \cdot v \oplus \gamma \cdot w \\ \overset{\rho_3}{=}\, &\xi_1^3 \cdot E^{-1}(x) \oplus \xi_2^3 \cdot E^{-1}(y) \oplus \xi_3^3 \cdot z \oplus \xi_4^3 \cdot u \oplus \xi_5^3 \cdot v \oplus \xi_6^3 \cdot w.\end{aligned} \tag{5}$$

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 (5) 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 $\xi_1^3 \overset{AES}{\to} d \oplus \xi_5^2$ and $c \overset{AES}{\to} b$ of AES round function, so we have $\rho_3 = \rho_E(\xi_1^3 \to d \oplus \xi_5^2)\rho_E(c \to b)$.

For f_4, we have

$$\xi_1^3 \cdot x \oplus c \cdot y \oplus e \cdot z \oplus f \cdot u \oplus \xi_5^2 \cdot v \oplus \gamma \cdot w \overset{\rho_4}{=} \xi_1^4 \cdot x \oplus \xi_2^4 \cdot (y \boxplus z) \oplus \xi_3^4 \cdot u \oplus \xi_4^4 \cdot v \oplus \xi_5^4 \cdot w, \tag{6}$$

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 $c, e \to \xi_2^4$ of addition modulo 2^{32}. Obviously $\rho_4 = \rho_A(c, e \to \xi_2^4)$.

For f_5, the approximation equation is

$$\xi_1^3 \cdot x \oplus \xi_2^4 \cdot y \oplus f \cdot z \oplus \xi_5^2 \cdot u \oplus \gamma \cdot v \overset{\rho_5}{=} \xi_1^5 \cdot x \oplus \xi_2^5 \cdot y \oplus \xi_3^5 \cdot z \oplus \xi_4^5 \cdot u \oplus \xi_5^5 \cdot E^{-1}(v), \tag{7}$$

which is equivalent to

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

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 (7) can be reduced to $q \cdot E^{-1}(v) \oplus \gamma \cdot v \overset{\rho_5}{=} 0$, which is the linear approximation $q \overset{AES}{\to} \gamma$ of AES round function, so $\rho_5 = \rho_E(q \to \gamma)$.

For f_6, we have

$$\xi_1^3 \cdot x \oplus \xi_2^4 \cdot y \oplus f \cdot z \oplus \xi_5^2 \cdot u \oplus q \cdot v \overset{\rho_6}{=} \alpha \cdot x \oplus \alpha \cdot y \oplus h \cdot u \oplus \beta \cdot (z \boxplus v). \qquad (8)$$

By $\rho_6 \neq 0$ we know that $\xi_1^3 = \xi_2^4 = \alpha, \xi_5^2 = h$, and (8) can be simplified to $\beta \cdot (z \boxplus v) \oplus f \cdot z \oplus q \cdot v \overset{\rho_6}{=} 0$, which is the linear approximation $f, q \to \beta$ of addition modulo 2^{32}. So we have $\rho_6 = \rho_A(f, q \to \beta)$.

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

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

B The proof of $\boldsymbol{d} = (0, 0, 0, 0)$ under $\boldsymbol{dL} = (\mathbf{0}\boldsymbol{x}\mathbf{000000}*, 0, 0, 0, \mathbf{0}\boldsymbol{x}\mathbf{000000}*, 0, 0, 0)$

Here we denote 128-bit vector $d = (d_7, d_6, ..., d_0)$ in which $d_i \in GF(2^{16})$, and $\boldsymbol{\beta}$ the binary matrix form of $\beta \in GF(2^{16})$. By the relation

$$L(x, y) = \beta * x \oplus \beta^{-1} * y \oplus (x >> 48) \oplus (y << 80),$$

we have

$$\begin{aligned} d \cdot L(x, y) &= d \cdot \beta * x \oplus d \cdot \beta^{-1} * y \oplus d \cdot (x >> 48) \oplus d \cdot (y << 80) \\ &= d \cdot \beta * x \oplus d \cdot \beta^{-1} * y \oplus [(d << 48) \cdot x] \oplus [(d >> 80) \cdot y], \end{aligned}$$

which is equivalent to

$$\begin{aligned} d\boldsymbol{L} = (&d_7\boldsymbol{\beta} \oplus d_4, ..., d_3\boldsymbol{\beta} \oplus d_0, d_2\boldsymbol{\beta}, ..., d_0\boldsymbol{\beta}, \\ &d_7\boldsymbol{\beta}^{-1}, ..., d_3\boldsymbol{\beta}^{-1}, d_2\boldsymbol{\beta}^{-1} \oplus d_7, ..., d_0\boldsymbol{\beta}^{-1} \oplus d_5). \end{aligned}$$

Recall

$$d\boldsymbol{L} = (0x000000*, 0, 0, 0, 0x000000*, 0, 0, 0),$$

we can observe that

$$d_2\boldsymbol{\beta} = d_1\boldsymbol{\beta} = d_0\boldsymbol{\beta} = d_7\boldsymbol{\beta}^{-1} = d_5\boldsymbol{\beta}^{-1} = d_4\boldsymbol{\beta}^{-1} = d_3\boldsymbol{\beta}^{-1} = d_1\boldsymbol{\beta}^{-1} \oplus d_6 = 0.$$

As $\boldsymbol{\beta}$ is invertible, we can get $d = (0, 0, 0, 0)$.