Fast Correlation Attacks on Grain-like Small State Stream Ciphers and Cryptanalysis of Plantlet, Fruit-v2 and Fruit-80

Shichang Wang[1,2], Meicheng Liu[1(✉)], Dongdai Lin[1], and Li Ma[1,2]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
{wangshichang,liumeicheng,ddlin,mali}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

Abstract. The fast correlation attack (FCA) is one of the most important cryptanalytic techniques against LFSR-based stream ciphers. In CRYPTO 2018, Todo et al. found a new property for the FCA and proposed a novel algorithm which was successfully applied to the Grain family of stream ciphers. Nevertheless, these techniques can not be directly applied to Grain-like small state stream ciphers with keyed update, such as Plantlet, Fruit-v2, and Fruit80. In this paper, we study the security of Grain-like small state stream ciphers by the fast correlation attack. We first observe that the number of required parity-check equations can be reduced when there are multiple different parity-check equations. With exploiting the Skellam distribution, we introduce a sufficient condition to identify the correct LFSR initial state and derive a new relationship between the number and bias of the required parity-check equations. Then a modified algorithm is presented based on this new relationship, which can recover the LFSR initial state no matter what the round key bits are. Under the condition that the LFSR initial state is known, an algorithm is given against the degraded system and to recover the NFSR state at some time instant, along with the round key bits.
As cases study, we apply our cryptanalytic techniques to Plantlet, Fruit-v2 and Fruit-80. As a result, for Plantlet our attack takes $2^{73.75}$ time complexity and $2^{73.04}$ keystream bits to recover the full 80-bit key. Regarding Fruit-v2, $2^{55.33}$ time complexity and $2^{55.59}$ keystream bits are token to determine the secret key. As for Fruit-80, $2^{64.46}$ time complexity and $2^{62.79}$ keystream bits are required to recover the secret key. Moreover, we have implemented our attack methods on a toy version of Fruit-v2. The attack matches the expected complexities predicted by our theoretical analysis quite well, which proves the validity of our cryptanalytic techniques.

Keywords: Fast correlation attack · Stream cipher · Grain-like · Plantlet · Fruit-v2 · Fruit-80.

1 Introduction

Stream ciphers play an important role in symmetric-key cryptosystems. Commonly, they are used to generate a keystream of arbitrary length from a secret key and initialization vector (IV). There are many well-known stream ciphers: such as Grain-v1 [18], Trivium [8] both in the eSTREAM portfolio of hardware category, and Grain-128a [1] standardized by ISO/IEC. Common to these stream ciphers is that they have an internal state length of at least twice the size of the security margin to thwart time-memory-data tradeoff (TMDTO) attacks [7].

A new line of research emerged with publication of Sprout [2], which reduces the size of internal state of lightweight stream ciphers below the boundary induced by TMDTO attacks. Sprout has a Grain-like structure and uses two 40-bit feedback shift registers (FSR). In comparison to traditional stream ciphers, Sprout uses the 80-bit key not only for initializing internal state during the initialization phase but also in the state update function of the non-linear feedback shift register (NFSR) during the subsequent keystream generation phase. Unfortunately, Sprout was broken [21] shortly after it was proposed and some more analysis against Sprout were given in [30, 3, 14]. However, an increasing number of researchers' interest is sparked in the underlying design principle of Sprout. So far, there are several Grain-like small state stream ciphers, e.g., Plantlet [25], Fruit [28, 15], Lizard [16], which are designed by following the above essential ideas. Since the designers of Fruit have changed the specification of their stream ciphers several times [28], we will follow the versioning scheme suggested in [17]. Due to the pseudo-linearity property of the weak output function, Fruit-v0 was broken by the fast correlation attack (FCA) in [31]. Fruit-v1 is tweaked to remove the vulnerability. However, there was a weak key attack against Fruit-v1 based on an insecure choice of the round key function in [17]. A more recent version of Fruit, Fruit-v2 named by the designers [28], is proposed to implement fixes for the discovered vulnerabilities. The lack of a well-understood theoretical work in this design paradigm domain apparently restricts the confidence that people have on such primitives. This motivates us to study the security of these Grain-like small state stream ciphers against the well-tailored attacks for them.

In this paper, we study the security of these Grain-like small state stream ciphers by the fast correlation attack, which is one of the most important cryptanalytic techniques against linear feedback shift register (LFSR)-based ciphers. The initial idea of correlation attack was introduced by [26], and it exploited the bias between sequences of the LFSR and keystream. If we guess the correct LFSR initial state, the high bias is observed. Otherwise, we assume that the statistic of bias behaves at random. The simple correlation attack takes a time complexity of $N2^n$, where N is the length of keystream sequence and n is the size of the LFSR. Following up the correlation attack, many algorithms called as the fast correlation attacks have been proposed to avoid the exhaustive search of the LFSR initial state by using parity-check equations. The fast correlation attack algorithms are further divided into iterative algorithms and one-pass algorithms. In iterative algorithms, starting from the keystream sequence, the parity-check equations are used to modify the value of keystream bits in order to converge towards the LFSR sequence, and recover the LFSR initial state [23, 19, 9]. But they have requirements such as the number of taps in the LFSR is significantly small or the bias of parity-check equations is significantly high. Therefore, their applications are limited to experimental ciphers and have not applied to modern concrete stream ciphers. Regarding one-pass algorithms, the evaluation of parity-check equations enable us to directly compute the correct value of the LFSR state [10, 20, 24], and they have been successfully applied to modern concrete stream ciphers [6, 31, 27]. To avoid the exhaustive search of the LFSR initial state, several methods have been proposed to decrease the number of unknown bits in the LFSR initial state involved by the parity-check equations [11, 29]. Moreover, as showed in [11] the fast Walsh-Hadamard transform (FWHT) can be applied to accelerate the one-pass algorithms when the guess and evaluation procedure is regarded as a Walsh-Hadamard transform. Very recently, Todo et al. [27] found that the "commutative" feature of multiplication between $n \times n$ matrices and an n-bit fixed vector, which is generally used to construct parity-check equations. With the new property, the traditional wrong-initial-state hypothesis does not hold assuming there are multiple high-biasd linear masks. Therefore, they introduced a modified wrong-initial-state hypothesis. In previous fast correlation attacks, the multiple linear approximate equations are only useful to decrease the data complexity but not for the time complexity [6, 31]. Using the new wrong-initial-state hypothesis, they proposed a new FCA algorithm where multiple linear approximate equations can reduce both time and data complexities.

1.1 Our Contribution

Inspired by the new FCA algorithm exploiting new property against the Grain family of stream ciphers when there are multiple linear masks [27], we derive a new relationship on the number and bias of required parity-check equations, then present a modified FCA algorithm on Grain-like small state stream ciphers. Under the condition that the LFSR initial state is known, we consider the degraded system and give an algorithm to recover the NFSR state at some time instant, along with the round key bits.

- In traditional fast correlation attacks, the number of required parity-check equations is $\Omega = \frac{4m \ln 2}{(\epsilon^c)^2}$ to identify the unique correct LFSR initial state [31, 6, 10], where m is the size of the LFSR state and ϵ^c is twice as many as the bias of parity-check equations. Since the size of the LFSR is always much small in Grain-like small state stream ciphers, no valid attack against them can be obtained by using directly Proposition 1 proposed in [27]. We first observe that the number of required parity-check equations can be reduced when there are multiple different parity-check equations. With exploiting the Skellam distribution, we introduce a sufficient condition to identify the unique correct LFSR initial state and derive a new relationship between the number and bias of required parity-check equations as $\Omega = \frac{2^{\frac{7}{2}}\sqrt{m \ln 2}}{\sqrt{r}(\epsilon^c)^2}$, where r is the number of different parity-check equations. From the new relationship, we can use fewer parity-check equations, about $\frac{1}{\sqrt{r}}$ times, to identify the correct LFSR initial state when there are r different parity-check equations.
- With the periodic property of the round key function $RKF(\cdot)$, we sample the parity-check equations at a time interval equal to the period of the round key bits to reduce the dimension of unknown variables from the secret key. Then we adjust the original algorithm proposed in [27] to make two

majority polls and the new algorithm (Algorithm 1) can recover the LFSR initial state no matter what the round key bits are.

- We consider the degraded system assuming that the LFSR initial state is known and conclude that the degraded system is feasible to be attacked. Since the size of the NFSR state is always much smaller than the security margin, the exhaustive search of all the possible value of the NFSR state is often feasible. With the periodic property of the $RKF(\cdot)$ and the technique described in Section 3.3, we present an algorithm (Algorithm 2) which calls the state checking subroutine to recover the NFSR state at some time instant, along with the round key bits.

Applications We apply our new relationship and our attack algorithms (Algorithm 1 and 2) to the Grain-like small state stream ciphers, Plantlet [25], Fruit-v2 [28] and Fruit-80 [15]. As a result, for Plantlet our attack takes $2^{73.75}$ time complexity and $2^{73.04}$ keystream bits to recover the full 80-bit key. Regarding Fruit-v2, our attack takes $2^{55.33}$ time complexity and $2^{55.59}$ keystream bits to determine the secret key. As for Fruit-80, $2^{64.46}$ time complexity and $2^{62.79}$ keystream bits are required to recover the secret key. The data complexity of these attacks can be cut down at cost of increasing attack time. The results are listed in Table 1.

Comparisons with Previous Results Next, we compare results of our algorithms with previous attacks against Plantlet, Fruit-v2 and Fruit-80, and they are summarized in Table 1. The time complexity of our attacks is measured by multiplication of matrices with dimension equal to the size m of the LFSR, while the others are measured by cipher encryption. The former is about equivalent to updating the LFSR for m times, and thus it much faster than the latter.

Table 1. Summary of attacks on Plantlet, Fruit-v2 and Fruit-80.

Stream cipher	Type of attack	Time	Memory	Data	Reference
Plantlet	distinguishing attack	2^{55}	2^{61}	2^{61}	[17]
	key recovery attack	$2^{76.26}$	$2^{31.25}$	$2^{84.6\dagger}$	[4]
	key recovery attack	$2^{73.75}$	2^{45}	$2^{73.04}$	Sect. 4.3
	key recovery attack	$2^{79.74}$	2^{51}	$2^{67.04}$	Sect. 4.3
Fruit-v2	key recovery attack	$2^{76.67}$	–	–	[12]
	key recovery attack	$2^{55.33}$	2^{31}	$2^{55.59}$	Sect. 5.3
	key recovery attack	$2^{67.00}$	2^{43}	$2^{43.59}$	Sect. 5.3
Fruit-80	key recovery attack	$2^{64.46}$	2^{37}	$2^{62.79}$	Sect. 6.2
	key recovery attack	$2^{69.99}$	2^{43}	$2^{56.79}$	Sect. 6.2

† It requires $2^{54.6}$ IVs with 2^{30} keystream bits for each IV, totally $2^{84.6}$ keystream bits.

Plantlet is a stronger version of Sprout [2] and some modifications are introduced in order to account for attacks which have been discovered against Sprout [21, 30, 3, 14]. More precisely, the LFSR's size is increased from 40 bits to 61 bits and the round key function is a linear function such that sequentially using one key bit at per clock. Before this paper, there is a distinguishing attack based on TMDTO against Plantlet in [17], which takes 2^{55} time complexity, 2^{61} data complexity and 2^{61} memory complexity. In a distinguishing attack, the algorithm (or distinguisher) allows to distinguish the keystream produced by the target cipher from a random bitstream with high probability, but no information of the secret key can be obtained.

In parallel and independently with our work, Banik et al. [4] presented a key recovery attack on Plantlet with time complexity of $2^{76.26}$ Plantlet encryption, data complexity of $2^{84.6}$ keystream bits, and memory of $2^{31.25}$ bits. More exactly, for the data complexity it requires $2^{54.6}$ IVs, with 2^{30} keystream bits for each IV.

A more recent version of Fruit, Fruit-v2 [28], is proposed to implement fixes for the discovered vulnerabilities, which are found in Fruit-0 [31, 13] and Fruit-v1 [17]. Since the designer of Fruit removed the pseudo-linearity property of the filtering function from Fruit-v2, the previous fast correlation attack methods in [31] are not applicable on Fruit-v2. Moreover, the key taps of the round key function in Fruit-v2 were changed, and the same set of keys found in [17] are not weak any more. Nevertheless, our modified fast correlation attack algorithms can break Fruit-2 thanks to our new observations and techniques.

The divide-and-conquer method has been an important attack against the different versions of Fruit, with exploiting the bias of the round key bits on the NFSR update function in [13, 12]. Especially and very recently, Dey et al. in [12] give a attack against Fruit-v2, where the authors claim that the time complexity is $2^{76.67}$ Fruit encryption. Note that the unit of the time complexity is "1 Fruit encryption", and every Fruit encryption contains 210 rounds of the stream cipher initial clock. The time complexity of our fast correlation attack is $2^{57.06}$, where the unit of the time complexity is at most one multiplication with fixed matrices whose dimension is equal to the size of the LFSR, which is more efficient than the unit given by the initialization of stream ciphers. Our attack is more than 2^{19} times faster than the attack given in [12], but it requires more data than their work.

Fruit-80 is a final version of Fruit stream cipher, which is formally published in [15]. The most significant difference between Fruit-80 and the previous version is that the key bits are involving directly in producing every bit of the keystream bit. As far as we know, there is no key recovery attack reported on Fruit-80 in the literatures.

1.2 Paper Organization

This paper is organized as follows. In Section 2, we present a generic model of Grain-like small state stream ciphers and review the new property of the LFSR-based stream ciphers which was found in [27]. In Section 3, the divide-and-conquer fast correlation attacks are given against the generic model. First, we show how to derive the desirable parity-check equations and modify the original algorithm in [27] to recover the LFSR initial state no matter what the round key bits are in Section 3.1. In Section 3.2, we derive the new relationship between the bias and number of required parity-check equations. Under the LFSR initial state is known, an algorithm is proposed to recover the NFSR state at some time instant and the round key bits in Section 3.3. In the subsequent subsection, we give an analysis for complexities of our attacks against the generic model. As applications, we carry out our attack methods against Plantlet, Fruit-v2 and Fruit-80 in Section 4, Section 5 and Section 6, respectively. Finally in Section 7, a practical experiment is presented on a toy version of Fruit-v2.

2 Preliminaries

In this section, we give a generalized model of Grain-like small state stream ciphers and review the new feature for the fast correlation attacks found in [27].

2.1 The Generalized Model of Grain-like Small State Stream Ciphers

Abstracting from the primitives such as Sprout, Fruit and Plantlet, we present the generalized model for Grain-like small state stream ciphers as depicted in Fig. 1. In this unified framework, some properties from Grain-like small state stream ciphers are discussed and our cryptanalytic techniques against them will be presented in the subsequent section. The generic model is specified by the following items in the keystream generation phase.

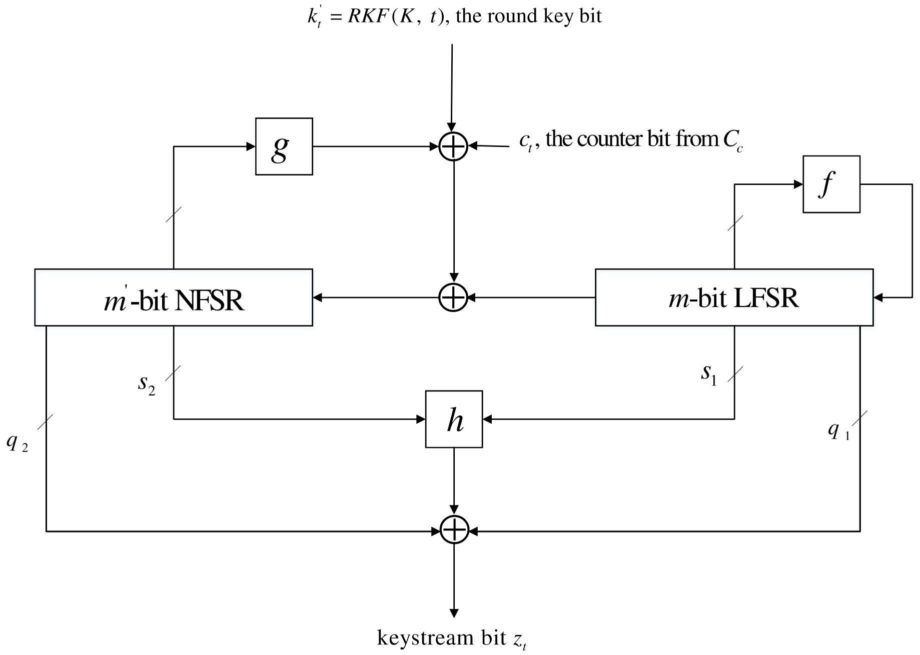
Components **LFSR:** Let m be the size of linear feedback shift register (LFSR) and $L^{(t)} = (l_t, \cdots, l_{t+m-1})$ be the internal state of the LFSR at time instant t. The LFSR is updated recursively and independently by a linear Boolean function f as $L^{(t+1)} = (l_{t+1}, \cdots, l_{t+m})$ with $l_{t+m} = f(L^{(t)})$. We assume this update process is invertible, and the inverse process is $L^{(t-1)} = (l_{t-1}, \cdots, l_{t+m-2})$ with $l_{t-1} = f'(L^{(t)})$.

NFSR and Counter: Let m' be the size of non-linear feedback shift register (NFSR) and $N^{(t)} = (n_t, \cdots, n_{t+m'-1})$ be the internal state of the NFSR at time instant t. The NFSR is updated recursively as defined in the following:

$$n_{t+m'} = k'_t \oplus c_t \oplus l_t \oplus g(N^{(t)}), \tag{1}$$

$$N^{(t+1)} = (n_{t+1}, \cdots, n_{t+m'}),$$

Fig. 1. The generic model for the Grain-like small state stream ciphers



where k_t' is the round key bit at time instant t, c_t is the counter bit from the counter C_c at time instant t, l_t is the output of the LFSR at time instant t and $g(\cdot)$ is a non-linear Boolean function. The round key bit is generated by the round key function, which is explained below. Similarly, we assume that this update process of the NFSR is invertible, and the inverse process is $N^{(t-1)} = (n_{t-1}, \cdots, l_{n+m'-2})$ with $n_{t-1} = k_{t-1}' \oplus c_{t-1} \oplus l_{t-1} \oplus g'(N^{(t)})$.

The counter C_c is a counter register whose initial value and way of working are public.

Round Key Function: The round key function denoted by $RKF(\cdot)$ continuously generates the round key bit which is provided as input to the update function of the NFSR. Namely, $k_t' = RKF(K, t)$, where $K = (k_0, \cdots, k_{\kappa-1})$ is the κ-bit secret key and κ is the security margin.

Output Function: The output function is determined by

$$z_t = h\left(L^{(t)}_{\mathbb{T}_{h,L}}, N^{(t)}_{\mathbb{T}_{h,N}}\right) \oplus \bigoplus_{b_1 \in \mathbb{B}_1} l_{t+b_1} \oplus \bigoplus_{b_2 \in \mathbb{B}_2} n_{t+b_2}, \tag{2}$$

where h is a non-linear filtering function, $L^{(t)}_{\mathbb{T}_{h,L}} = (l_{t+\gamma_1}, \cdots, l_{t+\gamma_{s_1}})$ is a subset of $L^{(t)}$ and the input variables of h from the LFSR with $0 \le \gamma_1 < \cdots < \gamma_{s_1} \le m-1$, $N^{(t)}_{\mathbb{T}_{h,N}} = (n_{t+\delta_1}, \cdots, n_{t+\delta_{s_2}})$ is a subset of $N^{(t)}$ and the input variables of h from the NFSR with $0 \le \delta_1 < \cdots < \delta_{s_2} \le m'-1$, $\mathbb{B}_1 = \{\sigma_1, \cdots, \sigma_{q_1}\}$ and $\mathbb{B}_2 = \{\eta_1, \cdots, \eta_{q_2}\}$ are the sets of the LFSR and NFSR taps respectively, with $0 \le \sigma_1 < \cdots < \sigma_{q_1} \le m-1$ and $0 \le \eta_1 < \cdots < \eta_{q_2} \le m'-1$.

Assumed Properties We assume that the generic model has the following two properties which are exploited by our attack methods in the subsequent section.

1. Assuming that the $RKF(\cdot)$ is periodic, so are the round key bits. Let d be the least positive integer such that $k_{t+d}' = k_t'$ for any $t \ge 0$, i.e., the round key bits repeat in a cycle of length d. Besides, our generic model can also cover the case where the counter bits c_t are unknown. In this case, we just assume that c_t is also periodic.
2. Assuming that the necessary condition holds for applying successfully Algorithm 1 of our attack methods, i.e.,

$$\begin{aligned}\sqrt{m \ln 2} \le &2^{\kappa - \frac{7}{2} + \frac{1}{2}s_1 \times |\mathbb{T}_z| + 2((s_2+1)\times|\mathbb{T}_z| + q_2)} \\ &\times \epsilon_h^{2|\mathbb{T}_z|} \times \epsilon_{g^*}^{2q_2},\end{aligned}$$

where $\mathbb{T}_z$ is a tap set of keystream bits which is needed to be determined later and $|\mathbb{T}_z|$ is the number of elements in the set, s_1 and s_2 are the number of input variables of h from the LFSR and NFSR respectively, q_2 is the number of the NFSR masking variables of the output function, ϵ_h and ϵ_{g^*} are the biases of the linear approximations for the functions h and g respectively.

Our generalized model can cover Plantlet [25], Fruit-v2 [28] and Fruit-80 [15], but not Lizard [16]. Compared with the generic model, the difference of Fruit-80 is that the key bits are involved directly in the output function. However, this would have no impact on applying our fast correlation attack algorithms to Fruit-80. Regarding Lizard, there is no LFSR and two NFSRs of different sizes are used instead.

2.2 LFSR-Based Stream Ciphers

In this subsection, we review the new feature found in [27] which is directly useful to improve the efficiency of the fast correlation attacks.

The target of the fast correlation attacks is the LFSR-based stream ciphers, which include the generic model of Grain-like small state stream ciphers as a special case. Let the primitive polynomial

$$f(x) = c_0 + c_1x^1 + c_2x^2 + \cdots + c_{m-1}x^{m-1} + x^m$$

be the feedback polynomial of the LFSR and $L^{(t)} = (l_t, \cdots, l_{t+m-1})$ be the m-bit internal state of the LFSR at time instant t. Then, the LFSR outputs l_t and the state is updated to $L^{(t+1)}$ as

$$L^{(t+1)} = L^{(t)} \times F = L^{(t)} \times \begin{bmatrix} 0 & \cdots & 0 & 0 & c_0 \\ 1 & \cdots & 0 & 0 & c_1 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 1 & 0 & c_{m-2} \\ 0 & \cdots & 0 & 1 & c_{m-1} \end{bmatrix},$$

where F is the state transition matrix of the LFSR, the operator × represents the matrix multiplication and here is multiplication between $1 \times m$ matrix and $m \times m$ matrix. Furthermore, any internal state of the LFSR can be expressed by the initial state and the state transition matrix as

$$L^{(t)} = L^{(0)} \times F^t \quad \forall t \geq 0, \tag{3}$$

where F^t is the t-th power of F.

Theorem 1 (New Feature [27]). *Let F be the state transition matrix of the LFSR whose feedback polynomial is the primitive polynomial $f(x) = c_0 + c_1x^1 + \cdots + c_{m-1}x^{m-1} + x^m$ and $\boldsymbol{u}$ is an m-bit column vector, i.e.,*

$$F = \begin{bmatrix} 0 & \cdots & 0 & 0 & c_0 \\ 1 & \cdots & 0 & 0 & c_1 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 1 & 0 & c_{m-2} \\ 0 & \cdots & 0 & 1 & c_{m-1} \end{bmatrix}.$$

Then $F^t \times \boldsymbol{u} = F_{\boldsymbol{u}} \times \boldsymbol{g}_t$, where F^t is the t-th power of F, $\boldsymbol{g}_t$ is the first column of matrix F^t and

$$F_{\boldsymbol{u}} = \left[\boldsymbol{u}, F^1 \times \boldsymbol{u}, \cdots, F^{m-1} \times \boldsymbol{u}\right]$$

Remark 1. Note that the notation $\boldsymbol{g}_t$ is defined as the 1-st column vector of F^t, and then the i-th column vector of F^t is represented as $\boldsymbol{g}_{t+i-1}$, $1 \leq i \leq n$.

In [27], Todo et al. give the proof of the above theorem using a finite field $GF(2^m)$, where the primitive polynomial is the feedback polynomial of the LFSR.

3 Divide-and-Conquer Fast Correlation Attacks

Here, we present a high-level review of our divide-and-conquer fast correlation attacks against the generalized model of Grain-like small state stream ciphers. Since the LFSR updates independently, we first recover the LFSR initial state through the fast correlation attacks exploiting multiple linear masks. To reduce the dimension of unknown variables from the secret key, the parity-check equations are sampled at a time interval equal to the period of the round key bits. Then we modify the original algorithm proposed in [27] to make two majority polls and the new algorithm (Algorithm 1) can recover the LFSR initial state no matter what the round key bits are. The details of this procedure will be described in Section 3.1. No valid attack against Grain-like small state stream ciphers can be obtained by using directly Proposition 1 proposed in [27], because the size of the LFSR is always much small in these small state ciphers. We first observe that the number of required parity-check equations can be reduced when there are multiple different parity-check equations. With exploiting the Skellam distribution, we derive a new relationship between the number and bias of required parity-check equations for applying successfully Algorithm 1 to recover the correct value of the LFSR initial state in Section 3.2. Once the initial state of the LFSR is determined, there is not protection of the internal state variables of the LFSR in the keystream bits. We get a degraded system of equations on output keystream bits and the internal state variables of the NFSR. We can relate these internal state variables by the update function of the NFSR involving the round key bits. This leads to new increasing unknown internal state variables. We propose instead to use the non-linear filtering function to derive relations between these variables, inspired by the observation in [5]. Compared to their technique, we do not require the property of linearity of the filtering function or the property of pseudo-linearity which is used to break Fruit-v0 in [31]. Further, we can run the update function of the NFSR forwards to obtain the round key bits from the internal state variables of the NFSR. Due to that the round key bits are periodic, we can carry out a state checking procedure where we compare the round key bits of two different repetition cycles. Since the size of the NFSR in Grain-like small state stream ciphers is always much smaller than the security margin, exhaustively searching all the possible value of the NFSR state is often feasible. Through the state checking procedure, we can recover the correct value of the NFSR state at some time instant which is consistent with the given keystream, along with the round key bits. The above process will be specified as Algorithm 2 in Section 3.3. In the subsequent subsection, we give the complexities analysis of our attack methods against the generic model.

3.1 Independent Recovery of the LFSR Initial State with Multiple Linear Masks

In this subsection, we will show how to recover independently the initial state of the LFSR by the fast correlation attacks. First, we show how to derive the desirable parity-check equations for our generic model. Inspired by the work on the Grain family of stream ciphers in [27], the linear approximate representations are given for the generic model of small state stream ciphers. Compared to the analysis of Grain family, there are the round key bits involved in every linear approximate representation. To reduce the dimension of unknown variables from the secret key, we sample the parity-check equations at a time interval equal to the period of the round key bits. Then the original algorithm of [27] is modified to make two majority polls such that the new algorithm (Algorithm 1) can recover the correct value of the LFSR initial state no matter what the round key bits are. Another difference between small state stream ciphers and Grain family is that the size of the LFSR is always too small to obtain a valid attack by using directly Proposition 1 proposed in [27]. However, under the new observation, we can use fewer parity-check equations in Algorithm 1 when there are multiple different parity-check equations.

Constructing the Parity-check Equations There are three steps to construct the desirable parity-check equations, which are used in our modified fast correlation attack algorithms in subsequent content.

Step 1. Linear Approximate Representations The description of the generic model of Grain-like small state stream ciphers can be found in Section 2.1. Due to the involvement of the NFSR masking bits in the expression of z_t, it is infeasible to derive any useful approximate representation involving only the LFSR state bits when we consider one single keystream bit z_t. Therefore, we expect to derive the linear approximate representations for the sum of some keystream bits. Considering the sum of keystream bits over the set of taps $\mathbb{T}_z$, i.e., $\bigoplus_{i\in\mathbb{T}_z} z_{t+i}$, we can use the LFSR and NFSR state bits to represent it due to the output function Eq.(2). Namely,

$$\begin{aligned}\bigoplus_{i\in\mathbb{T}_z} z_{t+i} &= \bigoplus_{i\in\mathbb{T}_z}\left(h\left(L^{(t+i)}_{\mathbb{T}_{h,L}}, N^{(t+i)}_{\mathbb{T}_{h,N}}\right)\oplus\bigoplus_{b_1\in\mathbb{B}_1} l_{t+i+b_1}\oplus\bigoplus_{b_2\in\mathbb{B}_2} n_{t+i+b_2}\right)\\ &= \bigoplus_{i\in\mathbb{T}_z} h\left(L^{(t+i)}_{\mathbb{T}_{h,L}}, N^{(t+i)}_{\mathbb{T}_{h,N}}\right)\oplus\bigoplus_{i\in\mathbb{T}_z}\left(\bigoplus_{b_1\in\mathbb{B}_1} l_{t+i+b_1}\right)\oplus\bigoplus_{b_2\in\mathbb{B}_2}\left(\bigoplus_{i\in\mathbb{T}_z} n_{t+b_2+i}\right).\end{aligned}$$

To eliminate the NFSR masking bits from $\bigoplus_{i\in\mathbb{T}_z} z_{t+i}$, an appropriate set of taps $\mathbb{T}_z$ is chosen such that $\bigoplus_{i\in\mathbb{T}_z} n_{t+b_2+i}$ has a high bias. Considering the best linear approximation of the NFSR update function Eq.(1) with bias ϵ_{g^*} as follows,

$$n_{t+m'} \approx k'_t \oplus c_t \oplus l_t \oplus \bigoplus_{i\in\mathbb{I}_g} n_{t+i},$$

we choose the set of taps as $\mathbb{T}_z = \mathbb{I}_g \cup \{m'\}$, for simplicity we continue to use the notation $\mathbb{T}_z$ in the following. Then, the sum of the NFSR masking bits becomes

$$\begin{aligned}\bigoplus_{i\in\mathbb{T}_z} n_{t+b_2+i} &= \bigoplus_{i\in\mathbb{I}_g} n_{t+b_2+i}\oplus n_{t+b_2+m'}\\ &= k'_{t+b_2}\oplus c_{t+b_2}\oplus l_{t+b_2}\oplus g^*(N^{(t+b_2)}) \qquad \forall\, b_2,\end{aligned}$$

where $g^*(N^{(t)}) = \bigoplus_{i\in\mathbb{I}_g} n_{t+i}\oplus g(N^{(t)})$ and it has the same bias ϵ_{g^*}, i.e., $\Pr[g^*(N^{(t)}) = 0] = \frac{1}{2}+\epsilon_{g^*}$. Therefore, we have

$$\begin{aligned}\bigoplus_{i\in\mathbb{T}_z} z_{t+i} = &\bigoplus_{i\in\mathbb{T}_z}\left(\bigoplus_{b_1\in\mathbb{B}_1} l_{t+i+b_1}\right)\oplus\bigoplus_{b_2\in\mathbb{B}_2} l_{t+b_2}\oplus\bigoplus_{i\in\mathbb{T}_z} h\left(L^{(t+i)}_{\mathbb{T}_{h,L}}, N^{(t+i)}_{\mathbb{T}_{h,N}}\right)\oplus\bigoplus_{b_2\in\mathbb{B}_2} g^*(N^{(t+b_2)})\\ &\oplus\bigoplus_{b_2\in\mathbb{B}_2} k'_{t+b_2}\oplus\bigoplus_{b_2\in\mathbb{B}_2} c_{t+b_2}.\end{aligned}$$

Next we consider the linear approximate representation of $h(L^{(t+i)}_{\mathbb{T}_{h,L}}, N^{(t+i)}_{\mathbb{T}_{h,N}})$. Let $\boldsymbol{a}_i \in \{0,1\}^{s_1+s_2}$ be the input linear mask of h function at time instant $t+i$, i.e., $\boldsymbol{a}_i = (a_i[1],\cdots,a_i[s_1+s_2])$. Then

$$\begin{aligned}h\left(L^{(t+i)}_{\mathbb{T}_{h,L}}, N^{(t+i)}_{\mathbb{T}_{h,N}}\right) &\approx \boldsymbol{a}_i\cdot\left(L^{(t+i)}_{\mathbb{T}_{h,L}}, N^{(t+i)}_{\mathbb{T}_{h,N}}\right)^T\\ &= \boldsymbol{a}_i[1,\cdots,s_1]\cdot\left(L^{(t+i)}_{\mathbb{T}_{h,L}}\right)^T\oplus\boldsymbol{a}_i[s_1+1,\cdots,s_1+s_2]\cdot\left(N^{(t+i)}_{\mathbb{T}_{h,N}}\right)^T\end{aligned}$$

with bias $\epsilon_{h,i}(\boldsymbol{a}_i)$, where $\boldsymbol{a}_i[x,\cdots,y]$ denotes a subvector indexed from x-th bit to y-th bit, the operator $(\cdot)^T$ is the transpose of a row vector and the dot operator $\cdot$ between a row vector and a column vector represents the usual inner GF(2)-product. There are $|\mathbb{T}_z|$ active h functions which need to be approximated. Let $\boldsymbol{a}_{\mathbb{T}_z} \in \{0,1\}^{(s_1+s_2)\times|\mathbb{T}_z|}$ be the concatenated linear mask of all the $\boldsymbol{a}_i$ satisfying $i\in\mathbb{T}_z$. The total bias of all the approximated h functions depends on $\boldsymbol{a}_{\mathbb{T}_z}$, and it is computed as $\epsilon_{h,\mathbb{T}_z}(\boldsymbol{a}_{\mathbb{T}_z}) = 2^{|\mathbb{T}_z|-1}\times\prod_{i\in\mathbb{T}_z}\epsilon_{h,i}(\boldsymbol{a}_i)$ because of the piling-up lemma.

Under the bias $\epsilon_{h,\mathbb{T}_z}(\boldsymbol{a}_{\mathbb{T}_z})$, we get

$$\begin{aligned}\bigoplus_{i\in\mathbb{T}_z} z_{t+i} \approx &\bigoplus_{i\in\mathbb{T}_z}\left(\bigoplus_{b_1\in\mathbb{B}_1} l_{t+i+b_1}\right)\oplus\bigoplus_{b_2\in\mathbb{B}_2} l_{t+b_2}\oplus\bigoplus_{i\in\mathbb{T}_z}\boldsymbol{a}_i[1,\cdots,s_1]\cdot\left(L^{(t+i)}_{\mathbb{T}_{h,L}}\right)^T\oplus\bigoplus_{b_2\in\mathbb{B}_2} k'_{t+b_2}\oplus\bigoplus_{b_2\in\mathbb{B}_2} c_{t+b_2}\\ &\oplus\left(\bigoplus_{i\in\mathbb{T}_z}\boldsymbol{a}_i[s_1+1,\cdots,s_1+s_2]\cdot\left(N^{t+i}_{\mathbb{T}_{h,N}}\right)^T\oplus\bigoplus_{b_2\in\mathbb{B}_2} g^*(N^{(t+b_2)})\right).\end{aligned}$$

All the terms involved in the internal states of the LFSR and the sum of the round key bits $\bigoplus_{b_2\in\mathbb{B}_2} k'_{t+b_2}$ will be guessed in our fast correlation attacks. Note that c_t is a known constant bit. Therefore, if the bias of last term in the above approximate representation is high, we could carry out our fast correlation attacks. Let

$$\epsilon_{g^*,\mathbb{B}_2}(\boldsymbol{a}_{\mathbb{T}_z}) = \Pr\left[\bigoplus_{i\in\mathbb{T}_z}\boldsymbol{a}_i[s_1+1,\cdots,s_1+s_2]\cdot\left(N^{t+i}_{\mathbb{T}_{h,N}}\right)^T\oplus\bigoplus_{b_2\in\mathbb{B}_2} g^*(N^{(t+b_2)}) = 0\right]-\frac{1}{2}$$

and the bias is independent on $\boldsymbol{a}_i[1,\cdots,s_1]$ for all $i \in \mathbb{T}_z$.

For any fixed $\boldsymbol{a}_{\mathbb{T}_z}$, we can derive the following linear approximate representation

$$\begin{aligned}\bigoplus_{i\in\mathbb{T}_z} z_{t+i} \approx \bigoplus_{i\in\mathbb{T}_z}\left(\bigoplus_{b_1\in\mathbb{B}_1} l_{t+i+b_1}\right) \oplus \bigoplus_{b_2\in\mathbb{B}_2} l_{t+b_2} \oplus \bigoplus_{i\in\mathbb{T}_z} \boldsymbol{a}_i[1,\cdots,s_1]\cdot\left(L^{(t+i)}_{\mathbb{T}_{h,L}}\right)^T \\ \oplus \bigoplus_{b_2\in\mathbb{B}_2} k'_{t+b_2} \oplus \bigoplus_{b_2\in\mathbb{B}_2} c_{t+b_2}\end{aligned} \tag{4}$$

and the bias is evaluated as $2\times\epsilon_{h,\mathbb{T}_z}(\boldsymbol{a}_{\mathbb{T}_z})\times\epsilon_{g^*,\mathbb{B}_2}(\boldsymbol{a}_{\mathbb{T}_z})$.

Step 2. Linear Approximate Equations With Eq.(3) $L^{(t)} = L^{(0)}\times F^t$, for any fixed $\boldsymbol{a}_{\mathbb{T}_z}$, we rewrite Eq.(4) as

$$\bigoplus_{i\in\mathbb{T}_z} z_{t+i} \approx L^{(0)}\cdot\left(F^t\times U(\boldsymbol{a}_{\mathbb{T}_z})\right) \oplus \bigoplus_{b_2\in\mathbb{B}_2} k'_{t+b_2} \oplus \bigoplus_{b_2\in\mathbb{B}_2} c_{t+b_2},$$

where

$$U(\boldsymbol{a}_{\mathbb{T}_z}) = \bigoplus_{i\in\mathbb{T}_z}\left(\bigoplus_{b_1\in\mathbb{B}_1}\boldsymbol{g}_{i+b_1} \oplus \bigoplus_{j\in\{1,\cdots,s_1\}} a_i[j]\cdot\boldsymbol{g}_{i+\mathbb{T}_{h,L}[j]}\right) \oplus \bigoplus_{b_2\in\mathbb{B}_2}\boldsymbol{g}_{b_2},$$

F is the state transition matrix of the LFSR, $\boldsymbol{g}_q$ is the first column of the matrix F^q and $\mathbb{T}_{h,L}[j]$ is the j-th element of $\mathbb{T}_{h,L} = (\gamma_1,\cdots,\gamma_{s_1})$. From the above linear approximate representations, we can derive the linear approximate equation with a fixed linear mask $\boldsymbol{u}$

$$\bigoplus_{i\in\mathbb{T}_z} z_{t+i} \approx L^{(0)}\cdot\left(F^t\times\boldsymbol{u}\right) \oplus \bigoplus_{b_2\in\mathbb{B}_2} k'_{t+b_2} \oplus \bigoplus_{b_2\in\mathbb{B}_2} c_{t+b_2}, \tag{5}$$

where $\boldsymbol{u}\in\{0,1\}^m$ is a column vector. If different $\boldsymbol{a}_{\mathbb{T}_z}$'s derive the same linear mask $\boldsymbol{u}$, the corresponding biases should be added up to get the bias of $\boldsymbol{u}$, i.e., $\epsilon_{\boldsymbol{u}} = \sum_{\{\boldsymbol{a}_{\mathbb{T}_z}|U(\boldsymbol{a}_{\mathbb{T}_z})=\boldsymbol{u}\}} 2\times\epsilon_{h,\mathbb{T}_z}(\boldsymbol{a}_{\mathbb{T}_z})\times\epsilon_{g^*,\mathbb{B}}(\boldsymbol{a}_{\mathbb{T}_z})$. As a rough estimation for the generic cipher model, we can find $r = 2^{s_1\times|\mathbb{T}_z|}$ different linear masks $\boldsymbol{u}$ with the bias

$$\begin{aligned}\epsilon &= 2^{s_2\times|\mathbb{T}_z|+1}\times\left(2^{|\mathbb{T}_z|-1}\epsilon_h^{|\mathbb{T}_z|}\right)\times\left(2^{q_2-1}\epsilon_{g^*}^{q_2}\right)\\ &= 2^{(s_2+1)\times|\mathbb{T}_z|+q_2-1}\times\epsilon_h^{|\mathbb{T}_z|}\times\epsilon_{g^*}^{q_2}\end{aligned}$$

where ϵ_h and ϵ_{g^*} are the biases of the linear approximations for the functions h and g respectively, s_1 and s_2 are the number of input variables of h from the LFSR and NFSR respectively, q_2 is the number of the NFSR masking variables of the output function, i.e., $q_2 = |\mathbb{B}_2|$.

Step 3. Building the Parity-check Equations Let $\hat{z}_t = \bigoplus_{i\in\mathbb{T}_z} z_{t+i}$, $\hat{k}_t = \bigoplus_{b_2\in\mathbb{B}_2} k'_{t+b_2}$, $\hat{c}_t = \bigoplus_{b_2\in\mathbb{B}_2} c_{t+b_2}$ and $e_{t,j}$ be the random noise introduced by the corresponding linear approximation with linear mask $\boldsymbol{u}_j$ for the sum of keystream bits $\hat{z}_t$. From Eq.(5), we actually obtain a noisy system with r different linear approximate equations on the unknown variables $L^{(0)} = (l_0,\cdots,l_{m-1})$ and $\hat{k}_t$, which is rewritten as

$$L^{(0)}\cdot\left(F^t\times\boldsymbol{u}_j\right)\oplus\hat{z}_t\oplus\hat{c}_t\oplus\hat{k}_t = e_{t,j}\quad t\geq 0,\ j=1,\cdots,r$$

where $e_{t,j}$ are the random variables satisfying $\Pr[e_{t,j}=0] = \frac{1}{2}+\epsilon_j$ and $\epsilon_j\approx\epsilon$.

From the Assumed Property 1 that the $RKF(\cdot)$ is periodic, we have that the unknown round key bits k'_t has a cycle of length d, i.e., $k'_{t_0+dt'} = k'_{t_0}$ for $t_0 = 0,\cdots,d-1$ and any $t'\geq 0$. Accordingly, we get that

$$L^{(0)}\cdot\left(F^{t_0+dt'}\times\boldsymbol{u}_j\right)\oplus\hat{z}_{t_0+dt'}\oplus\hat{c}_{t_0+dt'}\oplus\hat{k}_{t_0} = e_{t_0+dt',j}\quad t'\geq 0,\ j=1,\cdots,r.$$

To reduce the dimension of unknown variables from the secret key, we sample parity-check equations at a time interval equal to the period of the round key bits. Namely, by choosing $t_0 = 0$, we receive a noisy system on $m+1$ unknown variables $L^{(0)} = (l_0,\cdots,l_{m-1})$ and $\hat{k}_0$

$$L^{(0)}\cdot\left(F^{dt'}\times\boldsymbol{u}_j\right)\oplus\hat{z}_{dt'}\oplus\hat{c}_{dt'}\oplus\hat{k}_0 = e_{dt',j}\quad t'=0,\cdots,\mathit{\Omega}-1,\ j=1,\cdots,r. \tag{6}$$

where Ω is a parameter to be determined later.

According to Theorem 1, we get $L^{(0)} \cdot (F^{dt'} \times \boldsymbol{u}_j) = (L^{(0)} \times F_{\boldsymbol{u}_j}) \cdot \boldsymbol{g}_{dt'}$, thus Eq.(6) can be rewritten as

$$\left(L^{(0)} \times F_{\boldsymbol{u}_j}\right) \cdot \boldsymbol{g}_{dt'} \oplus \hat{z}_{dt'} \oplus \hat{c}_{dt'} \oplus \hat{k}_0 = e_{dt',j} \quad t' = 0, \cdots, \Omega - 1,\ j = 1, \cdots, r. \tag{7}$$

where $F_{\boldsymbol{u}_j}$ defined in Theorem 1 can be computed from the linear mask $\boldsymbol{u}_j$ and $\boldsymbol{g}_t$ is the first column vector of the matrix F^t. Here the random noises are satisfying $\Pr[e_{t,j} = 0] = \frac{1}{2} + \epsilon_j = \frac{1}{2}\left(1 + \epsilon_j^c\right)$ and $\epsilon_j^c \approx 2\epsilon = 2^{(s_2+1)\times|\mathbb{T}_z|+q_2} \times \epsilon_h^{|\mathbb{T}_z|} \times \epsilon_{g^*}^{q_2}$ for $j = 1, \cdots, r = 2^{s_1 \times |\mathbb{T}_z|}$.

Fast Correlation Attacks with Multiple Linear Masks At the former part, we have derived r linear masks with bias about ϵ for the generic cipher model. In the cryptanalysis of the concrete Grain-like small state stream ciphers, ϵ is usually chosen as a threshold for the bias of linear masks. Therefore, we regard $\epsilon\,(\epsilon > 0)$ as the lower bound for absolute value of the bias of linear masks in the following discuss. Besides, we assume that there are r_0 linear masks $\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_{r_0}$ with positive bias and r_1 linear masks $\boldsymbol{u}_{r_0+1}, \boldsymbol{u}_{r_0+2}, \cdots, \boldsymbol{u}_{r_0+r_1}$ with negative bias. Note that the threshold ϵ might be closed to even smaller than $2^{-\frac{\kappa}{2}}$, and $r = r_0 + r_1$.

To use the new wrong-initial-state hypothesis introduced in [27], we construct parity-check equations from Eq.(7), i.e., using the linear mask $\boldsymbol{g}_{dt'}$ instead of $F^{dt'} \times \boldsymbol{u}_j$. Namely,

$$(l'_0, \cdots, l'_{m-1}) \cdot \boldsymbol{g}_{dt'} \oplus \hat{z}_{dt'} \oplus \hat{c}_{dt'} \oplus \hat{k}_0, \quad t = 1, \cdots, \Omega$$

where $L'^{(0)} = (l'_0, \cdots, l'_{m-1})$ is the guessed value of the LFSR state $L^{(0)} \times F_{\boldsymbol{u}_j}$, $\boldsymbol{g}_{dt'}$ is the first column of matrix $F^{dt'}$, $\hat{z}_{dt'}, \hat{c}_{dt'}$ and $\hat{k}_0$ are the sum of the keystream bits, counter bits and round key bits, respectively. Here we introduce the indicator for every parity-check equation as

$$\Delta_{t'}(l'_0, \cdots, l'_{m-1}) = (l'_0, \cdots, l'_{m-1}) \cdot \boldsymbol{g}_{dt'} \oplus \hat{z}_{dt'} \oplus \hat{c}_{dt'} \oplus \hat{k}_0.$$

If the value of $L'^{(0)} = (l'_0, \cdots, l'_{m-1})$ is guessed as $L^{(0)} \times F_{\boldsymbol{u}_j}$ and the sum of the round key bits $\hat{k}_0$ is correctly guessed, then $\Delta_{t'}(L'^{(0)}) = e_{dt',j}$, $j = 1, \cdots, r$ and $\Pr[\Delta_{t'}(L'^{(0)}) = 0] = \frac{1}{2}(1 + \epsilon_j^c)$, where $|\epsilon_j^c| \geq \epsilon^c = 2\epsilon$. Besides, if the value of $L'^{(0)}$ is guessed as $L^{(0)} \times F_{\boldsymbol{u}_j}$ and the sum of the round key bits $\hat{k}_0$ is wrongly guessed, then $\Delta_{t'}(L'^{(0)})$ will flip the value of $e_{dt',j}$, i.e., $\Delta_{t'}(L'^{(0)}) = e_{dt',j} \oplus 1$, $j = 1, \cdots, r$, thus $\Pr[\Delta_{t'}(L'^{(0)}) = 0] = 1 - \frac{1}{2}(1 + \epsilon_j^c) = \frac{1}{2}(1 - \epsilon_j^c)$, where $|-\epsilon_j^c| \geq \epsilon^c$. Therefore, no matter what the sum of the round key bits $\hat{k}_0$ is guessed, we can get a highly biased indicator for every parity-check equation which has the bias of possibly different sign. Hereinafter, we just ignore the sum of the round key bits $\hat{k}_0$ in the indicator, i.e., $\Delta_{t'}(l'_0, \cdots, l'_{m-1}) = (l'_0, \cdots, l'_{m-1}) \cdot \boldsymbol{g}_{dt'} \oplus \hat{z}_{dt'} \oplus \hat{c}_{dt'}$. Finally, if the guessed initial state $L'^{(0)}$ is not in the set $\{L^{(0)} \times F_{\boldsymbol{u}_j}, j = 1, \cdots, r\}$, $\Delta_{t'}(L'^{(0)})$ will always be assumed to behave at random and $\Pr[\Delta_{t'}(L'^{(0)}) = 0] = \frac{1}{2}$.

For simplicity of the analysis, we just use ϵ^c instead of the true value of bias of parity-check equations, which is the lower bound for absolute value of all the ϵ_j^c, i.e., $|\epsilon_j^c| \geq \epsilon^c$. Let $\mathbb{S}_h$ be the set of values of the LFSR initial state that have a highly biased indicator when we construct parity-check equations using $\boldsymbol{g}_{dt'}$ together with $\hat{z}_{dt'} \oplus \hat{c}_{dt'}$ and $\mathbb{S}_l$ the set of remaining values, i.e., $\mathbb{S}_h = \{L^{(0)} \times F_{\boldsymbol{u}_j}, j = 1, \cdots, r\}$ and $\mathbb{S}_l = \{0,1\}^m \setminus \mathbb{S}_h$. For the indicator of the parity-check equations, we define the statistic $\mathcal{E}$ of its bias as

$$\mathcal{E}(l'_0, \cdots, l'_{m-1}) = \sum_{t'=0}^{\Omega-1} (-1)^{\Delta_{t'}(l'_0, \cdots, l'_{m-1})}$$

According to the central limit theorem, we have

$$\mathcal{E}(L'^{(0)}) \sim \mathcal{N}(\Omega\epsilon^c, \Omega(1 - (\epsilon^c)^2)) \approx \mathcal{N}(\Omega\epsilon^c, \Omega)$$

or

$$\mathcal{E}(L'^{(0)}) \sim \mathcal{N}(-\Omega\epsilon^c, \Omega)$$

when $L'^{(0)} \in \mathbb{S}_h$, where $(\epsilon^c)^2$ is enough small to make the above approximation, and

$$\mathcal{E}(L'^{(0)}) \sim \mathcal{N}(0, \Omega)$$

when $L'^{(0)} \in \mathbb{S}_l$, where $\mathcal{N}(\cdot,\cdot)$ is the normal distribution with the specified expectation and variance. A naive method of evaluating Ω parity-check equations for all the possible values of the LFSR initial state has a time complexity of $2^m\Omega$, which is quite inefficient. As Chose et al. have showed in [11], the fast Walsh-Hadamard transform (FWHT) can be successfully applied to accelerate the guess and evaluation procedure. According to the mask pattern of the parity-check equations, we regroup the parity-check equations and define an integer-valued function $w(\cdot) : \{0,1\}^m \to \mathbb{Z}$ as

$$w(\boldsymbol{a}) = \sum_{t' \in \{0,\cdots,\Omega-1 | \boldsymbol{g}_{dt'} = \boldsymbol{a}^T\}} (-1)^{\hat{z}_{dt'} \oplus \hat{c}_{dt'}}$$

where $\boldsymbol{a} \in \{0,1\}^m$ is a row vector and the sum is computed over the set of integers. Therefore, the statistic at one value $L'^{(0)} = (l'_0, \cdots, l'_{m-1})$ can be computed as follows

$$\begin{aligned}
&\mathcal{E}(l'_0, \cdots, l'_{m-1}) \\
=& \sum_{t'=0}^{\Omega-1} (-1)^{\Delta_{t'}(l'_0,\cdots,l'_{m-1})} \\
=& \sum_{t'=0}^{\Omega-1} (-1)^{(l'_0,\cdots,l'_{m-1})\cdot \boldsymbol{g}_{dt'} \oplus \hat{z}_{dt'} \oplus \hat{c}_{dt'}} \\
=& \sum_{\boldsymbol{a}\in\{0,1\}^m} \left(\sum_{t' \in \{0,\cdots,\Omega-1 | \boldsymbol{g}_{dt'} = \boldsymbol{a}^T\}} (-1)^{\hat{z}_{dt'} \oplus \hat{c}_{dt'}} \right) \cdot \\
&(-1)^{\boldsymbol{a}\cdot(l'_0,\cdots,l'_{m-1})^T} \\
=& \sum_{\boldsymbol{a}\in\{0,1\}^m} w(\boldsymbol{a}) \cdot (-1)^{\boldsymbol{a}\cdot(l'_0,\cdots,l'_{m-1})^T} \\
=& W(l'_0, \cdots, l'_{m-1}),
\end{aligned}$$

where $W(l'_0, \cdots, l'_{m-1})$ is the Walsh transform of $w(\boldsymbol{a})$ at the point $(l'_0, \cdots, l'_{m-1})$. From the above, we can use FWHT to evaluate Ω parity-check equations for all the possible value of the LFSR m-bit state with time complexity $\Omega + m2^m$ and memory complexity 2^m.

Fortunately, for small state stream ciphers, the size m of the LFSR is always much smaller than the security margin κ, thus guessing the whole of LFSR state is often feasible. To make our attacks more flexible, we will ignore β bits of the LFSR state and guess its partial $n-\beta$ bits by exploiting the technique presented in [27]. With appropriately choosing a value for the parameter β, a more efficient attack can be derived. The bypassed β bits can be fixed to any constant and we set them to all zeros in the following discussion.

The original algorithm proposed in [27] is modified to make two majority polls at the last processing step so that the LFSR initial state can be recovered no matter what the round key bits are. Now we present the modified algorithm for recovering the initial state of the LFSR as Algorithm 1, where two cases of the majority polls at part 3 are corresponding to two possible values of $\hat{k}_0$. Note that the notations $th\,(th > 0)$ and $th_p\,(th_p > 0)$ are the thresholds for the filtering restriction of statistical test and the judging condition of being choose as candidates of the LFSR initial state, which will be determined in the following sections.

3.2 New Relationship Between the Number and Bias of Required Parity-Check Equations

In the traditional fast correlation attacks of [31, 6, 10], to identify the unique correct value of the LFSR initial state with a high probability, the number of required parity-check equations Ω should be chosen as $\Omega \geq \frac{(2-(\epsilon^c)^2)2m\ln 2}{(\epsilon^c)^2} \approx \frac{4m\ln 2}{(\epsilon^c)^2}$, where m is the size of the LFSR and ϵ^c is twice as many as bias of the parity-check equations. Since the size of the LFSR is always much small in Grain-like small state stream ciphers, no valid attack against them can be obtained by using directly Proposition 1 proposed in [27]. We first observe that the number of required parity-check equations can be reduced when there are multiple different parity-check equations. With exploiting the Skellam distribution, we introduce a sufficient condition to identify the unique correct LFSR initial state and derive a new relationship

Algorithm 1 Recovery of the LFSR Initial State

```
Input: Given keystream bits, $\{z_t\}$; The state transition matrix of the LFSR from the target stream cipher, $F$;
    The inverse of matrices corresponding to all the highly biased linear masks, $\{F_{\boldsymbol{u}_j}^{-1}\}_{j=1}^{r}$.
    Parameters: The size of bypassed bits, $\beta, 0 \leq \beta < m$; The number of parity-check equations, $\Omega$; The
    threshold for the statistical test, $th$; The threshold for the majority polls, $th_p$.
Output: A set of candidates of the LFSR initial state.
        /* Part 1: Prepare $w(\boldsymbol{a})$ for evaluation of the parity-check equations */
    Calculate and store the integer-valued function $w(\boldsymbol{a}) = \sum_{t' \in \{0,\cdots,\Omega-1 | \boldsymbol{g}_{dt'}[1,\cdots,m-\beta]=\boldsymbol{a}^T\}} (-1)^{\hat{z}_{dt'} \oplus \hat{c}_{dt'}}$, $\forall \boldsymbol{a} \in$
    $\{0,1\}^{m-\beta}$, where $\boldsymbol{g}_{dt'}$ is the first column of matrix $F^{dt'}$, $\hat{z}_{dt'} = \bigoplus_{i \in \mathbb{T}_z} z_{dt'+i}$ is the sum of bits from the given
    keystream bits and $\hat{c}_{dt'} = \bigoplus_{b_2 \in \mathbb{B}_2} c_{dt'+b_2}$ is the constant sum from the counter $C_c$ of the target stream cipher.
        /* Part 2: Filter the LFSR initial states with the fast Walsh-Hadamard transform */
    Compute the Walsh spectrum of $w(\boldsymbol{a})$ though the subroutine of the fast Walsh-Hadamard transform, i.e.,
    $[W(L'^{(0)})] = FWHT([w(\boldsymbol{a})])$, where $L'^{(0)} \in \{0,1\}^{m-\beta}$;
    if $W(L'^{(0)}) \geq th$ then
        Store $L'^{(0)}$ in the set $\mathbb{V}_0$.
    end if
    if $W(L'^{(0)}) \leq -th$ then
        Store $L'^{(0)}$ in the set $\mathbb{V}_1$
    end if
        /* Part 3: Identify candidates of the LFSR initial state with two majority polls */
    for all $\alpha \in \{0,1\}$ do
        Set the voting counter to zero, i.e., $poll_\alpha[\cdot] = 0$, and make the $\alpha$-th majority poll:
        for all $i \in \{0,1\}$ do
            for all $L'^{(0)} \in \mathbb{V}_i$ do
                Compute $\bar{L}^{(0)} = (L'^{(0)} || 0^\beta) \times F_{\boldsymbol{u}_j}^{-1}$, $\forall j \in J_{i \oplus \alpha}$, then $poll_\alpha[\bar{L}^{(0)}]$ increment by 1, where $J_0 =$
                    $\{1, \cdots, r_0\}$, $J_1 = \{r_0+1, \cdots, r\}$.
                if $poll_\alpha[\bar{L}^{(0)}] \geq th_p$ then
                    Keep $\bar{L}^{(0)}$ in the set of candidates of the LFSR initial state $\mathbb{C}_\alpha$.
                end if
            end for
        end for
    end for
    return $\mathbb{C}_0 \cup \mathbb{C}_1$.
```

between the number and bias of required parity-check equations as $\Omega = \frac{2^{\frac{7}{2}}\sqrt{m \ln 2}}{\sqrt{r}(\epsilon^c)^2}$, where r is the number of different parity-check equations. Therefore, when there are r different parity-check equations, we can reduce the number of required parity-check equations to about $\frac{1}{\sqrt{r}}$ times of the traditional methods.

Let p_1 be the probability that the random variable following $\mathcal{N}(0, \Omega)$ is greater than th, and let p_2 be the probability that the random variable following $\mathcal{N}(\Omega\epsilon^c, \Omega)$ is greater than th. Let $Q(x)$ be the tail distribution function of the standard normal distribution, i.e., $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{y^2}{2}}\, dy$, thus the probability that every value in $\mathbb{S}_l$ has an empirical statistic $\mathcal{E}(L'^{(0)}))$ greater than th is approximately $p_1 = Q(\frac{th}{\sqrt{\Omega}})$ and the probability that every value with positive bias in $\mathbb{S}_h$ has an empirical statistic $\mathcal{E}(L'^{(0)}))$ greater than th is approximately $p_2 = Q(\frac{th - \Omega\epsilon^c}{\sqrt{\Omega}})$. Note that the probability that the random variable following $\mathcal{N}(0, \Omega)$ is smaller than $-th$ is also p_1 and the probability that the random variable following $\mathcal{N}(-\Omega\epsilon^c, \Omega)$ is smaller than $-th$ is also p_2. Assuming that the values in $\mathbb{S}_h$ is random uniformly distributed, then the expected number of values in $\mathbb{S}_h$ which pass the statistical test is $r_0 2^{-\beta} p_2 + r_1 2^{-\beta} p_2$ even when β $(0 \leq \beta \leq \log(r))$ bits are bypassed. Moreover, there are about $2^{m-\beta}p_1 + 2^{m-\beta}p_1$ values in $\mathbb{S}_l$ which pass the statistical test. Therefore, the expected size of $\mathbb{V}_0$ and $\mathbb{V}_1$ are $2^{m-\beta}p_1 + r_0 2^{-\beta} p_2$ and $2^{m-\beta}p_1 + r_1 2^{-\beta} p_2$, respectively.

In Proposition 1 of [27], Todo et al. proposed to set $\Omega = (m-\beta)2^{m-\beta} = r2^{n-\beta}p_1$ to make a balance between the complexities of three parts in Algorithm 1, and the probability of recovering successfully the LFSR initial state was computed for different choices of β. Unfortunately, with setting $\Omega = (m-\beta)2^{m-\beta}$, the probability of recovering successfully the LFSR initial state is always 0 because the size m of the LFSR is often too small in Grain-like small state stream ciphers. Therefore, when there are multiple different parity-check equations, we need a new relationship between the number and bias of required parity-check equations for applying successfully Algorithm 1 to recover the correct value of the LFSR initial state.

Theorem 2. *Let m be the size of the LFSR from the target stream cipher and β $(0 \leq \beta \leq \log(r))$ be the size of bypassed bits. Assuming that there are r different parity-check equations whose absolute value of bias is greater than $\frac{\epsilon^c}{2}$, then the number Ω of required parity-check equations for Algorithm 1 to succeed is*

$$\Omega \approx \frac{2^{\frac{\beta+7}{2}}\sqrt{m \ln 2}}{\sqrt{r}(\epsilon^c)^2},$$

more precisely,

$$\Omega = \frac{4\left(Q^{-1}\left(\frac{1}{2}\left(1-\sqrt{A(2-A)}\right)\right)\right)^2}{(\epsilon^c)^2},$$

where $A = \frac{2^\beta m \ln 2}{r} < 1$ and Q^{-1} is the inverse function of Q-function.

Proof. Here we only give details of the proof when the sum of the round key bits $\hat{k}_0$ is zero, and the another situation can be argued in a similar way.

When $\alpha = 0$, i.e., for the 0-th majority poll, every wrong value appears about

$$\mu_1 = (r2^{m-\beta}p_1 + (r_0^2 + r_1^2)2^{-\beta}p_2)2^{-m} \approx r2^{-\beta}p_1$$

times, where $r2^{-\beta}p_1 >> (r_0^2 + r_1^2)2^{-(m+\beta)}p_2$ holds in the useful attack parameters, and the correct value $L^{(0)}$ appears

$$\mu_2 = (r_0 + r_1)2^{-\beta}p_2 = r2^{-\beta}p_2$$

times. Therefore, the number of occurrences that every wrong state value appears, denoted by X_1, follows the Poisson distribution with expected value μ_1, and the number of occurrences that the correct state value appears, denoted by X_2, follows the Poisson distribution with expected value μ_2, i.e., $X_1 \sim \mathit{Pois}(\mu_1)$ and $X_2 \sim \mathit{Pois}(\mu_2)$. Let $Y = X_1 - X_2$, the difference of X_1 and X_2, thus the random variable Y follows a Skellam distribution assuming that X_1 and X_2 are two statistically independent random variables, i.e., $Y \sim \mathit{Skellam}(\mu_1, \mu_2)$ with $\mu_1 < \mu_2$. With the property of Skellam distribution, we get a upper bound for the probability that a wrong value has a better rank than the correct value, i.e., $\Pr[X_1 \geq X_2] = \Pr[Y \geq 0] \leq \exp(-(\sqrt{\mu_1} - \sqrt{\mu_2})^2) = \exp(-r2^{-\beta}(\sqrt{p_1} - \sqrt{p_2})^2)$. Thus to identify the unique correct value $L^{(0)}$, we introduce the sufficient condition as

$$\exp(-r2^{-\beta}(\sqrt{p_1} - \sqrt{p_2})^2) < 2^{-m}$$

i.e.,

$$\sqrt{p_2} - \sqrt{p_1} > \frac{2^{\frac{\beta}{2}}\sqrt{m \ln 2}}{\sqrt{r}} \tag{8}$$

From this sufficient condition, we can heuristically obtain that a good choice for the threshold of the statistical test should make the difference $\sqrt{p_2} - \sqrt{p_1}$ as great as possible. With $th = \frac{1}{2}\Omega\epsilon^c$, the difference $p_2 - p_1 = Q(-\frac{1}{2}\sqrt{\Omega}\epsilon^c) - Q(\frac{1}{2}\sqrt{\Omega}\epsilon^c) = 1 - 2Q(\frac{1}{2}\sqrt{\Omega}\epsilon^c)$ is largest and $\sqrt{p_2} - \sqrt{p_1} = \frac{p_2 - p_1}{\sqrt{p_2} + \sqrt{p_1}} = \frac{1-2Q(\frac{1}{2}\sqrt{\Omega}\epsilon^c)}{\sqrt{p_2}+\sqrt{p_1}}$, where $p_1 + p_2 = 1$.

The difference $\sqrt{p_2} - \sqrt{p_1}$ can be evaluated using tables or the tight upper and lower bounds for the Q-function.

Moreover, $\sqrt{p_2} - \sqrt{p_1}$ can be lower bounded by using

$$Q(x) < \frac{1}{2}e^{-\frac{x^2}{2}}, \quad x \geq 0$$

and we can derive

$$\begin{aligned}\sqrt{p_2} - \sqrt{p_1} &> \frac{1 - e^{-\frac{\Omega(\epsilon^c)^2}{8}}}{\sqrt{p_2} + \sqrt{p_1}} \\ &\geq \frac{1 - e^{-\frac{\Omega(\epsilon^c)^2}{8}}}{2\sqrt{\frac{p_1+p_2}{2}}} \\ &= \frac{1}{\sqrt{2}}\left(1 - e^{-\frac{\Omega(\epsilon^c)^2}{8}}\right)\end{aligned}$$

Next, we convert the sufficient condition to a stronger version

$$\frac{1}{\sqrt{2}}\left(1-e^{-\frac{\Omega(\epsilon^c)^2}{8}}\right) \geq \frac{2^{\frac{\beta}{2}}\sqrt{m\ln 2}}{\sqrt{r}}.$$

With the Taylor Series of e^x, i.e., $e^x=\sum_{n=0}^{\infty}\frac{x^n}{n!}$, the approximation $e^x \approx 1+x$ can be obtained. Therefore, the stronger sufficient condition can be written as

$$\frac{1}{\sqrt{2}}\left(1-(1-\frac{\Omega(\epsilon^c)^2}{8})\right) \geq \frac{2^{\frac{\beta}{2}}\sqrt{m\ln 2}}{\sqrt{r}},$$

which indicates

$$\Omega \geq \frac{2^{\frac{\beta+7}{2}}\sqrt{m\ln 2}}{\sqrt{r}(\epsilon^c)^2}. \tag{9}$$

More precisely, from the sufficient condition Eq.(8) we can derive that

$$p_1 = Q\left(\frac{1}{2}\sqrt{\Omega}\epsilon^c\right) \leq \frac{1}{2}\left(1-\sqrt{A(2-A)}\right),$$

where $A=\frac{2^{\beta}m\ln 2}{r}<1$. Accordingly, we get the precise result as

$$\Omega \geq \frac{4\left(Q^{-1}\left(\frac{1}{2}\left(1-\sqrt{A(2-A)}\right)\right)\right)^2}{(\epsilon^c)^2}, \tag{10}$$

where Q^{-1} is the inverse function of Q-function.

We choose the value whose poll is maximum as a candidate of the LFSR initial state and keep it in the set $\mathbb{C}_0$. Therefore, if $th=\frac{1}{2}\Omega\epsilon^c$ and Ω is set as Eq.(9) or Eq.(10) in the precise mode, the probability that a wrong value is chosen as a candidate of the LFSR initial state is less than 2^{-m}.

Regarding the 1-th majority poll, the correct value does not appear and every wrong value appears about μ_1 times, which is the same as the 0-th majority poll. Similarly, the value whose poll is maximum will be chosen as a candidate of the LFSR initial state and kept in the set $\mathbb{C}_1$.

In conclusion, if $th=\frac{1}{2}\Omega\epsilon^c$ and Ω is set as Eq.(9) or Eq.(10) in the precise mode, the output $\mathbb{C}_0 \cup \mathbb{C}_1$ of Algorithm 1 contains the correct value of the LFSR initial state with a high probability. □

Practically, we can set a lower threshold th_p for the number of occurrences that one value appears such that many values could be chosen as candidates of the LFSR initial state and more flexible attacks would be implemented. Let p_c the probability that the correct value will be chosen as a candidate and p_w the probability that a wrong value will be chosen as a candidate, then

$$p_c = \sum_{x_1=th_p}^{\infty}\frac{\mu_1^{x_1}e^{-\mu_1}}{x_1!},$$

$$p_w = \sum_{x_2=th_p}^{\infty}\frac{\mu_2^{x_2}e^{-\mu_2}}{x_2!}.$$

In cryptanalysis, the threshold th_p is expected to chosen appropriately such that the correct value will pass the majority polls with a high probability, i.e., p_c is very closed to 1, meanwhile all the wrong values will be filtered out as much as possible, i.e., p_w is very small.

3.3 Recovery of the NFSR State at some time instant and the Round Key Bits

Now we first consider the degraded system assuming that the LFSR initial state is known and derive relations between the internal state variables of the NFSR. Then an algorithm of recovering the NFSR state at some time instant and the round key bits will be proposed in the following.

The Degraded System Suppose that the attacker somehow knows the LFSR initial state $L^{(0)} = (l_0, \cdots, l_{m-1})$ and has access to some keystream bits. Since the LFSR updates independently, the attacker can clock the LFSR forwards and backwards to remove its protection over the keystream bits. The resultant system becomes an NFSR which is nonlinearly updated, involving the periodic round key bits and the output of this system is filtered by a nonlinear function h. Given the NFSR state $N^{(t)} = (n_t, \cdots, n_{t+m'-1})$ at time instant t, we rewrite the keystream bit for the generic model as

$$z_t = \bigoplus_{b_2 \in \mathbb{B}_2} \underline{n_{t+b_2}} \oplus h\left(L^{(t)}_{\mathbb{T}_{h,L}}, \underline{N^{(t)}_{\mathbb{T}_{h,N}}}\right) \oplus \bigoplus_{b_1 \in \mathbb{B}_1} l_{t+b_1} \tag{11}$$

where any internal LFSR variable is known, $\mathbb{T}_{h,N} = (\delta_1, \cdots, \delta_{s_2})$ and $\mathbb{B}_2 = \{\eta_1, \cdots, \eta_{q_2}\}$ are the sets of the NFSR taps with $0 \leq \delta_1 < \cdots < \delta_{s_2} \leq m' - 1$ and $0 \leq \eta_1 < \cdots < \eta_{q_2} \leq m' - 1$.

In the following, we will show that with some probability any internal state variable of the NFSR can be computed from the value of the NFSR state variables at a fixed time instant t_0 and of some keystream bits, under the condition that the LFSR initial state $L^{(0)}$ is known. To avoid the influence of masking of the round key bit in the NFSR update function, we instead use recursively the output function, i.e., Eq.(11), to derive relationships between these variables and keystream bits, by extending the technique in [5]. Compared to [31], the pseudo-linearity property of the filtering function h is not required in our following process.

Here we only discuss the case $\eta_{q_2} > \delta_{s_2}$ to illustrate the process, while the other cases can be handled by induction in a similar way. In this case, η_{q_2} is the highest tap value of the NFSR variables $(n_t, \cdots, n_{t+m'-1})$ involved in the keystream bit z_t. Assuming that the LFSR initial state $L^{(0)} = (l_0, \cdots, l_{m-1})$ is known, we will express each NFSR state variable n_i, $i \geq t_0 + m'$, as a function of the NFSR state variables $N^{(t_0)} = (n_{t_0}, \cdots, n_{t_0+m'-1})$ and of some keystream bits.

We first consider how to express $n_{t_0+m'}$. According to Eq.(11), $z_{t_0+m'-\eta_{q_2}}$ is the first keystream bit which certainly depends on $n_{t_0+m'}$, thus we have

$$n_{t_0+m'} = z_{t_0+m'-\eta_{q_2}} \oplus \bigoplus_{b_2 \in \mathbb{B}_2 \setminus \{\eta_{q_2}\}} \underline{n_{t_0+m'-\eta_{q_2}+b_2}} \oplus h\left(L^{(t_0+m'-\eta_{q_2})}_{\mathbb{T}_{h,L}}, \underline{N^{(t_0+m'-\eta_{q_2})}_{\mathbb{T}_{h,N}}}\right) \oplus \bigoplus_{b_1 \in \mathbb{B}_1} l_{t_0+m'-\eta_{q_2}+b_1}.$$

Next we assume that for all i with $t_0 + m' \leq i < t_0 + m' + j$, all the bits n_i have be expressed as a function of the NFSR state variables at time instant t_0 and of some keystream bits. Note that $z_{t_0+m'-\eta_{q_2}+j}$ is the first keystream bit which is certainly dependent on $n_{t_0+m'+j}$, which indicates that

$$\begin{aligned} n_{t_0+m'+j} = {} & z_{t_0+m'-\eta_{q_2}+j} \bigoplus_{b_2 \in \mathbb{B}_2 \setminus \{\eta_{q_2}\}} \underline{n_{t_0+m'-\eta_{q_2}+j+b_2}} \oplus h\left(L^{(t_0+m'-\eta_{q_2}+j)}_{\mathbb{T}_{h,L}}, \underline{N^{(t_0+m'-\eta_{q_2}+j)}_{\mathbb{T}_{h,N}}}\right) \\ & \oplus \bigoplus_{b_1 \in \mathbb{B}_1} l_{t_0+m'-\eta_{q_2}+j+b_1}. \end{aligned}$$

That is, the variables $n_{t_0+m'+j}$ is expressed as a function of a keystream bit $z_{t_0+m'-\eta_{q_2}+j}$ and of the NFSR variables $n_{t_0+m'+i}$ with $i < j$. By induction assumption $n_{t_0+m'+j}$ can be expressed as a function of the NFSR state variables at time instant t_0 and of keystream bits $\{z_{t_0+m'-\eta_{q_2}+j} | j \geq 0\}$.

If we perform recursively this substitution process over Θ times, we can compute the value of Θ variables $\{n_{t_0+m'+j} | \, j = 0, \cdots, \Theta - 1\}$ from the value of the NFSR state variables at time instant t_0 and of some keystream bits $\{z_{t_0+m'-\eta_{q_2}+j} | \, j = 0, \cdots, \Theta - 1\}$, where Θ is a parameter to be determined later and $d \leq \Theta \leq d + (m' + m)$.

Regarding the case that $\eta_{q_2} \leq \delta_{s_2}$, when we express $n_{t_0+m'}$, some terms of $h_1(L^{(t_0+m'-\eta_{q_2})}_{\mathbb{T}_{h,L}}, \underline{N^{(t_0+m'-\eta_{q_2})}_{\mathbb{T}_{h,N}})}$ contains the later internal state variables $n_{t_0+m'+j}$, $j \geq 1$. To derive successfully relations by induction, the outcome of these terms is guessed, by extending the technique proposed in [22]. Let p_g be the probability of the most probable guess, and we simply search in the keystream for the place where this is satisfied. In this case, the probability that all the guesses for these terms are right is $\rho = p_g^{\Theta}$ and the expect length of the keystream is around $\frac{1}{\rho} = p_g^{-\Theta}$.

Algorithm of recovering the NFSR State and the Round Key Bits Now we present the process of recovering the NFSR state at some time instant and the round key bits as Algorithm 2. Before giving into details of the algorithm, let us take a little bit about how it works.

Algorithm 2 Recovery of the NFSR State at Some Time Instant and the Round Key Bits

```
Input: A set of candidates for the LFSR initial state, $\mathbb{C}$; Some keystream bits $\{z_t\}$.
    Parameter: The number of ticks for state checking, $m' + \theta$, and setting $\theta = \log(|\mathbb{C}|)$; The probability that
    all the internal state variables of NFSR $\{n_{t+m'+i}\}_{i=0}^{d+m'+\theta-1}$ are correctly computed, $\rho$.
Output: The whole state of the NFSR at some time instant $t_0$, $N^{(t_0)}$; The round key bits, $\{k'_{t_0+i}\}_{i=0}^{d-1}$; The
    correct initial state of the LFSR, $L^{(0)}$.
 1: for all $\bar{L}^{(0)} \in \mathbb{C}$ do
 2:     for all $t \in \{0, \cdots, \frac{1}{\rho} - 1\}$ do
 3:         for all $N^{(t)} \in \{0,1\}^{m'}$ do
          /* Subroutine for state checking */
 4:             for $i = 0$ to $d - 1$ do
 5:                 Compute $n_{t+m'+i}$ from $z_{t+m'-\eta_{q_2}}, \cdots, z_{t+m'-\eta_{q_2}+i}$ with the technique described in Section ??;
 6:                 Compute $k'_{t+i} = n_{t+m'+i} \oplus l_{t+i} \oplus c_{t+i} \oplus g(N^{(t+i)})$ and store it at the $i$-th position of the array
                        $\xi$, i.e., $\xi[i] = k'_{t+i}$.
 7:             end for
 8:             for $i = 0$ to $m' + \theta - 1$ do
 9:                 Compute $n_{t+m'+d+i}$ from $z_{t+m'-\eta_{q_2}}, \cdots, z_{t+m'-\eta_{q_2}+d+i}$;
10:                 Compute $e_i = n_{t+m'+d+i} \oplus l_{t+d+i} \oplus c_{t+d+i} \oplus g(N^{(t+d+i)})$;
11:                 if $e_i \neq \xi[i]$ then
12:                     goto Step 3.
13:                 end if
14:             end for
15:             $t_0 = t$;
16:             return The current guessed state of NFSR at time instant $t_0$, $N^{(t_0)}$; the current initial state of
                    LFSR, $L^{(0)}$; the round key bits, $\xi[i], i = 0, 1, \cdots, d - 1$.
17:         end for
18:     end for
19: end for
```

After identifying the candidates of the LFSR initial state by Algorithm 1, we can get the value of $\{n_{t+m'+i} | i = 0, \cdots, d + m' + \theta - 1\}$ from the value of the NFSR state variables at time instant t and of some keystream bits with the technique described previously. Moreover, $d + m' + \theta - 1$ consecutive round key bits $\{k'_{t+i} | i = 0, \cdots, d + m' + \theta - 1\}$ can be computed from the full internal state by using the NFSR update function. Due to the fact that the $RKF(\cdot)$ is periodic, we have $k'_{t+i} = k'_{t+d+i}$, for $i = 0, \cdots, m' + \theta - 1$. Since the size of the NFSR state for small state stream ciphers is often much smaller than its security margin, exhaustively searching over all the possible values of the NFSR state is always feasible. Therefore, we will try out all the possible values of the NFSR state at time instant t and carry out the above process to identify the correct value. To exclude $2^{m'} - 1$ wrong values of the NFSR state, we need m' ticks at most for the period checking of the round key bits and another θ ticks for the LFSR initial state. Once we recover successfully the NFSR state at some time instant, the round key bits will also be restored.

3.4 Analysis of Time and Data Complexities

In this subsection, we give the complexity analysis of our divide-and-conquer fast correlation attacks. When restoring the LFSR initial state, a threshold th_p is introduced to have a more flexible attack. As soon as th_p is set, we can compute the two probabilities p_c and p_w which indicate the probabilities that the correct value and a wrong value will be chosen as a candidate of the LFSR initial state, respectively. According to the different values of p_c and p_w, four attack scenarios could be encountered. Here we only give the scenario that will be used in the analysis of concrete small state stream ciphers later.

Attack Scenario: $p_c > 0.999$ and $2p_w < 2^{-m}$. In this scenario, the correct value $L^{(0)}$ of the m-bit LFSR initial state will always be chosen as a candidate, meanwhile none of wrong ones could be chosen in Algorithm 1, where doubling the probability is due to that there are two majority polls at part 3. This means that there is only one candidate for the LFSR initial state which will be inputted into Algorithm 2.

With the time complexity of Algorithm 1 and Algorithm 2 denoted by T_1 and T_2 respectively, we have the following theorem to estimate the time and data complexities of recovering the full internal state and the round key bits for the target stream cipher.

Theorem 3. *Let m and m' be the sizes of the LFSR and NFSR from the target stream cipher, respectively. Assuming that for the sum of keystream bits $\oplus_{i\in\mathbb{T}_z} z_{t+i}$, there are r linear approximate equations of the LFSR bits whose absolute value of bias is greater than $\frac{\epsilon^c}{2}$ $(\epsilon^c > 0)$, then the data complexity is $D = |\mathbb{T}_z| \frac{2^{\frac{\beta+7}{2}}\sqrt{m\ln 2}}{\sqrt{r}(\epsilon^c)^2}$, and the time complexity is $T = T_1 + T_2$, where T_1 and T_2 is the time complexities of Algorithm 1 and 2 respectively. With $th = \frac{2^{\frac{\beta+5}{2}}\sqrt{m\ln 2}}{\sqrt{r}\epsilon^c}$ and th_p determined as in the above scenario, the corresponding time complexities T_1 and T_2 are listed:*

Attack Scenario: *$T_1 = \frac{2^{\frac{\beta+7}{2}}\sqrt{m\ln 2}}{\sqrt{r}(\epsilon^c)^2} + r2^{m+1-\beta}p_1$ and $T_2 = p_g^{-(d+m')} \times 2^{m'} \times (d+m')$,*

where β $(0 \le \beta \le log(r))$ is the size of bypassed bits of the LFSR initial state which is chosen to make a balance between the two dominant parts of T_1, i.e., exhaustively searching the values of β such that T_1 achieves the minimum, p_g is the probability that one internal state variable of the NFSR is correctly computed during Algorithm 2 and d is the period of the round key bits and

$$p_1 = Q\left(\left(m\ln 2 \times r^{-1}2^{\beta+3}\right)^{\frac{1}{4}}\right).$$

Proof. Here we only give details of the proof for **Attack Scenario**, and the other scenarios can be treated in a similar way.

For the data complexity, as illustrated in Theorem 2, the sufficient condition to identify the unique correct initial state of the LFSR is $\Omega \ge \frac{2^{\frac{\beta+7}{2}}\sqrt{m\ln 2}}{\sqrt{r}(\epsilon^c)^2}$. Thus we can safely set $\Omega = \frac{2^{\frac{\beta+7}{2}}\sqrt{m\ln 2}}{\sqrt{r}(\epsilon^c)^2}$. Accordingly, the data complexity is $D = |\mathbb{T}_z|\Omega = |\mathbb{T}_z|\frac{2^{\frac{\beta+7}{2}}\sqrt{m\ln 2}}{\sqrt{r}(\epsilon^c)^2}$ keystream bits.

For the time complexity, we first find all the linear masks with absolute value of bias is greater than $\frac{\epsilon^c}{2}$ and compute the inverse of corresponding matrices $\{F_{\boldsymbol{u}_j}\}_{j=1}^r$ in preparation. Since the time cost in preparation is practical, we will not add it into the following estimation of the time complexity. With exploiting Theorem 1, we get the parity-check equations like $(L^{(0)} \times F_{\boldsymbol{u}_j}) \cdot \boldsymbol{g}_{dt'} \oplus \hat{z}_{dt'} \oplus \hat{c}_{dt'} \oplus \hat{k}_0 = e_{dt,j}$, $j = 1, \cdots, r$. For all the possible values of the LFSR initial state, the parity-check equations constructed from $\boldsymbol{g}_{dt'}$ and $\hat{z}_{dt'} \oplus \hat{c}_{dt'}$ are evaluated and a threshold is introduced to filter these state values. Subsequently, we multiply the inverse of matrices $F_{\boldsymbol{u}_j}^{-1}$ into all the state values which pass the statistical test and make two majority polls independently to identify candidates of the LFSR initial state. This process is detailed in Algorithm 1 and the complexity cost is counted precisely as follows. The preparation of $w(\boldsymbol{a})$ at part 1 will take a time complexity of Ω, while FWHT for $w(\boldsymbol{a})$ needs time complexity of $(m-\beta)2^{m-\beta}$ with memory $2^{m-\beta}$ at part 2. During part 3 of Algorithm 1, there are in total $(2^{m-\beta} \times 2p_1 + r2^{-\beta}p_2) \times r$ operations to multiply the whole set of matrices $\{F_{\boldsymbol{u}_j}^{-1}\}_{j=1}^r$ to all the state values which pass the statistical test for two majority polls. Therefore, the time complexity of Algorithm 1 is computed as $T_1 = \Omega + (m-\beta)2^{m-\beta} + r2^{m+1-\beta}p_1 + r^2 2^{-\beta}p_2$. According to Theorem 2, we can set $th = \frac{1}{2}\Omega\epsilon^c$ and $\Omega = \frac{2^{\frac{\beta+7}{2}}\sqrt{m\ln 2}}{\sqrt{r}(\epsilon^c)^2}$ for Algorithm 1 to identify successfully the LFSR initial state. Then, we have that $th = \frac{2^{\frac{\beta+5}{2}}\sqrt{m\ln 2}}{\sqrt{r}\epsilon^c}$, $p_1 = Q(\frac{th}{\sqrt{\Omega}}) = Q(\frac{1}{2}\sqrt{\Omega}\epsilon^c) = Q((m\ln 2 \times r^{-1}2^{\beta+3})^{\frac{1}{4}})$ and $p_2 = Q(\frac{th-\Omega\epsilon^c}{\sqrt{\Omega}}) = Q(-\frac{1}{2}\sqrt{\Omega}\epsilon^c) = 1 - p_1$. In the useful attack parameters, since the size of the LFSR state is always much smaller in Grain-like small state stream ciphers, $(m-\beta)2^{m-\beta} << r2^{m+1-\beta}p_1$ and $(m-\beta)2^{m-\beta} << \frac{2^{\frac{\beta+7}{2}}\sqrt{m\ln 2}}{\sqrt{r}(\epsilon^c)^2}$ always hold and we regard it negligible. Therefore, we only need to set β such that a balance between time complexities of part 1 and part 3 is achieved. Besides, in part 3 of Algorithm 1, since $r^2 2^{-\beta}p_2$ is significantly smaller than $r2^{m+1-\beta}p_1$ in the useful attack parameters, we treat it as negligible. Therefore, the time complexity becomes $T_1 = \frac{2^{\frac{\beta+7}{2}}\sqrt{m\ln 2}}{\sqrt{r}(\epsilon^c)^2} + r2^{m+1-\beta}p_1$ and β is exhaustively searched such that T_1 achieves the minimum. In this scenario, th_p is chosen such that $p_c > 0.999$ and $2p_w < 2^{-m}$. Therefore, only one candidate will be inputted into Algorithm 2. Then, we only need to exhaustively search all possible values of the NFSR state at some time instant and perform the subroutine of state checking to find the correct one. Since $d + m'$ internal state variables are correctly computed with probability $\rho = p_g^{d+m'}$, the expected length of the keystream to search is $\frac{1}{\rho} = p_g^{-(d+m')}$, where p_g is the probability that one internal state variable of the NFSR is correctly computed. After

Algorithm 2 completes, the full internal state and d consecutive round key bits can be restored. This process will take a time complexity of $p_g^{-(d+m')} \times 2^{m'} \times (d+m')$. □

Remark 2. According to Theorem 2, we can set $th = \frac{1}{2}\Omega\epsilon^c$ and $\Omega = \frac{4(Q^{-1}(\frac{1}{2}(1-\sqrt{A(2-A)})))^2}{(\epsilon^c)^2}$ in the precise mode, where $A = \frac{2^\beta m \ln 2}{r}$ and Q^{-1} is the inverse function of Q-function. Therefore, we have the corresponding Theorem 3 in the precise mode with the data complexity of $D = |\mathbb{T}_z| \frac{4(Q^{-1}(\frac{1}{2}(1-\sqrt{A(2-A)})))^2}{(\epsilon^c)^2}$ and time complexity $T_1 = \frac{4(Q^{-1}(\frac{1}{2}(1-\sqrt{A(2-A)})))^2}{(\epsilon^c)^2} + r2^{m+1-\beta}p_1$, where $p_1 = \frac{1}{2}(1-\sqrt{A(2-A)})$.

Remark 3. Here, we explain for the Assumed Property 2 of the generic model. When the size of bypassed bits β is equal to 0, we need $\Omega = \frac{2^{\frac{7}{2}}\sqrt{m\ln 2}}{\sqrt{r}(\epsilon^c)^2}$ parity-check equations to carry out Algorithm 1 of our attack methods. To have a more efficient attack than exhaustively searching the secret key, the necessary condition should be satisfied as $\Omega = \frac{2^{\frac{7}{2}}\sqrt{m\ln 2}}{\sqrt{r}(\epsilon^c)^2} \leq 2^\kappa$. The Assumed Property 2 is derived by plugging the rough estimation $r = 2^{s_1 \times |\mathbb{T}_z|}$ and $\epsilon^c = 2^{(s_2+1)\times|\mathbb{T}_z|+q_2} \times \epsilon_h^{|\mathbb{T}_z|} \times \epsilon_{g^*}^{q_2}$ in this inequality.

4 Applications: Plantlet Case

In this section, our divide-and-conquer fast correlation attacks will be applied to Plantlet. First, we show how to derive the linear approximate equations which are used to construct the desirable parity-check equations for Algorithm 1. Then under the condition that the LFSR initial state is known, we give a cryptanalysis of the degraded system of Plantlet for Algorithm 2. As a result, the complexities of recovering the secret key are presented in the following.

Before showing details of our attacks, we provide a brief description of Plantlet in the keystream generation phase. Plantlet is a bit-oriented stream cipher and utilizes an 80-bit secret key $K = (k_0, \cdots, k_{79})$ and a 90-bit public initial value $IV = (iv_0, \cdots, iv_{89})$ to generate the keystream. For Plantlet, there are four parts involved, a 61-bit LFSR whose state at time instant t is denoted by $L^{(t)} = (l_t, \cdots, l_{t+60})$, a linked 40-bit NFSR whose state at time instant t is denoted by $N^{(t)} = (n_t, \cdots, n_{t+39})$, an 80-bit fixed key register and a 9-bit counter register $C_c = (c_t^0, \cdots, c_t^8)$ allocated for the initialization/keystream generation. The first seven bits $(c_t^0, \cdots, c_t^6)$ of the counter are used to count cyclically from 0 to 79, i.e., it resets to 0 after 79 is reached. The two most significant bits realize a 2-bit counter to determine the number of elapsed clock cycles in the initialization phase, i.e., it is triggered by the resets of the lower 7 bits.

The LFSR is updated independently and recursively by a linear function as $l_{t+61} = f(L^{(t)}) = l_t \oplus l_{t+14} \oplus l_{t+20} \oplus l_{t+34} \oplus l_{t+43} \oplus l_{t+54}$. The NFSR is updated as defined in the following:

$$\begin{aligned} n_{t+40} &= k'_t \oplus c_t^4 \oplus l_t \oplus g(N^{(t)}) \\ &= k'_t \oplus c_t^4 \oplus l_t \oplus n_t \oplus n_{t+13} \oplus n_{t+19} \oplus n_{t+35} \oplus n_{t+39} \\ &\oplus n_{t+2}n_{t+25} \oplus n_{t+3}n_{t+5} \oplus n_{t+7}n_{t+8} \oplus n_{t+14}n_{t+21} \\ &\oplus n_{t+16}n_{t+18} \oplus n_{t+22}n_{t+24} \oplus n_{t+26}n_{t+32} \\ &\oplus n_{t+33}n_{t+36}n_{t+37}n_{t+38} \\ &\oplus n_{t+10}n_{t+11}n_{t+12} \oplus n_{t+27}n_{t+30}n_{t+31}, \end{aligned} \tag{12}$$

where k'_t is the round key bit and c_t^4 is the counter bit from C_c at time instant t. The round key function simply cyclically selects the next key bit, i.e., $k'_t = RKF(K,t) = k_{(t \bmod 80)}$, $t \geq 0$.

The filtering function is defined as

$$\begin{aligned} h\left(L^{(t)}_{\mathbb{T}_{h,L}}, N^{(t)}_{\mathbb{T}_{h,N}}\right) &= n_{t+4}l_{t+6} \oplus l_{t+8}l_{t+10} \oplus l_{t+32}l_{t+17} \\ &\oplus l_{t+19}l_{t+23} \oplus n_{t+4}l_{t+32}n_{t+38}, \end{aligned}$$

where the two subsets

$$L^{(t)}_{\mathbb{T}_{h,L}} = (l_{t+6}, l_{t+8}, l_{t+10}, l_{t+17}, l_{t+19}, l_{t+23}, l_{t+32})$$

and $N^{(t)}_{\mathbb{T}_{h,N}} = (n_{t+4}, n_{t+38})$. The entire output function is determined by

$$z_t = h\left(L^{(t)}_{\mathbb{T}_{h,L}}, N^{(t)}_{\mathbb{T}_{h,N}}\right) \oplus l_{t+30} \oplus \bigoplus_{b\in\mathbb{B}} n_{t+\mathbb{B}}, \tag{13}$$

where $\mathbb{B} = \{1, 6, 15, 17, 23, 28, 34\}$.

It is obviously that the Assumed Property 1 holds for Plantlet and the $RKF(\cdot)$ is periodic with a cycle of minimum length $d = 80$, i.e., $k_{t+80} = k_t$. As for the Assumed Property 2, we give a more accurate analysis in the following discuss.

4.1 Deriving Linear Approximate Equations

In this subsection, we expect to estimate the number and bias of different linear approximate equations which are used to construct the desirable parity-check equations. First, we derive the linear approximate representations for the sum of some keystream bits. Then, we could evaluate the number and bias of different linear approximate equations by exhaustively searching all the possible representations.

Linear Approximate Representations Considering the best linear approximation of the NFSR update function Eq.(12) with bias $2^{-9.02}$ as follows,

$$n_{t+40} \approx k'_t \oplus c^4_t \oplus l_t \oplus n_t \oplus n_{t+13} \oplus n_{t+19} \oplus n_{t+35} \oplus n_{t+39},$$

we choose the set of taps as $\mathbb{T}_z = \{0, 13, 19, 35, 39, 40\}$. Then, we have

$$\bigoplus_{i\in\mathbb{T}_z} z_{t+i} = \bigoplus_{i\in\mathbb{T}_z} l_{t+i+30} \oplus \bigoplus_{b\in\mathbb{B}} l_{t+b} \oplus \bigoplus_{i\in\mathbb{T}_z} h(L^{(t+i)}_{\mathbb{T}_{h,L}}, N^{(t+i)}_{\mathbb{T}_{h,N}}) \oplus \bigoplus_{b\in\mathbb{B}} g^*(N^{(t+b)}) \oplus \bigoplus_{b\in\mathbb{B}} k'_{t+b} \oplus \bigoplus_{b\in\mathbb{B}} c^4_{t+b},$$

where $g^*(N^{(t)}) = n_t \oplus n_{t+13} \oplus n_{t+19} \oplus n_{t+35} \oplus n_{t+39} \oplus g(N^{(t)})$ and it has the same bias, i.e., $\Pr[g^*(N^{(t)}) = 0] = \frac{1}{2} + 2^{-9.02}$. Let $\boldsymbol{a}_i \in \{0,1\}^9$ be the input linear mask for h function at time instant $t+i$, i.e., $\boldsymbol{a}_i = (a_i[0], \cdots, a_i[8])$. Then

$$h(L^{(t+i)}_{\mathbb{T}_{h,L}}, N^{(t+i)}_{\mathbb{T}_{h,N}}) \approx \boldsymbol{a}_i[0-6] \cdot \left(L^{(t+i)}_{\mathbb{T}_{h,L}}\right)^T \oplus \boldsymbol{a}_i[7-8] \cdot \left(N^{(t+i)}_{\mathbb{T}_{h,N}}\right)^T$$

with bias $\epsilon_{h,i}(\boldsymbol{a}_i) = \pm 2^{-5}$ or 0. Due to $|\mathbb{T}| = 6$, there are 6 active h functions which need to be approximated. Let $\boldsymbol{a}_{\mathbb{T}_z} \in \{0,1\}^{9\times 6}$ be the concatenated linear mask, i.e, $\boldsymbol{a}_{\mathbb{T}_z} = (\boldsymbol{a}_0, \boldsymbol{a}_{13}, \boldsymbol{a}_{19}, \boldsymbol{a}_{35}, \boldsymbol{a}_{39}, \boldsymbol{a}_{40})$. The total bias of all the approximated h functions is computed as $\epsilon_{h,\mathbb{T}_z}(\boldsymbol{a}_{\mathbb{T}_z}) = 2^{6-1} \times \prod_{i\in\mathbb{T}_z} \epsilon_{h,i}(\boldsymbol{a}_i)$ because of the piling-up lemma. Note that if we use $\boldsymbol{a}_i$ such that $\epsilon_{h,i}(\boldsymbol{a}_i) = 0$ for any $i \in \mathbb{T}_z$, then $\epsilon_{h,\mathbb{T}_z}$ is equal to zero. Otherwise, $\epsilon_{h,\mathbb{T}_z} = \pm 2^{-25}$.

Let

$$\epsilon_{g^*,\mathbb{B}}(\boldsymbol{a}_{\mathbb{T}_z}) = \Pr\left[\bigoplus_{i\in\mathbb{T}_z} \boldsymbol{a}_i[7-8] \cdot \left(N^{(t+i)}_{\mathbb{T}_{h,N}}\right)^T \oplus \bigoplus_{b\in\mathbb{B}} g^*(N^{(t+b)}) = 0\right] - \frac{1}{2}$$

and the bias is independent on $\boldsymbol{a}_i[0-6]$ for all $i \in \mathbb{T}_z$. We expect that the bias $\epsilon_{g^*,\mathbb{B}}(\boldsymbol{a}_{\mathbb{T}_z})$ is high and only care about the situation where the bias is not equal to zero. When $\boldsymbol{a}_i[7,8] = \boldsymbol{0}$ for all $i \in \mathbb{T}_z$, we have $\epsilon_{g^*,\mathbb{B}}(\boldsymbol{a}_{\mathbb{T}_z}) = \Pr\left[\bigoplus_{b\in\mathbb{B}} g^*(N^{(t+b)}) = 0\right] - \frac{1}{2}$. To compute the bias, we choose some variables such that $\bigoplus_{b\in\mathbb{B}} g^*(N^{(t+b)})$ can be divided into some pieces which have fewer variables and no common variables between them when the chosen variables are fixed. Then we evaluate the bias by applying the piling-up lemma to all the pieces when we try out all the possible values of chosen variables. Therefore, the bias $\epsilon_{g^*,\mathbb{B}}(\boldsymbol{a}_{\mathbb{T}_z})$ could be derived as soon as we add up the biases in all possible cases of the chosen variables.

Similarly, the bias $\epsilon_{g^*,\mathbb{B}}(\boldsymbol{a}_{\mathbb{T}_z})$ can be evaluated when $\boldsymbol{a}_i[7,8] \neq \boldsymbol{0}$ for any $i \in \mathbb{T}_z$. If one of $a_{19}[8]$, $a_{35}[8]$, $a_{39}[8]$ and $a_{40}[8]$ is 1, the bias is always 0 because n_{t+57}, n_{t+73}, n_{t+77} and n_{t+78} are not involved in $\bigoplus_{b\in\mathbb{B}} g^*(N^{(t+b)})$. We summarize $\epsilon_{g^*,\mathbb{B}}(\boldsymbol{a}_{\mathbb{T}_z})$ when $a_{19}[8]$, $a_{35}[8]$, $a_{39}[8]$ and $a_{40}[8]$ are 0 in Table 2.

For any fixed $\boldsymbol{a}_{\mathbb{T}_z}$, we can derive the following linear approximate representation

$$\bigoplus_{i\in\mathbb{T}_z} z_{t+i} \approx \bigoplus_{i\in\mathbb{T}_z} l_{t_i+30} \oplus \bigoplus_{b\in\mathbb{B}} l_{t+b} \oplus \bigoplus_{i\in\mathbb{T}_z} \boldsymbol{a}_i[0-6] \cdot \left(L^{(t+i)}_{\mathbb{T}_{h,L}}\right)^T \oplus \bigoplus_{b\in\mathbb{B}} k'_{t+b} \oplus \bigoplus_{b\in\mathbb{B}} c^4_{t+b}$$

and its bias is evaluated as $2 \times \epsilon_{h,\mathbb{T}_z}(\boldsymbol{a}_{\mathbb{T}_z}) \times \epsilon_{g^*,\mathbb{B}}(\boldsymbol{a}_{\mathbb{T}_z})$.

Table 2. Summary of bias when $\boldsymbol{a}_i[7-8]$ is fixed. Let $*$ be the arbitrary bit.

$a_0[7]$	$a_0[8]$	$a_{13}[7]$	$a_{13}[8]$	$a_{19}[7]$	$a_{35}[7]$	$a_{39}[7]$	$a_{40}[7]$	$\epsilon_{g^*,\mathbb{B}}$
0	0	0	0	0	0	0	0	$+2^{-21.87}$
1	0	0	0	0	0	0	0	$+2^{-21.87}$
0	1	0	0	0	0	0	0	$+2^{-21.87}$
1	1	0	0	0	0	0	0	$+2^{-21.87}$
0	0	1	0	0	0	0	0	$+2^{-21.87}$
1	0	1	0	0	0	0	0	$+2^{-21.87}$
0	1	1	0	0	0	0	0	$+2^{-21.87}$
1	1	1	0	0	0	0	0	$+2^{-21.87}$
0	0	0	1	0	0	0	0	$-2^{-25.61}$
1	0	0	1	0	0	0	0	$-2^{-25.61}$
0	1	0	1	0	0	0	0	$-2^{-25.61}$
1	1	0	1	0	0	0	0	$-2^{-25.61}$
0	0	1	1	0	0	0	0	$-2^{-25.61}$
1	0	1	1	0	0	0	0	$-2^{-25.61}$
0	1	1	1	0	0	0	0	$-2^{-25.61}$
1	1	1	1	0	0	0	0	$-2^{-25.61}$
0	0	0	0	0	0	1	0	$+2^{-22.31}$
1	0	0	0	0	0	1	0	$+2^{-22.31}$
0	1	0	0	0	0	1	0	$+2^{-22.31}$
1	1	0	0	0	0	1	0	$+2^{-22.31}$
0	0	1	0	0	0	1	0	$+2^{-22.31}$
1	0	1	0	0	0	1	0	$+2^{-22.31}$
0	1	1	0	0	0	1	0	$+2^{-22.31}$
1	1	1	0	0	0	1	0	$+2^{-22.31}$
0	0	0	1	0	0	1	0	$+2^{-24.29}$
1	0	0	1	0	0	1	0	$+2^{-24.29}$
0	1	0	1	0	0	1	0	$+2^{-24.29}$
1	1	0	1	0	0	1	0	$+2^{-24.29}$
0	0	1	1	0	0	1	0	$+2^{-24.29}$
1	0	1	1	0	0	1	0	$+2^{-24.29}$
0	1	1	1	0	0	1	0	$+2^{-24.29}$
1	1	1	1	0	0	1	0	$+2^{-24.29}$
0	0	0	0	0	0	0	1	$+2^{-21.87}$
1	0	0	0	0	0	0	1	$+2^{-21.87}$
0	1	0	0	0	0	0	1	$+2^{-21.87}$
1	1	0	0	0	0	0	1	$+2^{-21.87}$
0	0	1	0	0	0	0	1	$+2^{-21.87}$
1	0	1	0	0	0	0	1	$+2^{-21.87}$
0	1	1	0	0	0	0	1	$+2^{-21.87}$
1	1	1	0	0	0	0	1	$+2^{-21.87}$
0	0	0	1	0	0	0	1	$-2^{-25.61}$
1	0	0	1	0	0	0	1	$-2^{-25.61}$
0	1	0	1	0	0	0	1	$-2^{-25.61}$
1	1	0	1	0	0	0	1	$-2^{-25.61}$
0	0	1	1	0	0	0	1	$-2^{-25.61}$
1	0	1	1	0	0	0	1	$-2^{-25.61}$
0	1	1	1	0	0	0	1	$-2^{-25.61}$
1	1	1	1	0	0	0	1	$-2^{-25.61}$
0	0	0	0	0	0	1	1	$+2^{-22.31}$
1	0	0	0	0	0	1	1	$+2^{-22.31}$
0	1	0	0	0	0	1	1	$+2^{-22.31}$
1	1	0	0	0	0	1	1	$+2^{-22.31}$
0	0	1	0	0	0	1	1	$+2^{-22.31}$
1	0	1	0	0	0	1	1	$+2^{-22.31}$
0	1	1	0	0	0	1	1	$+2^{-22.31}$
1	1	1	0	0	0	1	1	$+2^{-22.31}$
0	0	0	1	0	0	1	1	$+2^{-24.29}$
1	0	0	1	0	0	1	1	$+2^{-24.29}$
0	1	0	1	0	0	1	1	$+2^{-24.29}$
1	1	0	1	0	0	1	1	$+2^{-24.29}$
0	0	1	1	0	0	1	1	$+2^{-24.29}$
1	0	1	1	0	0	1	1	$+2^{-24.29}$
0	1	1	1	0	0	1	1	$+2^{-24.29}$
1	1	1	1	0	0	1	1	$+2^{-24.29}$
$*$	$*$	$*$	$*$	1	$*$	$*$	$*$	0
$*$	$*$	$*$	$*$	$*$	1	$*$	$*$	0

Linear Approximate Equations From the linear approximate representations derived at the previous part, we can derive the following linear approximate equation with some fixed linear mask $\boldsymbol{u}$

$$\bigoplus_{i\in\mathbb{T}_z} z_{t+i} \approx L^{(0)} \cdot \left(F^t \times \boldsymbol{u}\right) \oplus \bigoplus_{b\in\mathbb{B}} k'_{t+b} \oplus \bigoplus_{b\in\mathbb{B}} c^4_{t+b},$$

where $\boldsymbol{u} \in \{0,1\}^{61}$ is a column vector. If different $\boldsymbol{a}_{\mathbb{T}_z}$'s derive the same linear mask $\boldsymbol{u}$, the corresponding biases should be added up to get the bias of $\boldsymbol{u}$, i.e., $\epsilon_{\boldsymbol{u}} = \sum_{\{\boldsymbol{a}_{\mathbb{T}_z} | U(\boldsymbol{a}_{\mathbb{T}_z}) = \boldsymbol{u}\}} 2 \times \epsilon_{h,\mathbb{T}_z}(\boldsymbol{a}_{\mathbb{T}_z}) \times \epsilon_{g^*,\mathbb{B}}(\boldsymbol{a}_{\mathbb{T}_z})$, where

$$
\begin{aligned}
U(\boldsymbol{a}_{\mathbb{T}_z}) = \bigoplus_{i \in \mathbb{T}_z} ((&a_i[0] \cdot \boldsymbol{g}_{i+6} \oplus a_i[1] \cdot \boldsymbol{g}_{i+8} \oplus a_i[2] \cdot \boldsymbol{g}_{i+10} \oplus a_i[3] \cdot \boldsymbol{g}_{i+17} \oplus a_i[4] \cdot \boldsymbol{g}_{i+19} \oplus a_i[5] \cdot \boldsymbol{g}_{i+23} \oplus a_i[6] \\
&\cdot \boldsymbol{g}_{i+32}) \oplus \boldsymbol{g}_{i+30}) \oplus \bigoplus_{b \in \mathbb{B}} \boldsymbol{g}_b.
\end{aligned}
$$

Clearly, since the function $U(\boldsymbol{a}_{\mathbb{T}_z})$ is independent on $\boldsymbol{a}_i[7,8]$ for any $i \in \mathbb{T}_z$, we need to sum up all biases with a non-zero $\epsilon_{g^*,\mathbb{B}}$ summarized in Table 2, where $\boldsymbol{a}_i[0,\cdots,6]$ is identical and only $\boldsymbol{a}_i[7,8]$ varies for $i \in \mathbb{T}_z$. Let V be a subset of $\{0,1\}^{9\times 6}$ whose elements are 64 corresponding vectors $\boldsymbol{a}_{\mathbb{T}_z}$ with non-zero $\epsilon_{g^*,\mathbb{B}}$ in Table 2. Moreover, there are some special relationships. When we focus on $a_0[4]$ and $a_{13}[0]$, corresponding 61-bit column vectors are identical because $\boldsymbol{g}_{0+19} = \boldsymbol{g}_{13+6} = \boldsymbol{g}_{19}$. That means $(a_0[4], a_{13}[0]) = (0,0)$ and $(a_0[4], a_{13}[0]) = (1,1)$ derive the same $\boldsymbol{u}$, and $(a_0[4], a_{13}[0]) = (1,0)$ and $(a_0[4], a_{13}[0]) = (0,1)$ also derive the same $\boldsymbol{u}$. We have 7 such relationships as showed in the following.

- $a_0[4]$ and $a_{13}[0]$ (since $\boldsymbol{g}_{0+19} = \boldsymbol{g}_{13+6} = \boldsymbol{g}_{19}$).
- $a_0[5]$ and $a_{13}[2]$ (since $\boldsymbol{g}_{0+23} = \boldsymbol{g}_{13+10} = \boldsymbol{g}_{23}$).
- $a_0[6]$ and $a_{13}[4]$ (since $\boldsymbol{g}_{0+32} = \boldsymbol{g}_{13+19} = \boldsymbol{g}_{32}$).
- $a_{13}[5]$ and $a_{19}[3]$ (since $\boldsymbol{g}_{13+23} = \boldsymbol{g}_{19+17} = \boldsymbol{g}_{36}$).
- $a_{13}[6]$ and $a_{35}[2]$ (since $\boldsymbol{g}_{13+32} = \boldsymbol{g}_{35+10} = \boldsymbol{g}_{45}$).
- $a_{35}[2]$ and $a_{39}[0]$ (since $\boldsymbol{g}_{35+10} = \boldsymbol{g}_{39+6} = \boldsymbol{g}_{45}$).
- $a_{35}[5]$ and $a_{39}[4]$ (since $\boldsymbol{g}_{35+23} = \boldsymbol{g}_{39+19} = \boldsymbol{g}_{58}$).

Let $\boldsymbol{w}_1, \cdots, \boldsymbol{w}_7$ be the (9×6)-bit vectors generated by the above relationships with the corresponding two positions are 1 but all the other positions are 0. Let $W = \mathrm{span}(\boldsymbol{w}_1, \cdots, \boldsymbol{w}_7)$ be the linear span whose basis is $\boldsymbol{w}_1, \cdots, \boldsymbol{w}_7$. Therefore, the bias of $\boldsymbol{u}$ denoted by $\epsilon_{\boldsymbol{u}}$ is estimated as $\epsilon_{\boldsymbol{u}} = \sum_{\boldsymbol{w} \in W} \sum_{\boldsymbol{v} \in V} 2 \times \epsilon_{h,\mathbb{T}_z}(\boldsymbol{a}_{\mathbb{T}_z} \oplus \boldsymbol{v} \oplus \boldsymbol{w}) \times \epsilon_{g^*,\mathbb{B}}(\boldsymbol{a}_{\mathbb{T}_z} \oplus \boldsymbol{v})$.

Note that $\boldsymbol{a}_{40}[0,\cdots,6]$ is not involved in W, and the absolute value of bias $\epsilon_{\boldsymbol{u}}$ is invariable as far as we use $\boldsymbol{a}_{40}[0,\cdots,6]$ satisfying $\epsilon_{h,0} = \pm 2^{-5}$. Therefore, we do not search $\boldsymbol{a}_{40}[0,\cdots,6]$ any more. Finally, we searched exhaustively 2^{35} $\boldsymbol{a}_{\mathbb{T}_z}[0,\cdots,6]$ with $\boldsymbol{a}_{40}[0,\cdots,6] = \boldsymbol{0}$ and there are 64 $\boldsymbol{a}_{40}[0,\cdots,6]$ such that $\epsilon_{h,0} = \pm 2^{-5}$. As a result, we found $r = 7077888 \times 64 \approx 2^{28.76}$ $\boldsymbol{u}$ whose absolute value of bias is greater than $\epsilon = 2^{-38.08}$.

4.2 The Degraded System

In the following, we will show that any internal state variable of the NFSR can be computed from the value of the NFSR state variables at fixed time instant t_0 and of some keystream bits, under the condition that the LFSR initial state $L^{(0)}$ is known.

Forwards: We first consider how to express n_{t_0+40}. According to Eq.(13), z_{t_0+6} is the first keystream bit which certainly depends on n_{t_0+40}, thus we have

$$
\begin{aligned}
n_{t_0+40} = z_{t_0+6} &\oplus (\underline{n_{t_0+7}} \oplus \underline{n_{t_0+12}} \oplus \underline{n_{t_0+21}} \oplus \underline{n_{t_0+23}} \oplus \underline{n_{t_0+29}} \oplus \underline{n_{t_0+34}}) \\
&\oplus (l_{t_0+12}\underline{n_{t_0+10}} \oplus l_{t_0+38}\underline{n_{t_0+10}n_{t_0+44}}) \\
&\oplus (l_{t_0+36} \oplus l_{t_0+14}l_{t_0+16} \oplus l_{t_0+38}l_{t_0+23} \oplus l_{t_0+25}l_{t_0+29}),
\end{aligned}
$$

where $l_{t_0+38}\underline{n_{t_0+10}n_{t_0+44}}$ is a quadratic term which contains the later internal state variable n_{t_0+44}. To derive successfully the linear relations by induction, the outcome of term $l_{t_0+38}\underline{n_{t_0+10}n_{t_0+44}}$ is guessed. The most probable guess would be that this term produces zero, since $\Pr[x\&y\&z = 0] = \frac{7}{8}$. Next we assume that for all i, $t_0 + 40 \le i < t_0 + 40 + j$, all the bits n_i have be expressed as a linear combination of the NFSR state variables at time instant t_0 and of some keystream bits when the outcome of term $l_{(i-34)+32}n_{(i-34)+4}n_{(i-34)+38}$ is guessed $t_0 \le i < t_0 + 40 + j$. Note that z_{t_0+6+j} is the first keystream bit which is certainly dependent on n_{t_0+40+j}, which indicates that

$$
\begin{aligned}
n_{t_0+40+j} = z_{t_0+6+j} &\oplus (\underline{n_{t_0+7+j}} \oplus \underline{n_{t_0+12+j}} \oplus \underline{n_{t_0+21+j}} \oplus \underline{n_{t_0+23+j}} \oplus \underline{n_{t_0+29+j}} \oplus \underline{n_{t_0+34+j}}) \\
&\oplus (l_{t_0+12+j}\underline{n_{t_0+10+j}} \oplus l_{t_0+38+j}\underline{n_{t_0+10+j}n_{t_0+44+j}}) \\
&\oplus (l_{t_0+36+j} \oplus l_{t_0+14+j}l_{t_0+16+j} \oplus l_{t_0+38+j}l_{t_0+23+j} \oplus l_{t_0+25+j}l_{t_0+29+j})
\end{aligned}
$$

and the variable n_{t_0+40+j} is expressed as a function of the internal NFSR variables n_i with $i < t_0+40+j$, n_{t_0+44+j} and of a keystream bit z_{t_0+6+j}. When we guess the outcome of term $l_{t_0+38+j}n_{t_0+10+j}n_{t_0+44+j}$, by induction assumption n_{t_0+40+j} can be expressed as a linear combination of the NFSR state variables at time instant t_0 and of keystream bits $\{z_{t_0+6+j}|j \geq 0\}$.

Whenever the outcomes of terms $\{l_{t_0+38+j}n_{t_0+10+j}n_{t_0+44+j}|j = 0, \cdots, \varTheta - 1\}$ are guessed, we can get the value of $\varTheta$ variables $\{n_{t_0+40+j}|\, j = 0, \cdots, \varTheta - 1\}$ from the value of the NFSR state variables at time instant t_0 and of keystream bits $\{z_{t_0+6+j}|\, j = 0, \cdots, \varTheta - 1\}$, where $\varTheta = d + m' + \theta = 80 + 40 + \log(2^{m+1}p_w + 1)$ and p_w is the probability that a wrong value is chosen as the LFSR initial state during Algorithm 1. The probability that all the guesses for items are right is $\rho = \left(\frac{7}{8}\right)^{\varTheta}$ and the expect length of the keystream is around $\frac{1}{\rho} = \left(\frac{7}{8}\right)^{-\varTheta}$.

Backwards: To get the value of $\varTheta$ variables $\{n_{t_0-j}|\, j = 1, \cdots, \varTheta\}$ from the value of the NFSR state variables at time instant t_0 and of keystream bits $\{z_{t_0-2-j}|\, j = 0, \cdots, \varTheta - 1\}$, we need perform recursively the following equation $\varTheta$ times

$$\begin{aligned} n_{t_0-1} = z_{t_0-2} &\oplus (\underline{n_{t_0+4}} \oplus \underline{n_{t_0+13}} \oplus \underline{n_{t_0+15}} \oplus \underline{n_{t_0+21}} \oplus \underline{n_{t_0+26}} \oplus \underline{n_{t_0+32}}) \\ &\oplus (l_{t_0+4}\underline{n_{t_0+2}} \oplus l_{t_0+30}\underline{n_{t_0+2}n_{t_0+36}}) \\ &\oplus (l_{t_0+28} \oplus l_{t_0+6}l_{t_0+8} \oplus l_{t_0+30}l_{t_0+15} \oplus l_{t_0+17}l_{t_0+21}). \end{aligned}$$

Similarly with Algorithm 2, we could carry out a state checking process to recover the NFSR sate and the round key bits with time complexity $2^{40} \times (80 + 40) \approx 2^{46.91}$.

4.3 Analysis of Complexities for Attacking Plantlet

As stated previously in Section 4.1, we have found $r = 452984832 \approx 2^{28.76}$ linear masks whose absolute value of bias is greater than $\frac{\epsilon^c}{2} = 2^{-38.08}$ in preparation. According to Theorem 3, we need $\varOmega = \frac{2^{\frac{\beta+7}{2}}\sqrt{61 \ln 2}}{\sqrt{r}(\epsilon^c)^2}$ parity-check equations to identify the correct LFSR initial state. Therefore, the data complexity is $D = 6 \times \frac{2^{\frac{\beta+7}{2}}\sqrt{61 \ln 2}}{\sqrt{r}(\epsilon^c)^2}$ keystream bits.

Attack Scenario: $T_1 = \frac{2^{\frac{\beta+7}{2}}\sqrt{61 \ln 2}}{\sqrt{r}(\epsilon^c)^2} + r2^{62-\beta}p_1$ and $T_2 = \left(\frac{8}{7}\right)^{120} \times 2^{40} \times (80 + 40) \approx 2^{70.02}$, where $p_1 = Q((61 \ln 2 \times r^{-1}2^{\beta+3})^{\frac{1}{4}})$, $r = 2^{28.76}$ and $\epsilon^c = 2^{-37.08}$. Moreover, $T_1 = 2^{65.98} \times 2^{\frac{\beta}{2}} + 2^{90.76-\beta}p_1$, where $p_1 = Q((61 \ln 2 \times 2^{-28.76}2^{\beta+3})^{\frac{1}{4}})$ and $p_2 = 1 - p_1$. To make a balance between two dominant terms of the time complexity, we choose $\beta = 16$, and complexities become $T_1 = 2^{74.61}$, $D = 2^{76.57}$. Furthermore, $p_1 \approx 0.3191$, $p_2 \approx 0.6809$ and $\mu_1 \approx r2^{-\beta}p_1 \approx 2205$, $\mu_2 = r2^{-\beta}p_2 \approx 4707$. Therefore, we choose $th_p = 2637$ and the probabilities are $p_c \approx 1$, $p_w = 2^{-62.09}$. Note that when $\beta = 16$, we have that $2^{90.76-\beta}p_1 >> 2^{57.52-\beta}p_2$ and $2^{90.76-\beta}p_1 >> (61 - \beta) \times 2^{61-\beta}$ are satisfied. In conclusion, the total time complexity of our divide-and-conquer fast correlation attack is $T = T_1 + T_2 \approx 2^{74.61}$.

According to Theorem 3 in the precise mode, we can get the following time and data complexities for attacking Plantlet with different choices of β, listed in Table 3.

Table 3. Time, memory and data complexities of attacking Plantlet.

Time	Memory	Data	β
$2^{79.74}$	2^{51}	$2^{67.04}$	10
$2^{78.73}$	2^{50}	$2^{68.04}$	11
$2^{77.72}$	2^{49}	$2^{69.04}$	12
$2^{76.70}$	2^{48}	$2^{70.04}$	13
$2^{75.69}$	2^{47}	$2^{71.04}$	14
$2^{74.68}$	2^{46}	$2^{72.04}$	15
$2^{73.75}$	2^{45}	$2^{73.04}$	16

As the size of bypassed bits β increases, the time complexity decreases temporarily, but more data complexity is required for a successful attack. When $\beta = 16$, the minimum time complexity is achieved, i.e., $T = 2^{73.75}$ and the required data complexity is $D = 2^{73.04}$. When we bypass no bit of the LFSR initial state, an attack could be launched with the minimum data complexity $D = 2^{67.04}$ and the time complexity of $2^{79.74}$.

Remark 4. For Plantlet, the round key bits are the whole secret key k_i, $0 \le i \le 79$.

5 Applications: Fruit-v2 Case

In this section, we apply our divide-and-conquer fast correlation attacks to Fruit-v2. First, we show how to construct the desirable parity-check equations for Fruit-v2. Then under the condition that the LFSR initial state is known, a cryptanalysis is given for the degraded system of Fruit-v2. In the following, the complexities of recovering the secret key are presented.

Before showing details of our attacks, we provide a brief description of Fruit-v2 in the initialization and keystream generation phases. Fruit-v2 is a bit-oriented stream cipher and utilizes an 80-bit secret key $K = (k_0, \cdots, k_{79})$ and a 70-bit public initial value $IV = (iv_0, \cdots, iv_{69})$ to generate the keystream. It is composed of a 43-bit LFSR whose state at time instant t is denoted by $L^{(t)} = (l_t, \cdots, l_{t+42})$, a linked 37-bit NFSR whose state at time instant is denoted by $N^{(t)} = (n_t, \cdots, n_{t+36})$, an 80-bit fixed key register and two counter registers, a 7-bit $C_r = (c_t^0, \cdots, c_t^6)$ and an 8-bit $C_c = (c_t^7, \cdots, c_t^{14})$ allocated for the round key function and the initialization/keystream generation, respectively. These two counters increase one by one independently at each clock, and work continually, i.e., after they becomes all ones, counting from zeros to all ones again. Note that c_t^6 and c_t^{14} are LSBs of the two counters respectively.

The LFSR is updated independently and recursively by a linear function as $l_{t+43} = f(L^{(t)}) = l_t \oplus l_{t+8} \oplus l_{t+18} \oplus l_{t+23} \oplus l_{t+28} \oplus l_{t+37}$. The NFSR is updated as defined in the following:

$$\begin{aligned} n_{t+37} &= k'_t \oplus c_t^3 \oplus l_t \oplus g(N^{(t)}) \\ &= k'_t \oplus c_t^3 \oplus l_t \oplus n_t \oplus n_{t+10} \oplus n_{t+20} \oplus n_{t+12}n_{t+3} \\ &\oplus n_{t+14}n_{t+25} \oplus n_{t+5}n_{t+23}n_{t+31} \\ &\oplus n_{t+8}n_{t+18} \oplus n_{t+28}n_{t+30}n_{t+32}n_{t+34}, \end{aligned} \tag{14}$$

where k'_t is the round key bit and c_t^3 is the counter bit from C_r at time instant t. The round key bit is generated by combining 6 bits of the key as $k'_t = RKF(K, t) = k_s k_{y+32} \oplus k_p k_{u+64} \oplus k_{q+16} \oplus k_{r+48}$. Here, the values of s, y, u, p, q and r are given as $s = (c_t^0 c_t^1 c_t^2 c_t^3 c_t^4)$, $y = (c_t^5 c_t^6 c_t^0 c_t^1 c_t^2)$, $u = (c_t^3 c_t^4 c_t^5 c_t^6)$, $p = (c_t^0 c_t^1 c_t^2 c_t^3)$, $q = (c_t^4 c_t^5 c_t^6 c_t^0 c_t^1)$ and $r = (c_t^2 c_t^3 c_t^4 c_t^5 c_t^6)$.

The filtering function is defined as

$$\begin{aligned} h\left(L^{(t)}_{\mathbb{T}_{h,L}}, N^{(t)}_{\mathbb{T}_{h,N}}\right) &= l_{t+6}l_{t+15} \oplus l_{t+1}l_{t+22} \oplus n_{t+35}l_{t+27} \\ &\oplus l_{t+11}l_{t+33} \oplus n_{t+1}n_{t+33}l_{t+42}, \end{aligned}$$

where the two subsets are

$$L^{(t)}_{\mathbb{T}_{h,L}} = (l_{t+1}, l_{t+6}, l_{t+11}, l_{t+15}, l_{t+22}, l_{t+27}, l_{t+33}, l_{t+42})$$

and $N^{(t)}_{\mathbb{T}_{h,N}} = (n_{t+1}, n_{t+33}, n_{t+35})$. The entire output function is determined by

$$z_t = h\left(L^{(t)}_{\mathbb{T}_{h,L}}, N^{(t)}_{\mathbb{T}_{h,N}}\right) \oplus l_{t+38} \oplus \bigoplus_{b \in \mathbb{B}} n_{t+\mathbb{B}}, \tag{15}$$

where $\mathbb{B} = \{0, 7, 13, 19, 24, 29, 36\}$.

During the initialization phase, the secret key bits are loaded to the NFSR and LFSR from LSB to MSB, i.e., $n_i = k_i$, $0 \le i \le 36$; $l_i = k_{37+i}$, $0 \le i \le 42$. Then the IV bits are padded to 130-bit $IV' = (iv'_0, \cdots, iv'_{129})$ by concatenating 1 bit one and 9 bits zeros to the head of IV, and 50 bits zeros to the end of IV. In the first step of the initialization, C_r, C_c are set to $\mathbf{0}$ and the cipher is clocked 130 rounds as follows: the LFSR is updated as $l_{t+43} = z_t \oplus iv'_t \oplus f(L^{(t)})$, while the NFSR is updated as $n_{t+37} = z_t \oplus iv'_t \oplus k'_t \oplus c_t^3 \oplus l_t \oplus g(N^{(t)})$, and no keystream is generated. Then, in the second step of the initialization, all bits of C_r are set equal to the LSBs of the NFSR except the last bit that is equal to the

LSB of the LFSR, and also l_{130} is set to 1. Hereafter the cipher is clock 80 rounds without the feedback in the LFSR and NFSR, i.e., the update function of the LFSR is changed to $l_{t+43} = f(N^{(t)})$, while the update function of the NFSR is changed to $n_{t+37} = k'_t \oplus c^3_t \oplus l_t \oplus g(N^{(t)})$, and no keystream is generated. After the initialization phase of 210 rounds clocks, the cipher enters the keystream generation phase and the keystream bits are produced.

Note that C_r is only known in the first step (130 rounds) of the initialization phase. Since C_r is fed from the LFSR and NFSR after the first 130 rounds, it is not known any more. However, the counter bit c^3_t is periodic with a cycle of length $2^4 = 16$, i.e., $c^3_{t+16} = c^3_t$, $\forall t \geq 0$, and the cycle is that 8 zeros followed by 8 ones.

It is obviously that the Assumed Property 1 holds for Fruit-v2 and the $RKF(\cdot)$ is periodic with a cycle of minimum length $d = 128$, i.e., $k'_{t+128} = k'_t$. Regarding the Assumed Property 2, we give a more accurate analysis in the following discuss.

5.1 Deriving Linear Approximate Equations

In this subsection, we expect to estimate the number and bias of different linear approximate equations which are used to construct the desirable parity-check equations. First, we derive the linear approximate representations for the sum of some keystream bits. Then, we could evaluate the number and bias of different linear approximate equations by exhaustively searching all the possible representations.

Linear Approximate Representations Considering the best linear approximation of the NFSR update function Eq.(14) with bias $2^{-4.6}$ as follows,

$$n_{t+37} \approx k'_t \oplus c^3_t \oplus l_t \oplus n_t \oplus n_{t+10} \oplus n_{t+20},$$

we choose the set of taps as $\mathbb{T}_z = \{0, 10, 20, 37\}$. Then, we have

$$\bigoplus_{i\in\mathbb{T}_z} z_{t+i} = \bigoplus_{i\in\mathbb{T}_z} l_{t+i+38} \oplus \bigoplus_{b\in\mathbb{B}} l_{t+b} \oplus \bigoplus_{i\in\mathbb{T}_z} h(L^{(t+i)}_{\mathbb{T}_{h,L}}, N^{(t+i)}_{\mathbb{T}_{h,N}}) \oplus \bigoplus_{b\in\mathbb{B}} g^*(N^{(t+b)}) \oplus \bigoplus_{b\in\mathbb{B}} k'_{t+b} \oplus \bigoplus_{b\in\mathbb{B}} c^3_{t+b},$$

where $g^*(N^{(t)}) = n_t \oplus n_{t+10} \oplus n_{t+20} \oplus g(N^{(t)})$ and it has the same bias $2^{-4.6}$, i.e., $\Pr[g^*(N^{(t)}) = 0] = \frac{1}{2} + 2^{-4.6}$.

Let $\boldsymbol{a}_i \in \{0,1\}^{11}$ be the input linear mask for h function at time instant $t+i$, $\boldsymbol{a}_i = (a_i[0], \cdots, a_i[10])$. Then

$$h(L^{(t+i)}_{\mathbb{T}_{h,L}}, N^{(t+i)}_{\mathbb{T}_{h,N}}) \approx \boldsymbol{a}_i[0-7] \cdot \left(L^{(t+i)}_{\mathbb{T}_{h,L}}\right)^T \oplus \boldsymbol{a}_i[8-10] \cdot \left(N^{(t+i)}_{\mathbb{T}_{h,N}}\right)^T$$

with bias $\epsilon_{h,i}(\boldsymbol{a}_i) = \pm\frac{3}{128}$ or $\pm\frac{1}{128}$. Due to $|\mathbb{T}| = 4$, there are 4 active h functions which need to be approximated. Let $\boldsymbol{a}_{\mathbb{T}_z} \in \{0,1\}^{11\times 4}$ be the concatenated linear mask, i.e., $\boldsymbol{a}_{\mathbb{T}_z} = (\boldsymbol{a}_0, \boldsymbol{a}_{10}, \boldsymbol{a}_{20}, \boldsymbol{a}_{37})$. The total bias of all the approximated h functions is computed as $\epsilon_{h,\mathbb{T}_z}(\boldsymbol{a}_{\mathbb{T}_z}) = 2^{4-1} \times \prod_{i\in\mathbb{T}_z} \epsilon_{h,i}(\boldsymbol{a}_i)$ because of the piling-up lemma.

Let

$$\epsilon_{g^*,\mathbb{B}}(\boldsymbol{a}_{\mathbb{T}_z}) = \Pr\left[\bigoplus_{i\in\mathbb{T}_z} \boldsymbol{a}_i[8-10] \cdot \left(N^{(t+i)}_{\mathbb{T}_{h,N}}\right)^T \oplus \bigoplus_{b\in\mathbb{B}} g^*(N^{(t+b)}) = 0\right] - \frac{1}{2}$$

and the bias is independent on $\boldsymbol{a}_i[0-7]$ for all $i \in \mathbb{T}_z$. If one of $a_0[8]$, $a_{10}[8]$ and $a_{37}[10]$ is 1, the bias is always 0 because n_{t+1}, n_{t+11} and n_{t+72} are not involved in $\bigoplus_{b\in\mathbb{B}} g^*(N^{(t+b)})$. We summarize $\epsilon_{g^*,\mathbb{B}}(\boldsymbol{a}_{\mathbb{T}_z})$ when $a_0[8]$, $a_{10}[8]$ and $a_{37}[10]$ are 0 in Table 4.

For any fixed $\boldsymbol{a}_{\mathbb{T}_z}$, we can derive the following linear approximate representation

$$\bigoplus_{i\in\mathbb{T}_z} z_{t+i} \approx \bigoplus_{i\in\mathbb{T}_z} l_{t_i+38} \oplus \bigoplus_{b\in\mathbb{B}} l_{t+b} \oplus \bigoplus_{i\in\mathbb{T}_z} \boldsymbol{a}_i[0-7] \cdot \left(L^{(t+i)}_{\mathbb{T}_{h,L}}\right)^T \oplus \bigoplus_{b\in\mathbb{B}} k'_{t+b} \oplus \bigoplus_{b\in\mathbb{B}} c^3_{t+b}$$

and its bias is evaluated as $2 \times \epsilon_{h,\mathbb{T}_z}(\boldsymbol{a}_{\mathbb{T}_z}) \times \epsilon_{g^*,\mathbb{B}}(\boldsymbol{a}_{\mathbb{T}_z})$.

Table 4. Summary of bias when $a_i[8-10]$, $i \in \mathbb{T}_z$ are fixed. Let $*$ be the arbitrary bit.

$a_0[9]$	$a_0[10]$	$a_{10}[9]$	$a_{10}[10]$	$a_{20}[8]$	$a_{20}[9]$	$a_{20}[10]$	$a_{37}[8]$	$a_{37}[9]$	$\epsilon_{g^*,\mathbb{B}}$
0	0	0	0	0	0	0	0	0	$+2^{-15.08}$
1	0	0	0	0	0	0	0	0	$+2^{-15.08}$
0	0	1	0	0	0	0	0	0	$+2^{-15.35}$
1	0	1	0	0	0	0	0	0	$-2^{-15.35}$
0	0	0	0	1	0	0	0	0	$+2^{-19.87}$
1	0	0	0	1	0	0	0	0	$+2^{-19.87}$
0	0	1	0	1	0	0	0	0	$+2^{-20.19}$
1	0	1	0	1	0	0	0	0	$-2^{-20.19}$
0	0	0	0	0	0	0	1	0	$+2^{-15.08}$
1	0	0	0	0	0	0	1	0	$+2^{-15.08}$
0	0	1	0	0	0	0	1	0	$+2^{-15.35}$
1	0	1	0	0	0	0	1	0	$-2^{-15.35}$
0	0	0	0	1	0	0	1	0	$+2^{-19.87}$
1	0	0	0	1	0	0	1	0	$+2^{-19.87}$
0	0	1	0	1	0	0	1	0	$+2^{-20.19}$
1	0	1	0	1	0	0	1	0	$-2^{-20.19}$
0	0	0	0	0	0	0	0	1	$+2^{-17.89}$
1	0	0	0	0	0	0	0	1	$+2^{-17.89}$
0	0	1	0	0	0	0	0	1	$+2^{-18.15}$
1	0	1	0	0	0	0	0	1	$-2^{-18.15}$
0	0	0	0	1	0	0	0	1	$+2^{-22.68}$
1	0	0	0	1	0	0	0	1	$+2^{-22.68}$
0	0	1	0	1	0	0	0	1	$+2^{-23.00}$
1	0	1	0	1	0	0	0	1	$-2^{-23.00}$
0	0	0	0	0	0	0	1	1	$+2^{-17.89}$
1	0	0	0	0	0	0	1	1	$+2^{-17.89}$
0	0	1	0	0	0	0	1	1	$+2^{-18.15}$
1	0	1	0	0	0	0	1	1	$-2^{-18.15}$
0	0	0	0	1	0	0	1	1	$+2^{-22.68}$
1	0	0	0	1	0	0	1	1	$+2^{-22.68}$
0	0	1	0	1	0	0	1	1	$+2^{-23.00}$
1	0	1	0	1	0	0	1	1	$-2^{-23.00}$
$*$	1	$*$	$*$	$*$	$*$	$*$	$*$	$*$	0
$*$	$*$	$*$	1	$*$	$*$	$*$	$*$	$*$	0
$*$	$*$	$*$	$*$	$*$	1	$*$	$*$	$*$	0
$*$	$*$	$*$	$*$	$*$	$*$	1	$*$	$*$	0

Linear Approximate Equations From the above linear approximate representations, we can derive the linear approximate equation with some fixed linear mask $\boldsymbol{u}$

$$\bigoplus_{i\in\mathbb{T}_z} z_{t+i} \approx L^{(0)} \cdot \left(F^t \times \boldsymbol{u}\right) \oplus \bigoplus_{b\in\mathbb{B}} k'_{t+b} \oplus \bigoplus_{b\in\mathbb{B}} c^3_{t+b},$$

where $\boldsymbol{u} \in \{0,1\}^{43}$ is a column vector. If different $\boldsymbol{a}_{\mathbb{T}_z}$'s derive the same linear mask $\boldsymbol{u}$, corresponding biases should be added up to get the bias of $\boldsymbol{u}$, i.e., $\epsilon_{\boldsymbol{u}} = \sum_{\{\boldsymbol{a}_{\mathbb{T}_z} | U(\boldsymbol{a}_{\mathbb{T}_z})=\boldsymbol{u}\}} 2 \times \epsilon_{h,\mathbb{T}_z}(\boldsymbol{a}_{\mathbb{T}_z}) \times \epsilon_{g^*,\mathbb{B}}(\boldsymbol{a}_{\mathbb{T}_z})$, where

$$\begin{aligned} U(\boldsymbol{a}_{\mathbb{T}_z}) = \bigoplus_{i\in\mathbb{T}_z} ((a_i[0] \cdot \boldsymbol{g}_{i+1} \oplus a_i[1] \cdot \boldsymbol{g}_{i+6} \oplus a_i[2] \cdot \boldsymbol{g}_{i+11} \oplus a_i[3] \cdot \boldsymbol{g}_{i+15} \oplus a_i[4] \cdot \boldsymbol{g}_{i+22} \oplus a_i[5] \cdot \boldsymbol{g}_{i+27} \oplus a_i[6] \\ \cdot \boldsymbol{g}_{i+33} \oplus a_i[7] \cdot \boldsymbol{g}_{i+42}) \oplus \boldsymbol{g}_{i+38}) \oplus \bigoplus_{b\in\mathbb{B}} \boldsymbol{g}_b. \end{aligned}$$

Since the function $U(\boldsymbol{a}_{\mathbb{T}_z})$ is independent on $\boldsymbol{a}_i[8,9,10]$, we need to sum up all biases with a non-zero $\epsilon_{g^*,\mathbb{B}}$ summarized in Table 4, where $\boldsymbol{a}_i[0,\cdots,7]$ is identical and only $\boldsymbol{a}_i[8,9,10]$ varies for $i \in \mathbb{T}_z$. Let V be subset of $\{0,1\}^{11\times 4}$ whose elements are 32 corresponding vectors $\boldsymbol{a}_{\mathbb{T}_z}$ with non-zero $\epsilon_{g^*,\mathbb{B}}$ in Table 4. Moreover, there are some special relationships similar to the case of Plantlet, and we have three relationships as showed in the following.

- $a_0[2]$ and $a_{10}[0]$ (since $\boldsymbol{g}_{0+11} = \boldsymbol{g}_{10+1} = \boldsymbol{g}_{11}$).
- $a_0[7]$ and $a_{20}[4]$ (since $\boldsymbol{g}_{0+42} = \boldsymbol{g}_{20+22} = \boldsymbol{g}_{42}$).
- $a_{10}[2]$ and $a_{20}[0]$ (since $\boldsymbol{g}_{10+11} = \boldsymbol{g}_{20+1} = \boldsymbol{g}_{21}$).

Let $\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3$ be the (11×4)-bit vectors generated by the above relationships with the corresponding two positions are 1 but all the other positions are 0. Let $W = \mathrm{span}(\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3)$ be the linear span whose basis is $\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3$. Therefore, the bias of $\boldsymbol{u}$ is estimated as $\epsilon_{\boldsymbol{u}} = \sum_{\boldsymbol{w}\in W} \sum_{\boldsymbol{v}\in V} 2 \times \epsilon_{h,\mathbb{T}_z}(\boldsymbol{a}_{\mathbb{T}_z} \oplus \boldsymbol{v} \oplus \boldsymbol{w}) \times \epsilon_{g^*,\mathbb{B}}(\boldsymbol{a}_{\mathbb{T}_z} \oplus \boldsymbol{v})$.

As a result, we searched exhaustively 2^{32} $\boldsymbol{a}_{\mathbb{T}_z}[0,\cdots,7]$ and found $r = 16777216 = 2^{24}$ $\boldsymbol{u}$ whose absolute value of bias is greater than $\epsilon = 2^{-29.52}$.

Remark 5. Since the counter bit is unknown in Fruit-v2, there is slight difference in constructed parity-check equations. We redefine d as the least common multiple of the two cycle lengths of k'_t and c^3_t. The round key bits k'_t and the counter bits c^3_t have cycles of length 128 and 16, respectively. Therefore, we have $d = 128$ and $k'_{t_0+dt'} \oplus c^3_{t_0+dt'} = k'_{t_0} \oplus c^3_{t_0}$. The parity-check equations are constructed as $\left(L^{(0)} \times F_{\boldsymbol{u}_j}\right) \cdot \boldsymbol{g}_{dt'} \oplus \hat{z}_{dt'} \oplus \hat{c}_0 \oplus \hat{k}_0$, where $\hat{c}_0 = \bigoplus_{b\in\mathbb{B}} c^3_b$ and $\hat{k}_0 = \bigoplus_{b\in\mathbb{B}} k'_b$. Then we treat the whole unknown bit $\hat{c}_0 \oplus \hat{k}_0$ for Fruit-v2 same to the $\hat{k}_0$ for the generic model in the remaining discuss.

5.2 The Degraded System

In the following, we will show that any internal state variable of the NFSR can be computed from the value of the NFSR state variables at fixed time instant t_0 and of some keystream bits, under the condition that the LFSR initial state $L^{(0)}$ is known.

Forwards: We first consider how to express n_{t_0+37}. According to Eq.(15), z_{t_0+1} is the first keystream bit which depends on n_{t_0+37}, thus we have

$$\begin{aligned} n_{t_0+37} = z_{t_0+1} &\oplus (\underline{n_{t_0+1}} \oplus \underline{n_{t_0+8}} \oplus \underline{n_{t_0+14}} \oplus \underline{n_{t_0+20}} \oplus \underline{n_{t_0+25}} \oplus \underline{n_{t_0+30}}) \\ &\oplus (l_{t_0+28}\underline{n_{t_0+36}} \oplus l_{t_0+43}\underline{n_{t_0+2}n_{t_0+34}}) \\ &\oplus (l_{t_0+39} \oplus l_{t_0+7}l_{t_0+16} \oplus l_{t_0+2}l_{t_0+23} \oplus l_{t_0+12}l_{t_0+34}), \end{aligned}$$

i.e., n_{t_0+37} has been expressed as a quadratic function of $N^{(t_0)} = (n_{t_0},\cdots,n_{t_0+36})$ and of a keystream bit z_{t_0+1}. Next we assume that for all $i :\ t_0 + 37 \leq i < t_0 + 37 + j$, all the bits n_i have be expressed as a nonlinear function of the NFSR state variables at time instant t_0 and of some keystream bits. Note that z_{t_0+1+j} is the first keystream bit dependent on n_{t_0+37+j}, which indicates that

$$\begin{aligned} n_{t_0+37+j} = z_{t_0+1+j} &\oplus (\underline{n_{t_0+1+j}} \oplus \underline{n_{t_0+8+j}} \oplus \underline{n_{t_0+14+j}} \oplus \underline{n_{t_0+20+j}} \oplus \underline{n_{t_0+25+j}} \oplus \underline{n_{t_0+30+j}}) \\ &\oplus (l_{t_0+28+j}\underline{n_{t_0+36+j}} \oplus l_{t_0+43+j}\underline{n_{t_0+2+j}n_{t_0+34+j}}) \\ &\oplus (l_{t_0+39+j} \oplus l_{t_0+7+j}l_{t_0+16+j} \oplus l_{t_0+2+j}l_{t_0+23+j} \oplus l_{t_0+12+j}l_{t_0+34+j}) \end{aligned}$$

and the variable n_{t_0+37+j} is expressed as a function of the internal NFSR variables n_i with $i < t_0+37+j$ and of a keystream bit z_{t_0+1+j}. By induction assumption n_{t_0+37+j} can be expressed as a function of the NFSR state variables at time instant t_0 and of keystream bits $\{z_{t_0+1+j} | j \geq 0\}$, under the condition that the LFSR initial state $L^{(0)}$ is known.

Remark 6. For Fruit-v2, the subroutine of state checking will exploit the periodic property of the secret information bits during Algorithm 2, i.e., $k'_i \oplus c^3_i = k'_{128+i} \oplus c^3_i, \forall i = 0,\cdots,37+\theta-1$. Therefore, the output of Algorithm 2 will be the full initial state of the target stream cipher $N^{(0)}$, $L^{(0)}$ and the secret information bits $k'_i \oplus c^3_i$ with $0 \leq i \leq 127$.

5.3 Analysis of Complexities for Attacking Fruit-v2

As stated previously in Section 5.1, we have found $r = 2^{24}$ linear masks whose absolute value of bias is greater than $\frac{\epsilon^c}{2} = 2^{-29.52}$ in preparation. According to Theorem 3, we need $\Omega = \frac{2^{\frac{\beta+7}{2}}\sqrt{43\ln 2}}{\sqrt{r}(\epsilon^c)^2}$ parity-check equations to identify the correct LFSR initial state. Therefore, the data complexity is $D = 4 \times \frac{2^{\frac{\beta+7}{2}}\sqrt{43\ln 2}}{\sqrt{r}(\epsilon^c)^2}$ keystream bits.

Attack Scenario: $T_1 = \frac{2^{\frac{\beta+7}{2}}\sqrt{43\ln 2}}{\sqrt{r}(\epsilon^c)^2} + r2^{44-\beta}p_1$ and $T_2 = 2^{37} \times (128 + 37) \approx 2^{44.37}$, where $p_1 = Q((43\ln 2 \times r^{-1}2^{\beta+3})^{\frac{1}{4}})$, $r = 2^{24}$ and $\epsilon^c = 2^{-28.52}$. Moreover, $T_1 = 2^{50.99} \times 2^{\frac{\beta}{2}} + 2^{68-\beta}p_1$, where $p_1 = Q((43\ln 2 \times 2^{-24}2^{\beta+3})^{\frac{1}{4}})$ and $p_2 = 1 - p_1$. To make a balance between the two dominant terms of time complexity, we choose $\beta = 11$, and complexities become $T_1 = 2^{57.06}, D = 2^{58.49}$. Furthermore, $p_1 \approx 0.3398$, $p_2 \approx 0.6602$ and $\mu_1 \approx r2^{-\beta}p_1 \approx 2784$, $\mu_2 = r2^{-\beta}p_2 \approx 5409$. Therefore, we choose $th_p = 3185$ and the probabilities are $p_c \approx 1, p_w \approx 2^{-44.17}$. Note that when $\beta = 11$, we have that $2^{68-\beta}p_1 >> 2^{48-\beta}p_2$

and $2^{68-\beta}p_1 >> (43-\beta) \times 2^{43-\beta}$ are satisfied. In conclusion, the total time complexity of our divide-and-conquer fast correlation attack is $T = T_1 + T_2 \approx 2^{57.06}$.

According to Theorem 3 in the precise mode, we can get the following time and data complexities for attacking Fruit-v2 with different choices of β, listed in Table 5.

Table 5. Time, memory and data complexities of attacking Fruit-v2.

Time	Memory	Data	β
$2^{67.00}$	2^{43}	$2^{43.59}$	0
$2^{66.00}$	2^{42}	$2^{44.59}$	1
$2^{64.99}$	2^{41}	$2^{45.59}$	2
$2^{63.99}$	2^{40}	$2^{46.59}$	3
$2^{62.99}$	2^{39}	$2^{47.59}$	4
$2^{61.98}$	2^{38}	$2^{48.59}$	5
$2^{60.98}$	2^{37}	$2^{49.59}$	6
$2^{59.97}$	2^{36}	$2^{50.59}$	7
$2^{58.96}$	2^{35}	$2^{51.59}$	8
$2^{57.95}$	2^{34}	$2^{52.59}$	9
$2^{56.95}$	2^{33}	$2^{53.59}$	10
$2^{56.01}$	2^{32}	$2^{54.59}$	11
$2^{55.33}$	2^{31}	$2^{55.59}$	12

As the size of bypassed bits β increases, the time complexity decreases temporarily, but more data complexity is required for a successful attack. When $\beta = 12$, the minimum time complexity is achieved, i.e., $T = 2^{55.33}$ and the required data complexity is $D = 2^{55.59}$. When we bypass no bit of the LFSR initial state, an attack could be launched with the minimum data complexity $D = 2^{43.59}$ and the time complexity of $2^{67.00}$.

Remark 7. Once we know the initial state $(L^{(0)}, N^{(0)})$ and the secret information bits $k'_i \oplus c_i^3$ with $0 \leq i \leq 127$ through Algorithm 1 and 2, we can run the inverse process of the initialization phase for 210 rounds and derive the secret key.

6 Applications: Fruit-80 Case

In this section, we apply our divide-and-conquer fast correlation attacks to Fruit-80. Like Fruit-v2, we first construct the desirable parity-check equations for Fruit-80. Then under the condition that the LFSR initial state is known, the complexities of recovering the secret key are presented.

Before showing details of our attacks, we provide a brief description of Fruit-80 in the initialization and keystream generation phases. Fruit-80 is a bit-oriented stream cipher and utilizes an 80-bit secret key $K = (k_0, \cdots, k_{79})$ and a 70-bit public initial value $IV = (iv_0, \cdots, iv_{69})$ to generate the keystream. It is composed of a 43-bit LFSR whose state at time instant t is denoted by $L^{(t)} = (l_t, \cdots, l_{t+42})$, a linked 37-bit NFSR whose state at time instant is denoted by $N^{(t)} = (n_t, \cdots, n_{t+36})$, an 80-bit fixed key register and a 7-bit counter registers $C_r = (c_t^0, \cdots, c_t^6)$ allocated for the round key function.

The LFSR is updated independently and recursively by a linear function as $l_{t+43} = f(L^{(t)}) = l_t \oplus l_{t+8} \oplus l_{t+18} \oplus l_{t+23} \oplus l_{t+28} \oplus l_{t+37}$. The NFSR is updated as defined in the following:

$$\begin{aligned} n_{t+37} &= k'_t \oplus l_t \oplus g(N^{(t)}) \\ &= k'_t \oplus l_t \oplus n_t \oplus n_{t+10} \oplus n_{t+20} \oplus n_{t+12}n_{t+3} \\ &\oplus n_{t+14}n_{t+25} \oplus n_{t+5}n_{t+23}n_{t+31} \\ &\oplus n_{t+8}n_{t+18} \oplus n_{t+28}n_{t+30}n_{t+32}n_{t+34}, \end{aligned} \qquad (16)$$

where k'_t is the round key bit at time instant t. The round key bit for g function is generated by combining 3 bits of the key as $k'_t = RKF(K,t) = k_r k_{p+16} k_{q+48} \oplus k_r k_{p+16} \oplus k_{p+16} k_{q+48} \oplus k_r k_{q+48} \oplus k_{p+16}$. Here, the values of r, p and q are given as $r = (c_t^0 c_t^1 c_t^2 c_t^3)$, $p = (c_t^1 c_t^2 c_t^3 c_t^4 c_t^5)$ and $q = (c_t^2 c_t^3 c_t^4 c_t^5 c_t^6)$.

The filtering function is defined as

$$h\left(L^{(t)}_{\mathbb{T}_{h,L}}, N^{(t)}_{\mathbb{T}_{h,N}}, k^*_t\right) = k^*_t(n_{t+36} \oplus l_{t+19}) \oplus l_{t+6}l_{t+15} \oplus l_{t+1}l_{t+22} \oplus n_{t+35}l_{t+27} \oplus n_{t+1}n_{t+24} \oplus n_{t+1}n_{t+33}l_{t+42},$$

where k^*_t is the round key bit at time instant t, the two subsets are

$$L^{(t)}_{\mathbb{T}_{h,L}} = (l_{t+1}, l_{t+6}, l_{t+15}, l_{t+19}, l_{t+22}, l_{t+27}, l_{t+42})$$

and $N^{(t)}_{\mathbb{T}_{h,N}} = (n_{t+1}, n_{t+24}, n_{t+33}, n_{t+35}, n_{t+36})$. The round key bit for h function is generated by combining the same 3 bits of the key as $k^*_t = RKF^*(K,t) = k_r k_{p+16} \oplus k_{p+16}k_{q+48} \oplus k_r k_{q+48} \oplus k_r \oplus k_{p+16} \oplus k_{q+48}$, where the values of r, p and q are defined in the above. The entire output function is determined by

$$z_t = h\left(L^{(t)}_{\mathbb{T}_{h,L}}, N^{(t)}_{\mathbb{T}_{h,N}}, k^*_t\right) \oplus l_{t+38} \oplus \bigoplus_{b\in\mathbb{B}} n_{t+\mathbb{B}}, \tag{17}$$

where $\mathbb{B} = \{0, 7, 19, 29, 36\}$.

In the initialization phase, Fruit-80 works in the same way as Fruit-v2 expect for only clocking 80 times in the first step of the initialization.

It is obviously that the Assumed Property 1 holds for Fruit-80 and the $RKF(\cdot)$ is periodic with a cycle of minimum length $d = 128$, i.e., $k'_{t+128} = k'_t$. Regarding the Assumed Property 2, we give a more accurate analysis in the following discuss.

6.1 Deriving Linear Approximate Equations

In this subsection, we expect to estimate the number and bias of different linear approximate equations which are used to construct the desirable parity-check equations. First, we derive the linear approximate representations for the sum of some keystream bits. Then, we could evaluate the number and bias of different linear approximate equations by exhaustively searching all the possible representations.

Linear Approximate Representations Like Fruit-v2, we consider the best linear approximation of the NFSR update function Eq.(16) with bias $2^{-4.6}$ and choose the set of taps as $\mathbb{T}_z = \{0, 10, 20, 37\}$. Then, we have

$$\bigoplus_{i\in\mathbb{T}_z} z_{t+i} = \bigoplus_{i\in\mathbb{T}_z} l_{t+i+38} \oplus \bigoplus_{b\in\mathbb{B}} l_{t+b} \oplus \bigoplus_{i\in\mathbb{T}_z} h(L^{(t+i)}_{\mathbb{T}_{h,L}}, N^{(t+i)}_{\mathbb{T}_{h,N}}, k^*_t) \oplus \bigoplus_{b\in\mathbb{B}} g^*(N^{(t+b)}) \oplus \bigoplus_{b\in\mathbb{B}} k'_{t+b},$$

where $g^*(N^{(t)}) = n_t \oplus n_{t+10} \oplus n_{t+20} \oplus g(N^{(t)})$ and it has the same bias $2^{-4.6}$, i.e., $\Pr[g^*(N^{(t)}) = 0] = \frac{1}{2} + 2^{-4.6}$. Note that for Fruit-80, $\mathbb{B} = \{0, 7, 19, 29, 36\}$.

Let $\boldsymbol{a}_i \in \{0,1\}^{12}$ be the input linear mask for h function at time instant $t+i$, $\boldsymbol{a}_i = (a_i[0], \cdots, a_i[11])$. Then

$$h(L^{(t+i)}_{\mathbb{T}_{h,L}}, N^{(t+i)}_{\mathbb{T}_{h,N}}) \approx \boldsymbol{a}_i[0-6] \cdot \left(L^{(t+i)}_{\mathbb{T}_{h,L}}\right)^T \oplus \boldsymbol{a}_i[7-11] \cdot \left(N^{(t+i)}_{\mathbb{T}_{h,N}}\right)^T$$

with bias $\epsilon_{h,i}(\boldsymbol{a}_i) = \pm 2^{-6}$, $\pm 2^{-7}$ or 0. Like Fruit-v2, there are 4 active h functions which need to be approximated. Let $\boldsymbol{a}_{\mathbb{T}_z} \in \{0,1\}^{12\times 4}$ be the concatenated linear mask, i.e., $\boldsymbol{a}_{\mathbb{T}_z} = (\boldsymbol{a}_0, \boldsymbol{a}_{10}, \boldsymbol{a}_{20}, \boldsymbol{a}_{37})$. The total bias of all the approximated h functions is computed as $\epsilon_{h,\mathbb{T}_z}(\boldsymbol{a}_{\mathbb{T}_z}) = 2^{4-1} \times \prod_{i\in\mathbb{T}_z} \epsilon_{h,i}(\boldsymbol{a}_i)$ because of the piling-up lemma.

Let

$$\epsilon_{g^*,\mathbb{B}}(\boldsymbol{a}_{\mathbb{T}_z}) = \Pr\left[\bigoplus_{i\in\mathbb{T}_z} \boldsymbol{a}_i[7-11] \cdot \left(N^{(t+i)}_{\mathbb{T}_{h,N}}\right)^T \oplus \bigoplus_{b\in\mathbb{B}} g^*(N^{(t+b)}) = 0\right] - \frac{1}{2}$$

and the bias is independent on $\boldsymbol{a}_i[0-6]$ for all $i \in \mathbb{T}_z$. If one of $a_0[7]$, $a_0[11]$, $a_{10}[7]$, $a_{10}[10]$, $a_{10}[11]$, $a_{20}[10]$, $a_{20}[11]$, $a_{37}[10]$ and $a_{37}[11]$ is 1, the bias is always 0 because n_{t+1}, n_{t+36}, n_{t+11}, n_{t+45}, n_{t+46},

Table 6. Summary of bias when $a_i[7-11]$, $i \in \mathbb{T}_z$ are fixed. Let $*$ be the arbitrary bit.

$a_0[8]$	$a_0[9]$	$a_0[10]$	$a_{10}[8]$	$a_{10}[9]$	$a_{20}[7]$	$a_{20}[8]$	$a_{20}[9]$	$a_{37}[7]$	$a_{37}[8]$	$a_{37}[9]$	$\epsilon_{g^*,\mathbb{B}}$
0	0	0	0	0	0	0	0	0	0	0	$+2^{-13.28}$
1	0	0	0	0	0	0	0	0	0	0	$+2^{-17.80}$
0	1	0	0	0	0	0	0	0	0	0	$+2^{-13.28}$
1	1	0	0	0	0	0	0	0	0	0	$+2^{-17.80}$
0	0	0	1	0	0	0	0	0	0	0	$+2^{-14.86}$
1	0	0	1	0	0	0	0	0	0	0	$+2^{-19.39}$
0	1	0	1	0	0	0	0	0	0	0	$+2^{-14.86}$
1	1	0	1	0	0	0	0	0	0	0	$+2^{-19.39}$
0	0	0	0	1	0	0	0	0	0	0	$+2^{-13.28}$
1	0	0	0	1	0	0	0	0	0	0	$+2^{-17.80}$
0	1	0	0	1	0	0	0	0	0	0	$-2^{-13.28}$
1	1	0	0	1	0	0	0	0	0	0	$-2^{-17.80}$
0	0	0	1	1	0	0	0	0	0	0	$+2^{-14.86}$
1	0	0	1	1	0	0	0	0	0	0	$+2^{-19.39}$
0	1	0	1	1	0	0	0	0	0	0	$-2^{-14.86}$
1	1	0	1	1	0	0	0	0	0	0	$-2^{-19.39}$
0	0	0	0	0	1	0	0	0	0	0	$+2^{-15.26}$
1	0	0	0	0	1	0	0	0	0	0	$+2^{-18.06}$
0	1	0	0	0	1	0	0	0	0	0	$+2^{-15.26}$
1	1	0	0	0	1	0	0	0	0	0	$+2^{-18.06}$
0	0	0	1	0	1	0	0	0	0	0	$+2^{-15.99}$
1	0	0	1	0	1	0	0	0	0	0	$+2^{-18.80}$
0	1	0	1	0	1	0	0	0	0	0	$+2^{-15.99}$
1	1	0	1	0	1	0	0	0	0	0	$+2^{-18.80}$
0	0	0	0	1	1	0	0	0	0	0	$+2^{-15.26}$
1	0	0	0	1	1	0	0	0	0	0	$+2^{-18.06}$
0	1	0	0	1	1	0	0	0	0	0	$-2^{-15.26}$
1	1	0	0	1	1	0	0	0	0	0	$-2^{-18.06}$
0	0	0	1	1	1	0	0	0	0	0	$+2^{-15.99}$
1	0	0	1	1	1	0	0	0	0	0	$+2^{-18.80}$
0	1	0	1	1	1	0	0	0	0	0	$-2^{-15.99}$
1	1	0	1	1	1	0	0	0	0	0	$-2^{-18.80}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$*$	$*$	1	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	0
$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	1	$*$	$*$	0

n_{t+55}, n_{t+56}, n_{t+72} and n_{t+73} are not involved in $\bigoplus_{b\in\mathbb{B}} g^*(N^{(t+b)})$. Therefore, we only need to compute $\epsilon_{g^*,\mathbb{B}}(\boldsymbol{a}_{\mathbb{T}_z})$ when $a_0[7]$, $a_0[11]$, $a_{10}[7]$, $a_{10}[10]$, $a_{10}[11]$, $a_{20}[10]$, $a_{20}[11]$, $a_{37}[10]$ and $a_{37}[11]$ are 0. Note that there are totally 512 non-zero $\epsilon_{g^*,\mathbb{B}}(\boldsymbol{a}_{\mathbb{T}_z})$ and we only list 32 of them in Table 6 due to space limitations.

For any fixed $\boldsymbol{a}_{\mathbb{T}_z}$, we can derive the following linear approximate representation

$$\bigoplus_{i\in\mathbb{T}_z} z_{t+i} \approx \bigoplus_{i\in\mathbb{T}_z} l_{t_i+38} \oplus \bigoplus_{b\in\mathbb{B}} l_{t+b} \oplus \bigoplus_{i\in\mathbb{T}_z} \boldsymbol{a}_i[0-6]\cdot\left(L^{(t+i)}_{\mathbb{T}_{h,L}}\right)^T \oplus \bigoplus_{b\in\mathbb{B}} k'_{t+b}$$

and its bias is evaluated as $2\times\epsilon_{h,\mathbb{T}_z}(\boldsymbol{a}_{\mathbb{T}_z})\times\epsilon_{g^*,\mathbb{B}}(\boldsymbol{a}_{\mathbb{T}_z})$.

Linear Approximate Equations From the above linear approximate representations, we can derive the linear approximate equation with some fixed linear mask $\boldsymbol{u}$

$$\bigoplus_{i\in\mathbb{T}_z} z_{t+i} \approx L^{(0)}\cdot\left(F^t\times\boldsymbol{u}\right)\oplus\bigoplus_{b\in\mathbb{B}} k'_{t+b},$$

where $\boldsymbol{u}\in\{0,1\}^{43}$ is a column vector. If different $\boldsymbol{a}_{\mathbb{T}_z}$'s derive the same linear mask $\boldsymbol{u}$, corresponding biases should be added up to get the bias of $\boldsymbol{u}$, i.e., $\epsilon_{\boldsymbol{u}} = \sum_{\{\boldsymbol{a}_{\mathbb{T}_z}|U(\boldsymbol{a}_{\mathbb{T}_z})=\boldsymbol{u}\}} 2\times\epsilon_{h,\mathbb{T}_z}(\boldsymbol{a}_{\mathbb{T}_z})\times\epsilon_{g^*,\mathbb{B}}(\boldsymbol{a}_{\mathbb{T}_z})$, where

$$\begin{aligned}U(\boldsymbol{a}_{\mathbb{T}_z}) = &\bigoplus_{i\in\mathbb{T}_z}((a_i[0]\cdot\boldsymbol{g}_{i+1}\oplus a_i[1]\cdot\boldsymbol{g}_{i+6}\oplus a_i[2]\cdot\boldsymbol{g}_{i+15}\oplus a_i[3]\cdot\boldsymbol{g}_{i+19}\oplus a_i[4]\cdot\boldsymbol{g}_{i+22}\oplus a_i[5]\cdot\boldsymbol{g}_{i+27}\oplus a_i[6]\\ &\cdot\boldsymbol{g}_{i+42})\oplus\boldsymbol{g}_{i+38})\oplus\bigoplus_{b\in\mathbb{B}}\boldsymbol{g}_b.\end{aligned}$$

Since the function $U(\boldsymbol{a}_{\mathbb{T}_z})$ is independent on $\boldsymbol{a}_i[7,\cdots,11]$, we need to sum up all biases with a non-zero $\epsilon_{g^*,\mathbb{B}}$ summarized in Table 6, where $\boldsymbol{a}_i[0,\cdots,6]$ is identical and only $\boldsymbol{a}_i[7,\cdots,11]$ varies for $i\in\mathbb{T}_z$. Let V be subset of $\{0,1\}^{12\times4}$ whose elements are 512 corresponding vectors $\boldsymbol{a}_{\mathbb{T}_z}$ with non-zero $\epsilon_{g^*,\mathbb{B}}$ in Table 6. Moreover, there are two special relationships similar to the case of Fruit-v2, as showed in the following.

- $a_0[6]$ and $a_{20}[4]$ (since $\boldsymbol{g}_{0+42} = \boldsymbol{g}_{20+22} = \boldsymbol{g}_{42}$).
- $a_{10}[6]$ and $a_{37}[2]$ (since $\boldsymbol{g}_{10+42} = \boldsymbol{g}_{37+15} = \boldsymbol{g}_{52}$).

Let $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ be the (12×4)-bit vectors generated by the above relationships with the corresponding two positions are 1 but all the other positions are 0. Let $W = \text{span}(\boldsymbol{w}_1, \boldsymbol{w}_2)$ be the linear span whose basis is $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$. Therefore, the bias of $\boldsymbol{u}$ is estimated as $\epsilon_{\boldsymbol{u}} = \sum_{\boldsymbol{w} \in W} \sum_{\boldsymbol{v} \in V} 2 \times \epsilon_{h,\mathbb{T}_z}(\boldsymbol{a}_{\mathbb{T}_z} \oplus \boldsymbol{v} \oplus \boldsymbol{w}) \times \epsilon_{g^*,\mathbb{B}}(\boldsymbol{a}_{\mathbb{T}_z} \oplus \boldsymbol{v})$.

As a result, we searched exhaustively 2^{28} $\boldsymbol{a}_{\mathbb{T}_z}[0, \cdots, 6]$ and found $r = 1048576 = 2^{20}$ $\boldsymbol{u}$ whose absolute value of bias is greater than $\epsilon = 2^{-31.62}$.

6.2 Analysis of Complexities for Attacking Fruit-80

In this subsection, we give the analysis of time and data complexity for recovering the secret key of Fruit-80. Once the correct value of the LFSR initial state is identified by using Algorithm 1, we can also get the sum of the round key bits $\hat{k}_0$. Similarly, $\hat{k}_i$ with $0 \leq i \leq 127$ can be recovered by using Algorithm 1, and further the round key bits. Then, the correct value of the NFSR initial state can be identified by exhaustively searching. Like Fruit-v2, we can run the inverse process of the initialization phase to derive the secret key.

As stated previously in Section 6.1, we have found $r = 2^{20}$ linear masks whose absolute value of bias is greater than $\frac{\epsilon^c}{2} = 2^{-31.62}$ in preparation. According to Theorem 3, we need $\Omega = \frac{2^{\frac{\beta+7}{2}} \sqrt{43 \ln 2}}{\sqrt{r}(\epsilon^c)^2}$ parity-check equations to identify the correct LFSR initial state. Therefore, the data complexity is $D = 4 \times \frac{2^{\frac{\beta+7}{2}} \sqrt{43 \ln 2}}{\sqrt{r}(\epsilon^c)^2}$ keystream bits.

Attack Scenario: $T_1 = \frac{2^{\frac{\beta+7}{2}} \sqrt{43 \ln 2}}{\sqrt{r}(\epsilon^c)^2} + r2^{44-\beta}p_1$, where $p_1 = Q((43 \ln 2 \times r^{-1} 2^{\beta+3})^{\frac{1}{4}})$, $r = 2^{20}$ and $\epsilon^c = 2^{-30.62}$. Moreover, $T_1 = 2^{57.19} \times 2^{\frac{\beta}{2}} + 2^{64-\beta}p_1$, where $p_1 = Q((43 \ln 2 \times 2^{-20} 2^{\beta+3})^{\frac{1}{4}})$ and $p_2 = 1 - p_1$. To make a balance between the two dominant terms of time complexity, we choose $\beta = 4$, and complexities become $T_1 = 2^{59.96}, D = 2^{61.19}$. Furthermore, $p_1 \approx 0.4030$, $p_2 \approx 0.5970$ and $\mu_1 \approx r2^{-\beta}p_1 \approx 26411$, $\mu_2 = r2^{-\beta}p_2 \approx 39125$. Therefore, we choose $th_p = 27626$ and the probabilities are $p_c \approx 1, p_w \approx 2^{-44.00}$. Note that when $\beta = 4$, we have that $2^{64-\beta}p_1 >> 2^{40-\beta}p_2$ and $2^{64-\beta}p_1 >> (43 - \beta) \times 2^{43-\beta}$ are satisfied.

Since we set the threshold th_p such that $p_c \approx 1$ and $p_w \approx 2^{-44.00}$, only the correct value of the LFSR initial state would have a poll greater than th_p in the two majority polls. Therefore, the sum of round key bits $\hat{k}_0$ can be recovered by setting it to the order α_0 of the α_0-th majority poll where the correct value of the LFSR initial state is identified. Similarly, $\hat{k}_i$ with $0 \leq i \leq 127$ can be recovered by carrying out d times Algorithm 1 with the corresponding parity-check equations. Then we can get the round key bits k'_i with $0 \leq i \leq 127$ by solving the linear system of $\hat{k}_i$. When we guess the value of the NFSR initial state, k^*_i with $0 \leq i \leq 127$ can be computed from k'_i and some keystream bits under the condition that the LFSR initial state is known. With exhaustively searching all the possible values of the NFSR initial state, we can find the correct one through the additional keystream bits. Once we know the initial state $(L^{(0)}, N^{(0)})$ and the round key bits k'_i and k^*_i with $0 \leq i \leq 127$, we can run the inverse process of the initialization phase for 160 rounds and derive the secret key.

In conclusion, the total time complexity of our attack is $T = d \times T_1 = 128 \times 2^{59.96} = 2^{66.96}$ and the total data complexity is $2^{66.19}$.

According to Theorem 3 in the precise mode, we can get the following time and data complexities for attacking Fruit-80 with different choices of β, listed in Table 7.

Table 7. Time, memory and data complexities of attacking Fruit-80.

Time	Memory	Data	β
$2^{69.99}$	2^{43}	$2^{56.79}$	0
$2^{68.99}$	2^{42}	$2^{57.79}$	1
$2^{67.98}$	2^{41}	$2^{58.79}$	2
$2^{66.98}$	2^{40}	$2^{59.79}$	3
$2^{66.00}$	2^{39}	$2^{60.79}$	4
$2^{65.09}$	2^{38}	$2^{61.79}$	5
$2^{64.46}$	2^{37}	$2^{62.79}$	6

As the size of bypassed bits β increases, the time complexity decreases temporarily, but more data complexity is required for a successful attack. When $\beta = 6$, the minimum time complexity is achieved, i.e., $T = 2^{64.46}$ and the required data complexity is $D = 2^{62.79}$. When we do not bypass any bit of the LFSR initial state, an attack could be launched with the minimum data complexity $D = 2^{56.79}$ and the time complexity of $2^{69.99}$.

7 Experimental Verification

We verify theoretical analysis of our attacks by applying the above Algorithm 1 and 2 to a toy Grain-like small state stream cipher, a reduced version of Fruit-v2. This toy cipher consists of a 21-bit LFSR whose state at time instant t is denoted by $L^{(t)} = (l_t, \cdots, l_{t+20})$, a linked 19-bit NFSR whose state at time instant t is denoted by $N^{(t)} = (n_t, \cdots, n_{t+18})$, a 40-bit fixed key register denoted by $K = (k_0, \cdots, k_{39})$, and a 6-bit counter register denoted by $C_r = (c_t^0, \cdots, c_t^5)$. The 21-bit LFSR is updated independently by a primitive polynomial as $l_{t+21} = l_t \oplus l_{t+2}$. The 19-bit NFSR is updated recursively as follows, $n_{t+19} = k'_t \oplus l_t \oplus c_t^3 \oplus g(N^{(t)})$, where nonlinear function $g(N^t) = n_t \oplus n_{t+5} \oplus n_{t+10} \oplus n_{t+12}n_{t+3} \oplus n_{t+2}n_{t+13}n_{t+15}$, c_t^3 is the 2-th LSB of the counter C_r, and k'_t is the round key bit generated by the round key function. The round key function is defined as $k'_t = RKF(K, C_r) = k_s k_{y+24} \oplus k_p k_{u+32} \oplus k_{q+12} \oplus k_{r+24}$, where the values of s, y, p, u, q, r are defined from C_r as $s = c_t^0 c_t^1 c_t^2 c_t^3 c_t^4, y = c_t^2 c_t^3 c_t^4, u = c_t^3 c_t^4 c_t^5, p = c_t^0 c_t^1 c_t^2 c_t^3, q = c_t^1 c_t^2 c_t^3 c_t^4$ and $r = c_t^2 c_t^3 c_t^4 c_t^5$. The keystream bit is generated as $z_t = h(L^{(t)}, N^{(t)}) \oplus l_{t+18} \oplus \bigoplus_{b \in \mathbb{B}} n_{t+b}$, where the filtering function is $h(L^{(t)}, N^{(t)}) = l_{t+1}l_{t+2} \oplus l_{t+7}l_{t+11} \oplus n_{t+1}n_{t+17}l_{t+20}$, and the set of the NFSR masking bits is $\mathbb{B} = \{0, 7, 18\}$.

7.1 Environment of experiments

Our attacks on the reduced version of Fruit-v2 have been fully implemented in C++ language on a single PC with Intel Core i5-7600K CPU @ 3.80GHz and 32GB RAM, which is running with Linux 18.04.

7.2 Preparation of linear masks

First, we need to find all the highly biased linear masks and derive the corresponding inverse of matrices $\{F_{\boldsymbol{u}_j}^{-1}\}_{j=1}^{r}$, which are inputs for Algorithm 1.

Just like the situation of Fruit-v2, considering the best linear approximation of the NFSR update function with $2^{-2.42}$ bias $n_{t+19} \approx k'_t \oplus l_t \oplus c_t^3 \oplus n_t \oplus n_{t+5} \oplus n_{t+10}$, the set of taps $\mathbb{T}_z$ is chosen as $\{0, 5, 10, 19\}$. Then, the sum of the keystream bits becomes

$$\bigoplus_{i \in \mathbb{T}_z} z_{t+i} = \bigoplus_{i \in \mathbb{T}_z} l_{t+i+18} \oplus \bigoplus_{b \in \mathbb{B}} l_{t+b} \oplus \bigoplus_{i \in \mathbb{T}_z} h(L^{(t+i)}, N^{(t+i)}) \oplus \bigoplus_{b \in \mathbb{B}} g^*(N^{(t+b)}) \oplus \bigoplus_{b \in \mathbb{B}} \left(k'_{t+b} \oplus c_{t+b}^3\right),$$

where $g^*(N^{(t)}) = n_{t+12}n_{t+3} \oplus n_{t+2}n_{t+13}n_{t+15}$ and $\mathbb{B} = \{0, 7, 18\}$. Consider linear approximate representations of $h(L^{(t+i)}, N^{(t+i)})$ at time instant $t + i$,

$$h(L^{(t+i)}, N^{(t+i)}) \approx \boldsymbol{a}_i[0-4] \cdot (l_{t+i+1}, l_{t+i+2}, l_{t+i+7}, l_{t+i+11}, l_{t+i+20}) \oplus \boldsymbol{a}_i[5-6] \cdot (n_{t+i+1}, n_{t+i+17})$$

with bias $\epsilon_{h,i}(\boldsymbol{a}_{\mathbb{T}_z}) = \pm 2^{-3.42}$ or $\pm 2^{-5.0}$. If one of $a_0[5], a_0[6], a_5[5], a_{10}[5], a_{10}[6]$, $a_{19}[6]$ is 1, the bias is always 0 because $\bigoplus_{b \in \{0,7,18\}} g^*(N^{(t+b)})$ does not involve n_{t+1}, n_{t+17}, n_{t+6}, n_{t+11}, n_{t+27} and n_{t+36}. Therefore, we only evaluated biases of $\bigoplus_{b \in \{0,7,18\}} g^*(N^{(t+b)})$, $\bigoplus_{b \in \{0,7,18\}} g^*(N^{(t+b)}) \oplus n_{22}$, $\bigoplus_{b \in \{0,7,18\}} g^*(N^{(t+b)}) \oplus n_{20}$ and $\bigoplus_{b \in \{0,7,18\}} g^*(N^{(t+b)}) \oplus n_{22} \oplus n_{20}$, and they are $2^{-5.09}, 2^{-7.42}, 2^{-5.83}$ and $-2^{-7.42}$ respectively, i.e., $\epsilon_{g^*,\mathbb{B}}$ only have the four above nonzero values.

For some fixed linear mask $\boldsymbol{u} \in \{0, 1\}^{21}$, we can derive the following linear approximate equation

$$\bigoplus_{i \in \mathbb{T}_z} z_{t+i} \approx L^{(0)} \cdot \left(F^t \times \boldsymbol{u}\right) \oplus \bigoplus_{b \in \mathbb{B}} \left(k'_{t+b} \oplus c_{t+b}^3\right),$$

and it have the bias $\epsilon_{\boldsymbol{u}} = \sum_{\{\boldsymbol{a}_{\mathbb{T}_z} | U(\boldsymbol{a}_{\mathbb{T}_z}) = \boldsymbol{u}\}} 2 \times \epsilon_{h,\mathbb{T}_z}(\boldsymbol{a}_{\mathbb{T}_z}) \times \epsilon_{g^*,\mathbb{B}}(\boldsymbol{a}_{\mathbb{T}_z})$, where $\epsilon_{h,\mathbb{T}_z}(\boldsymbol{a}_{\mathbb{T}_z}) = 2^{4-1} \times \prod_{i \in \mathbb{T}_z} \epsilon_{h,i}(\boldsymbol{a}_{\mathbb{T}_z})$ and

$$U(\boldsymbol{a}_{\mathbb{T}_z}) = \bigoplus_{i \in \mathbb{T}_z} ((a_i[0] \cdot \boldsymbol{g}_{i+1} \oplus a_i[1] \cdot \boldsymbol{g}_{i+2} \oplus a_i[2] \cdot \boldsymbol{g}_{i+7} \oplus a_i[3] \cdot \boldsymbol{g}_{i+11} \oplus a_i[4] \cdot \boldsymbol{g}_{i+20}) \oplus \boldsymbol{g}_{i+18}) \oplus \bigoplus_{b \in \mathbb{B}} \boldsymbol{g}_b.$$

Moreover, there are some special relationships similar to the case of Fruit-v2, and we have six relationships as showed in the following.

- $a_0[2]$ and $a_5[1]$ (since $\boldsymbol{g}_{0+7} = \boldsymbol{g}_{5+2} = \boldsymbol{g}_7$).
- $a_0[3]$ and $a_{10}[0]$ (since $\boldsymbol{g}_{0+11} = \boldsymbol{g}_{10+1} = \boldsymbol{g}_{11}$).
- $a_0[4]$ and $a_{19}[0]$ (since $\boldsymbol{g}_{0+20} = \boldsymbol{g}_{19+1} = \boldsymbol{g}_{20}$).
- $a_5[2]$ and $a_{10}[1]$ (since $\boldsymbol{g}_{5+7} = \boldsymbol{g}_{10+2} = \boldsymbol{g}_{12}$).
- $a_{10}[3]$ and $a_{19}[1]$ (since $\boldsymbol{g}_{10+11} = \boldsymbol{g}_{19+2} = \boldsymbol{g}_{21}$).
- $a_{10}[4]$ and $a_{19}[3]$ (since $\boldsymbol{g}_{10+20} = \boldsymbol{g}_{19+11} = \boldsymbol{g}_{30}$).

To find the linear masks $\boldsymbol{u}$ with high biases, we exhaustively searched 2^{20} $\boldsymbol{a}_i[0-4]$ for all $i \in \mathbb{T}_z$ and summed up those which derive the same $\boldsymbol{u}$. As a result, we found $r = 1024$ linear masks $\boldsymbol{u}$ whose absolute bias is greater than $\epsilon = 2^{-12.03}$.

7.3 Practical Running Time of Algorithm 1

By Theorem 3, to recover the unique correct LFSR initial state, we used $\Omega = \frac{2^{\frac{\beta+7}{2}}\sqrt{21\ln 2}}{\sqrt{1024(2^{-11.03})^2}} \approx 2^{22.49+\frac{\beta}{2}}$ parity-check equations and $D = 4 \times \Omega = 2^{24.49+\frac{\beta}{2}}$ keystream bits. The estimated time complexity of Algorithm 1 is $T_1 = 2^{22.49+\frac{\beta}{2}} + 1024 \times 2^{22-\beta} p_1 \approx 2^{22.49+\frac{\beta}{2}} + 2^{32-\beta} p_1$, where $p_1 = Q((21 \times \ln 2 \times 1024^{-1} \times 2^{\beta+3})^{\frac{1}{4}})$. Next we consider to set the best value of β such that a balance between time complexity of part 1 and part 3 is achieved. With setting $\beta = 5$, we have the minimum time complexity $T_1 \approx 2^{24.99} + 2^{23.42} \approx 2^{25.44}$. Moreover, the value of the LFSR state whose poll is maximum would be chosen as the correct one.

In our experiment, part 1 and part 3 of Algorithm 1 took 211 seconds (about 3.5 minutes) and 16 seconds, respectively. At last, Algorithm 1 outputted the correct LFSR initial state. In general, the experimental results match the theoretical analysis quite well.

7.4 Practical Running Time of Algorithm 2

We inputted the correct LFSR initial state which is obtained from Algorithm 1 into Algorithm 2. From the output function, we can derive that

$$\begin{aligned} n_{t_0+19+j} = {} & z_{t_0+1+j} \oplus (n_{t_0+1+j} \oplus n_{t_0+8+j}) \\ & \oplus n_{t_0+2+j} n_{t_0+18+j} l_{t_0+21+j} \\ & \oplus (l_{t_0+19+j} \oplus l_{t_0+2+j} l_{t_0+3+j} \oplus l_{t_0+8+j} l_{t_0+12+j}) \,. \end{aligned}$$

Like Fruit-v2, we can get the value for $\{n_{t_0+19+j} | j = 0, \cdots, (2^6+19)-1\}$ from the value of $N^{(t_0)} = (n_{t_0}, \cdots, n_{t_0+18})$ and of keystream bits $\{z_{t_0+1+j} | j = 0, \cdots, (2^6+19)-1\}$ using recursively the above equation. Therefore, we just exhaustively search out all the possible value of the NFSR 19-bit initial state and carry out the subroutine of state checking to find the correct one. The estimation of theoretical time complexity is $2^{19} \times (2^6 + 19) \approx 2^{25.38}$.

In our experiment, Algorithm 2 took less 1 second to output the correct NFSR initial state and the secret information bits. In general, the experimental result matches the theoretical analysis.

8 Future Work

In cryptanalysis of Plantlet, Fruit-v2 and Fruit-80, an extremely large amount of data is need to carry out our attack methods. However, the maximum lengths of the produced keystream for Plantlet, Fruit-v2 and Fruit-80 are 2^{30}, 2^{43} and 2^{43} bits in each initialization [25, 28, 15], respectively. In future work, we expect to decrease the data complexity of our attacks. Besides, we are considering to propose a more generic model of Grain-like small state stream ciphers which could cover Lizard.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grants No. 61872359 and No. 61672516) and Youth Innovation Promotion Association CAS.

References

1. Ågren, M., Hell, M., Johansson, T., Meier, W.: Grain-128a: a new version of grain-128 with optional authentication. IJWMC **5**(1), 48–59 (2011). https://doi.org/10.1504/IJWMC.2011.044106, https://doi.org/10.1504/IJWMC.2011.044106
2. Armknecht, F., Mikhalev, V.: On lightweight stream ciphers with shorter internal states. In: Fast Software Encryption - 22nd International Workshop, FSE 2015, Istanbul, Turkey, March 8-11, 2015, Revised Selected Papers. pp. 451–470 (2015). https://doi.org/10.1007/978-3-662-48116-5_22, https://doi.org/10.1007/978-3-662-48116-5_22
3. Banik, S.: Some results on sprout. In: Progress in Cryptology - INDOCRYPT 2015 - 16th International Conference on Cryptology in India, Bangalore, India, December 6-9, 2015, Proceedings. pp. 124–139 (2015). https://doi.org/10.1007/978-3-319-26617-6_7, https://doi.org/10.1007/978-3-319-26617-6_7
4. Banik, S., Barooti, K., Isobe, T.: Cryptanalysis of plantlet. Cryptology ePrint Archive, Report 2019/702 (2019), https://eprint.iacr.org/2019/702
5. Berbain, C., Gilbert, H., Joux, A.: Algebraic and correlation attacks against linearly filtered non linear feedback shift registers. In: Selected Areas in Cryptography, 15th International Workshop, SAC 2008, Sackville, New Brunswick, Canada, August 14-15, Revised Selected Papers. pp. 184–198 (2008). https://doi.org/10.1007/978-3-642-04159-4_12, https://doi.org/10.1007/978-3-642-04159-4_12
6. Berbain, C., Gilbert, H., Maximov, A.: Cryptanalysis of grain. In: Fast Software Encryption, 13th International Workshop, FSE 2006, Graz, Austria, March 15-17, 2006, Revised Selected Papers. pp. 15–29 (2006). https://doi.org/10.1007/11799313_2, https://doi.org/10.1007/11799313_2
7. Biryukov, A., Shamir, A.: Cryptanalytic time/memory/data tradeoffs for stream ciphers. In: Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings. pp. 1–13 (2000). https://doi.org/10.1007/3-540-44448-3_1, https://doi.org/10.1007/3-540-44448-3_1
8. Cannière, C.D.: Trivium: A stream cipher construction inspired by block cipher design principles. In: Information Security, 9th International Conference, ISC 2006, Samos Island, Greece, August 30 - September 2, 2006, Proceedings. pp. 171–186 (2006). https://doi.org/10.1007/11836810_13, https://doi.org/10.1007/11836810_13
9. Canteaut, A., Trabbia, M.: Improved fast correlation attacks using parity-check equations of weight 4 and 5. In: Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding. pp. 573–588 (2000). https://doi.org/10.1007/3-540-45539-6_40, https://doi.org/10.1007/3-540-45539-6_40
10. Chepyzhov, V.V., Johansson, T., Smeets, B.J.M.: A simple algorithm for fast correlation attacks on stream ciphers. In: Fast Software Encryption, 7th International Workshop, FSE 2000, New York, NY, USA, April 10-12, 2000, Proceedings. pp. 181–195 (2000). https://doi.org/10.1007/3-540-44706-7_13, https://doi.org/10.1007/3-540-44706-7_13
11. Chose, P., Joux, A., Mitton, M.: Fast correlation attacks: An algorithmic point of view. In: Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings. pp. 209–221 (2002). https://doi.org/10.1007/3-540-46035-7_14, https://doi.org/10.1007/3-540-46035-7_14
12. Dey, S., Roy, T., Sarkar, S.: Some results on fruit. Des. Codes Cryptography **87**(2-3), 349–364 (2019). https://doi.org/10.1007/s10623-018-0533-y, https://doi.org/10.1007/s10623-018-0533-y
13. Dey, S., Sarkar, S.: Cryptanalysis of full round fruit. IACR Cryptology ePrint Archive **2017**, 87 (2017), http://eprint.iacr.org/2017/087
14. Esgin, M.F., Kara, O.: Practical cryptanalysis of full sprout with TMD tradeoff attacks. In: Selected Areas in Cryptography - SAC 2015 - 22nd International Conference, Sackville, NB, Canada, August 12-14, 2015, Revised Selected Papers. pp. 67–85 (2015). https://doi.org/10.1007/978-3-319-31301-6_4, https://doi.org/10.1007/978-3-319-31301-6_4
15. Ghafari, V.A., Hu, H.: Fruit-80: A secure ultra-lightweight stream cipher for constrained environments. Entropy **20**(3), 180 (2018). https://doi.org/10.3390/e20030180, https://doi.org/10.3390/e20030180
16. Hamann, M., Krause, M., Meier, W.: LIZARD - A lightweight stream cipher for power-constrained devices. IACR Trans. Symmetric Cryptol. **2017**(1), 45–79 (2017). https://doi.org/10.13154/tosc.v2017.i1.45-79, https://doi.org/10.13154/tosc.v2017.i1.45-79
17. Hamann, M., Krause, M., Meier, W., Zhang, B.: Design and analysis of small-state grain-like stream ciphers. Cryptography and Communications **10**(5), 803–834 (2018). https://doi.org/10.1007/s12095-017-0261-6, https://doi.org/10.1007/s12095-017-0261-6
18. Hell, M., Johansson, T., Meier, W.: Grain: a stream cipher for constrained environments. IJWMC **2**(1), 86–93 (2007). https://doi.org/10.1504/IJWMC.2007.013798, https://doi.org/10.1504/IJWMC.2007.013798
19. Johansson, T., Jönsson, F.: Improved fast correlation attacks on stream ciphers via convolutional codes. In: Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding. pp. 347–362 (1999). https://doi.org/10.1007/3-540-48910-X_24, https://doi.org/10.1007/3-540-48910-X_24

20. Johansson, T., Jönsson, F.: Fast correlation attacks through reconstruction of linear polynomials. In: Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2000, Proceedings. pp. 300–315 (2000). https://doi.org/10.1007/3-540-44598-6_19, https://doi.org/10.1007/3-540-44598-6_19
21. Lallemand, V., Naya-Plasencia, M.: Cryptanalysis of full sprout. In: Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I. pp. 663–682 (2015). https://doi.org/10.1007/978-3-662-47989-6_32, https://doi.org/10.1007/978-3-662-47989-6_32
22. Maximov, A., Biryukov, A.: Two trivial attacks on trivium. In: Selected Areas in Cryptography, 14th International Workshop, SAC 2007, Ottawa, Canada, August 16-17, 2007, Revised Selected Papers. pp. 36–55 (2007). https://doi.org/10.1007/978-3-540-77360-3_3, https://doi.org/10.1007/978-3-540-77360-3_3
23. Meier, W., Staffelbach, O.: Fast correlation attacks on certain stream ciphers. J. Cryptology **1**(3), 159–176 (1989). https://doi.org/10.1007/BF02252874, https://doi.org/10.1007/BF02252874
24. Mihaljevic, M.J., Fossorier, M.P.C., Imai, H.: Fast correlation attack algorithm with list decoding and an application. In: Fast Software Encryption, 8th International Workshop, FSE 2001 Yokohama, Japan, April 2-4, 2001, Revised Papers. pp. 196–210 (2001). https://doi.org/10.1007/3-540-45473-X_17, https://doi.org/10.1007/3-540-45473-X_17
25. Mikhalev, V., Armknecht, F., Müller, C.: On ciphers that continuously access the non-volatile key. IACR Trans. Symmetric Cryptol. **2016**(2), 52–79 (2016). https://doi.org/10.13154/tosc.v2016.i2.52-79, https://doi.org/10.13154/tosc.v2016.i2.52-79
26. Siegenthaler, T.: Decrypting a class of stream ciphers using ciphertext only. IEEE Trans. Computers **34**(1), 81–85 (1985). https://doi.org/10.1109/TC.1985.1676518, https://doi.org/10.1109/TC.1985.1676518
27. Todo, Y., Isobe, T., Meier, W., Aoki, K., Zhang, B.: Fast correlation attack revisited - cryptanalysis on full grain-128a, grain-128, and grain-v1. In: Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II. pp. 129–159 (2018). https://doi.org/10.1007/978-3-319-96881-0_5, https://doi.org/10.1007/978-3-319-96881-0_5
28. Vahid Amin Ghafari, H.H., Chen, Y.: Fruit-v2: Ultra-lightweight stream cipher with shorter internal state. Cryptology ePrint Archive, Report 2016/355 (2016), https://eprint.iacr.org/2016/355
29. Zhang, B., Feng, D.: Multi-pass fast correlation attack on stream ciphers. In: Selected Areas in Cryptography, 13th International Workshop, SAC 2006, Montreal, Canada, August 17-18, 2006 Revised Selected Papers. pp. 234–248 (2006). https://doi.org/10.1007/978-3-540-74462-7_17, https://doi.org/10.1007/978-3-540-74462-7_17
30. Zhang, B., Gong, X.: Another tradeoff attack on sprout-like stream ciphers. In: Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II. pp. 561–585 (2015). https://doi.org/10.1007/978-3-662-48800-3_23, https://doi.org/10.1007/978-3-662-48800-3_23
31. Zhang, B., Gong, X., Meier, W.: Fast correlation attacks on grain-like small state stream ciphers. IACR Trans. Symmetric Cryptol. **2017**(4), 58–81 (2017). https://doi.org/10.13154/tosc.v2017.i4.58-81, https://doi.org/10.13154/tosc.v2017.i4.58-81