IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2017, No. 4, pp. 58–81. DOI:10.13154/tosc.v2017.i4.58-81

Fast Correlation Attacks on Grain-like Small State Stream Ciphers

Bin Zhang[1,2,3], Xinxin Gong[1,2] and Willi Meier[4]

[1] TCA Labaratory, State Key Laboratory of Computer Science (SKLCS), Institute of Software, Chinese Academy of Sciences, Beijing, China
{zhangbin,gongxinxin}@tca.iscas.ac.cn
[2] State Key Laboratory of Cryptology, P.O.Box 5159, Beijing, 100878, China
[3] University of Chinese Academy of Sciences, Beijing, 100049, China
[4] University of Applied Sciences and Arts Northwestern Switzerland (FHNW), Windisch, Switzerland
willi.meier@fhnw.ch

Abstract. In this paper, we study the security of Grain-like small state stream ciphers by fast correlation attacks, which are commonly regarded as classical cryptanalytic methods against LFSR-based stream ciphers. We extend the cascaded structure adopted in such primitives in general and show how to restore the full internal state part-by-part if the non-linear combining function meets some characteristic. As a case study, we present a key recovery attack against Fruit, a tweaked version of Sprout that employs key-dependent state updating in the keystream generation phase. Our attack requires $2^{62.8}$ Fruit encryptions and $2^{22.3}$ keystream bits to determine the 80-bit secret key. Practical simulations on a small-scale version confirmed our results.

Keywords: Cryptanalysis · Stream ciphers · Fast correlation attacks · Linear approximation · Fruit

1 Introduction

Design of secure small state stream ciphers for constrained hardware applications is an important line of work in recent years, which extends the design paradigm domain of lightweight stream ciphers in theory and provides interesting primitives for low power devices like passive RFID tags in practice. Such small state ciphers often utilize a key-dependent state updating in both, initialization and keystream generation phases, to thwart time/memory/data tradeoff attacks [5], and the non-linear feedback shift registers (NFSR) are main building blocks to resist (fast) correlation [6, 7, 8, 19, 20, 24] and algebraic attacks [9, 10]. So far, there are several candidates available in this domain, i.e., Sprout [1], Fruit [13], Plantlet [25] and Lizard [15], which are designed in an ad-hoc way following the above essential ideas.

On the other hand, the lack of a well-understood theoretical study in this domain apparently restricts the confidence that people have on such primitives. The event that Sprout has been broken shortly after its publication [2, 29, 12, 21, 22], has put heavy shadows on this kind of ciphers. To remedy the situation, three new primitives are proposed, i.e., Fruit, Plantlet and Lizard, which are designed carefully with the lessons learned from Sprout in mind. It is expected that lower area, thus power consumption could be achieved by using a fixed non-volatile secret key and the key-dependent state updating in an adequate way. This motivates us to study the security of these small primitives against a new type of attacks that is well-tailored for them.

Received: 2017-06-01, Revised: 2017-09-01, Accepted: 2017-11-01, Published: 2017-12-15

In this paper, we study the security of these Grain-like small state stream ciphers by fast correlation attacks, the classical cryptanalytic methods against LFSR-based stream ciphers. We first define a generalized model for such small state ciphers extracted from the real-world primitives, which adopts a cascaded structure to connect several NFSRs and exploits the key-dependent state updating in the keystream generation phase. It is shown that if the non-linear combining function used to generate the final keystream has some pseudo-linear properties, i.e., for each state candidate of the independently updating register, the combining function becomes linear with respect to the involved variables coming from the other registers, then the whole system could be converted into a degraded sub-system which is a linearly filtered NFSR in nature, whether dynamic or not. It is further demonstrated that we could restore the full internal state of the model in a divide-and-conquer manner by utilizing the fast correlation attacks on random probabilistic linear systems derived from the degraded sub-system. The well-known Fast Walsh Transform (FWT) [26] plays a central role in building the efficient distinguishers in the attack. Based on our attack, new general design criteria are suggested for the model. As a case study, we describe a key recovery attack against the Fruit stream cipher [13], a tweaked version of Sprout to address the previous weaknesses and suggested for practical applications in constrained hardware environments. Our attack requires $2^{62.8}$ Fruit encryptions and $2^{22.3}$ keystream bits to determine the 80-bit secret key, which clearly breaks the 80-bit security claim. Note that there is another attack on the target version of Fruit in [11] with higher complexity and the most recent version of Fruit taking into account the attacks presented herein is scratched by a weak key attack in [16], while our attack applies to all the keys and reveals a set of insights on such small state designs in its own right. In addition, our attack works for any round key generation algorithm, e.g., whether round keys are balanced or not (previous attacks on Sprout in [21] and Fruit in [11] exploit specific properties of round key construction). Practical experiments on a small-scale version of the primitive well confirmed our results.

This paper is structured as follows. In Section 2, we present a brief description of Fruit and propose a generic model for Grain-like NFSR-based small state stream ciphers, which inherits the spirit of the corresponding real-world designs. Then in Section 3, we present a high-level general description of our attack. In Section 4, we discuss how to construct parity-checks based on the specified property of the non-linear combining function both in the model and in Fruit itself. In Section 5, a dedicated fast correlation attack is developed on the generic model, interleaved by the application to Fruit at each step with the theoretical complexity analysis. Section 6 provides the experimental results. Finally, some conclusions are drawn in Section 7 with the new general design criteria on such primitives.

2 The Grain-like Small State Stream Ciphers

In this section, we will first provide a brief description of the Fruit stream cipher proposed in [13] as far as relevant to our work, and then present the generic model for Grain-like small state stream ciphers.

2.1 Description of Fruit

Fruit is a bit-oriented stream cipher adopting a Grain-like structure and utilizes an 80-bit secret key $K = (k_0, k_1, ..., k_{79})$ and a 70-bit public initial value $IV = (iv_0, iv_1, ..., iv_{69})$ to generate the keystream. As shown in Fig. 1, there are five parts involved, a 43-bit LFSR whose state at time t is denoted by $S^t = (s_t, s_{t+1}, ..., s_{t+42})$, a linked 37-bit NFSR whose state at time t is denoted by $N^t = (n_t, n_{t+1}, ..., n_{t+36})$, an 80-bit fixed key register, and two counter registers, a 7-bit $C_r = (c_t^0, ..., c_t^6)$ and an 8-bit $C_c = (c_t^7, ..., c_t^{14})$, allocated for the round key function and for the initialization/keystream generation, respectively. Note that c_t^6 and c_t^{14} are the LSBs of the two counters respectively. These two counters increase by 1 at each tick, and work continually, i.e., after they become all ones, counting from zeros to all ones again.

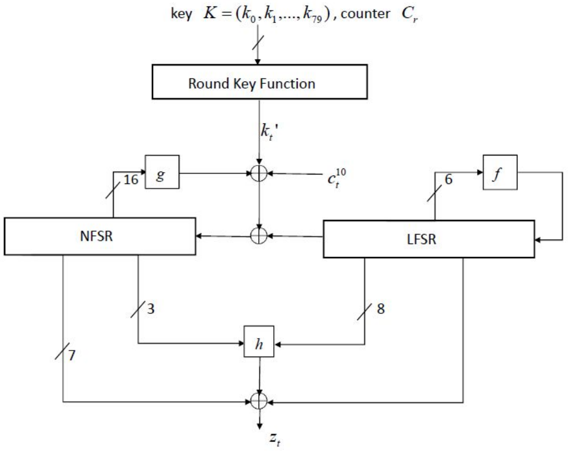

Figure 1: The keystream generation of Fruit

The 43-bit LFSR is updated independently and recursively by a linear function f as $s_{t+43} = f(S^t) = s_t \oplus s_{t+8} \oplus s_{t+18} \oplus s_{t+23} \oplus s_{t+28} \oplus s_{t+37}$. The NFSR is updated recursively by a non-linear feedback function g defined as

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

where k_t' is the round key bit, and c_t^{10}, the 4-th LSB of C_c, is the counter bit generated at time t. Define the values of sv, y, u, p, q, r from the counter C_r as $sv = c_t^0 c_t^1 c_t^2 c_t^3 c_t^4 c_t^5$, $y = c_t^3 c_t^4 c_t^5$, $u = c_t^4 c_t^5 c_t^6$, $p = c_t^0 c_t^1 c_t^2 c_t^3 c_t^4$, $q = c_t^1 c_t^2 c_t^3 c_t^4 c_t^5$ and $r = c_t^3 c_t^4 c_t^5 c_t^6$, then the round key bit k_t' is generated by combining 6 bits of the key as $k_t' = k_{sv}k_{y+64} \oplus k_p k_{u+72} \oplus k_{q+32} \oplus k_{r+64}$.

Given the internal state (S^t, N^t) at time t, the filter function h produces $h_t = n_{t+1}s_{t+15} \oplus s_{t+1}s_{t+22} \oplus n_{t+35}s_{t+27} \oplus n_{t+33}s_{t+11} \oplus s_{t+6}s_{t+33}s_{t+42}$, and the keystream bit is generated as

$$z_t = h_t \oplus s_{t+38} \oplus n_t \oplus n_{t+7} \oplus n_{t+13} \oplus n_{t+19} \oplus n_{t+24} \oplus n_{t+29} \oplus n_{t+36}.$$

During the key/IV setup phase, first load the key bits in the following way: $n_i = k_i, 0 \leq i \leq 36$; $s_i = k_{i+37}, 0 \leq i \leq 42$. Then pad the IV to 130 bits by concatenating 1 bit one and 9 bit zeros to the head of IV and 50 bit zeros to the end of IV as

$$IV' = \underbrace{1000000000}_{10}\underbrace{iv_0 iv_1 ... iv_{69}}_{70}\underbrace{000...000}_{50} = iv_0' iv_1' \cdots iv_{129}'.$$

In the first step of the initialization, set $C_r = C_c = 0$ and run the cipher 130 rounds as follows: the LFSR is updated as $s_{t+43} = z_t \oplus iv_t' \oplus f(S^t)$, while the NFSR is updated as $n_{t+37} = z_t \oplus iv_t' \oplus k_t' \oplus s_t \oplus c_t^{10} \oplus g(N^t)$, and no keystream bit is generated. Next comes the second step of the initialization, first set all bits of C_r equal to the LSBs of the NFSR except the last bit of C_r that is set to the LSB of the LFSR, and then set s_{130} to 1. Hereafter run the cipher 80 rounds without the feedback in the LFSR and NFSR, i.e.,

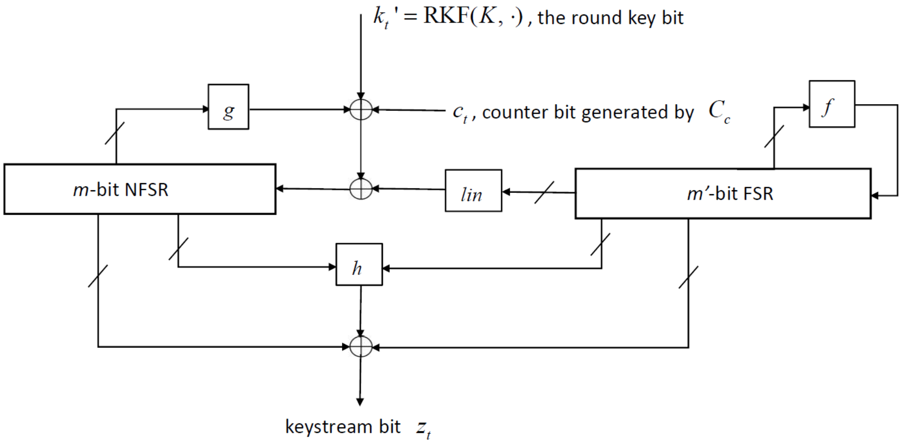
$k_t' = \mathrm{RKF}(K, \cdot)$, the round key bit
c_t, counter bit generated by C_c
g
f
m-bit NFSR
lin
m'-bit FSR
h
keystream bit z_t

Figure 2: The generic model for the Grain-like small state stream ciphers

the LFSR updating function is changed to $s_{t+43} = f(S^t)$, the NFSR updating function is changed to $n_{t+37} = k_t' \oplus s_t \oplus c_t^{10} \oplus g(N^t)$, and no keystream bit is generated.

After the initialization phase, the keystream generation phase starts and the keystream bits are produced.

2.2 The Generalized Model

Inspired by Fruit as well as by other similar primitives, now we present the generalized model for Grain-like small state stream ciphers as depicted in Fig.2, which is helpful in the sense that we could study some special properties/choices more clearly in a unified framework. The following notations will be used in the model.

- $N^t = (n_t, n_{t+1}, ..., n_{t+m-1})$, the m-bit internal state of the cascaded NFSR at time t.

- $S^t = (s_t, s_{t+1}, ..., s_{t+m'-1})$, the m'-bit internal state of the FSR at time t, which updates independently in a invertible way, with a either linear or non-linear feedback function, in the keystream generation phase.

- $K = (k_0, k_1, ..., k_{l-1})$, the l-bit secret key, which satisfies $l \leq m + m' \leq 2l$.

- $k_t' = \mathrm{RKF}(K, \cdot)$, the round key bit generated at time t.

- C_c, a round counter for the NFSR state updating.

- c_t, a counter bit generated by the counter C_c at time t.

- $P_{S^t} = \{s_{t+\alpha_1}, s_{t+\alpha_2}, ..., s_{t+\alpha_{j_1}}\}$, a subset of S^t and the input variables of the filter function h, introduced below, from the FSR, $0 \leq \alpha_1 < \alpha_2 < ... < \alpha_{j_1} \leq m' - 1$.

- $P_{N^t} = \{n_{t+\beta_1}, n_{t+\beta_2}, ..., n_{t+\beta_{j_2}}\}$, a subset of N^t and the input variables of the filter function h from the NFSR, $0 \leq \beta_1 < \beta_2 < ... < \beta_{j_2} \leq m - 1$.

- $Q_{S^t} = \{s_{t+\sigma_1}, s_{t+\sigma_2}, ..., s_{t+\sigma_{r_1}}\}$, a subset of S^t and the input variables of the linear Boolean function ϕ, introduced below, from the FSR, $0 \leq \sigma_1 < \sigma_2 < ... < \sigma_{r_1} \leq m' - 1$.

- $Q_{N^t} = \{n_{t+\eta_1}, n_{t+\eta_2}, ..., n_{t+\eta_{r_2}}\}$, a subset of N^t and the input variables of the linear Boolean function ϕ from the NFSR, $0 \leq \eta_1 < \eta_2 < ... < \eta_{r_2} \leq m - 1$. There are five Boolean functions involved in the model: a (either linear or non-linear) Boolean function f, a non-linear Boolean function g, a linear Boolean function lin, a linear Boolean function ϕ and a non-linear filter function h. At each step, the FSR is updated independently by f, while the NFSR is updated by g with the round key bit k_t', the counter bit c_t, and some bits of the FSR as inputs. The round key bit k_t' at time t is generated by the round key function RKF, which takes the secret key K as part of the input. The model is specified by the following items in the keystream generation phase.

(1) Components

- Denote the initial state of the FSR by S^0. It is updated recursively and independently by f as $S^{t+1} = (s_{t+1}, s_{t+2}, \cdots, s_{t+m'})$ with $s_{t+m'} = f(S^t)$. We assume this process is invertible, and the inverse process is $S^{t-1} = (s_{t-1}, s_t, \cdots, s_{t+m'-2})$ with $s_{t-1} = f^{-1}(S^t)$.

- Denote the initial state of the NFSR by N^0. It is updated recursively by the non-linear Boolean function g, along with some elements generated by the secret key, the counter C_c, and the FSR, shown below.

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

  where k'_t is the round key bit generated at time t, c_t is a counter bit generated by the counter C_c at time t, and $lin(\cdot)$ is a linear Boolean function which represents the xor of some inputs from the FSR state S^t. Similarly, we assume this non-linear process is invertible, and the inverse process is computed as $n_{t-1} = k'_{t-1} \oplus c_{t-1} \oplus lin'(S^{t-1}) \oplus g^{-1}(N^t)$.

- A linear Boolean function $\phi(\cdot)$ from GF$(2)^{r_1+r_2}$ to GF(2) is used as one part of the output function, defined as

$$\phi_t \triangleq \phi(Q_{S^t}, Q_{N^t}) = \left(\bigoplus\nolimits_{i=1}^{r_1} s_{t+\sigma_i}\right) \oplus \left(\bigoplus\nolimits_{i=1}^{r_2} n_{t+\eta_i}\right),$$

  which takes r_1 input values from the FSR state S^t and r_2 input values from the NFSR state N^t, respectively.

- A filter function $h : \text{GF}(2)^{j_1+j_2} \to \text{GF}(2)$, $h_t \triangleq h(P_{S^t}, P_{N^t})$ is used as the other part of the output function, which takes j_1 input values from the FSR state S^t and j_2 input values from the NFSR state N^t, respectively.

- The output function $z(\cdot) = h(\cdot) \oplus \phi(\cdot)$, which generates the keystream $\{z_t\}_{t\geq 0}$ based on the inputs taken from both S^t and N^t for $t = 0, 1, ...$

(2) Keystream generation. As just stated, the keystream bit z_t is recursively computed as $z_t = h_t \oplus \phi_t,\ t = 0, 1, ...$

(3) Assumed properties

- (3.1). we assume the RKF is periodic, so are the round key bits. Let p be the least positive integer such that $k'_{t+p} = k'_t$ for any $t \geq 0$, i.e., the round key bits repeat in a cycle of length p. Besides, our model could also cover the case that the counter bits c_t are unknown. In this case, we only assume that c_t is periodic, i.e., there exists a least positive integer q such that $c_{t+q} = c_t$ for any $t \geq 0$.

- (3.2). (*Pseudo-linearity.*) For the filter function $h : \text{GF}(2)^{j_1+j_2} \to \text{GF}(2)$, $h_{P_{S^t}}(P_{N^t})$ with $P_{S^t} \in \text{GF}(2)^{j_1}$ and $P_{N^t} \in \text{GF}(2)^{j_2}$ is used to replace $h(\cdot)$ for a fixed given value of P_{S^t}. We assume for any choice of P_{S^t}, $h_{P_{S^t}}$ to be a *linear* Boolean function with respect to the inputs from P_{N^t}. Note that the FSR is updated independently, accordingly, for any possible value of the FSR initial state S^0, the outputs of the model depend linearly on the NFSR bits, thus the degraded system can be interpreted as a linearly filtered NFSR involving the secret round key bits, which have a known cycle p.

Here we stress that the NFSR in the model can be further decomposed into a series of cascaded smaller NFSRs, which could also be treated by our cryptanalysis. It is obvious that our generalized model could cover Grain v1 [17] and Fruit described above as special cases, but not Plantlet [25] and Lizard [15] so far, the reason for the latter is that the Assumed property (3.2) does not hold in the cases of Plantlet and Lizard (and the designers of Lizard were aware of the fast correlation attack on Fruit to be presented later).

For Grain v1, it fits into the model with the parameters $m = 80$, $m' = 80$ and $l = 80$, which are the lengths of the NFSR, the LFSR and the secret key, respectively. Since the secret key is not involved in the keystream generation phase, thus for any time t the round key bit k'_t is always 0, so is the counter bit c_t. Besides, the keystream bit z_t at time t is generated as $z_t = h(s_{t+3}, s_{t+25}, s_{t+46}, s_{t+64}, n_{t+63}) \oplus n_{t+1} \oplus n_{t+2} \oplus n_{t+4} \oplus n_{t+10} \oplus n_{t+31} \oplus n_{t+43} \oplus n_{t+56}$, where

$$h(\cdot) = s_{t+25} \oplus \underline{n_{t+63}} \oplus s_{t+3}s_{t+64} \oplus s_{t+46}s_{t+64} \oplus s_{t+64}\underline{n_{t+63}} \oplus s_{t+3}s_{t+25}s_{t+46}$$

$$\oplus\, s_{t+3}s_{t+46}s_{t+64} \oplus s_{t+3}s_{t+46}\underline{n_{t+63}} \oplus s_{t+25}s_{t+46}\underline{n_{t+63}} \oplus s_{t+46}s_{t+64}\underline{n_{t+63}},$$

where we use the underline to show the pseudo-linearity of $h(\cdot)$. To fit into the model, we have $P_{S^t} = \{s_{t+3}, s_{t+25}, s_{t+46}, s_{t+64}\}$, $Q_{S^t} = \varnothing$, $P_{N^t} = \{n_{t+63}\}$ and $Q_{N^t} = \{n_{t+1}, n_{t+2}, n_{t+4}, n_{t+10}, n_{t+31}, n_{t+43}, n_t$ Note that $h_{P_{S^t}}$ is a linear Boolean function with the inputs from P_{N^t}, accordingly, for any fixed value of the LFSR initial state of Grain v1, the outputs will depend linearly on the NFSR bits. However, due to the fact that the length of the LFSR in Grain v1 is already 80-bit, our attack will have a time complexity well above 2^{80}, thus becomes inefficient.

Next, Fruit fits into the model with the parameters $m = 37$, $m' = 43$ and $l = 80$, representing the lengths of the NFSR, the LFSR and the secret key, respectively. In Fruit, the secret key is involved in the NFSR state updating, and the round key bit k'_t is generated as

$$k'_t = k_{sv}k_{y+64} \oplus k_p k_{u+72} \oplus k_{q+32} \oplus k_{r+64} \triangleq \mathrm{RKF}(K, C^t_r),$$

where $C_r = (c^0_t, ..., c^6_t)$ is a 7-bit round counter allocated for the round key function. Note that k'_t is periodic with a cycle of length $p = 128$, i.e., $k'_{t+128} = k'_t$ for any $t \geq 0$. There is an 8-bit counter $C_c = (c^7_t, ..., c^{14}_t)$ in Fruit, where the 4th LSB c^{10}_t of C_c is employed in the NFSR state updating. It should be noted that C_c is deterministic at any time t, and is independent of the key, thus c^{10}_t is known. Actually, we figure out that the counter bit c^{10}_t is periodic with a cycle of length $q = 32$, i.e., $c^{10}_{t+32} = c^{10}_t$ for any $t \geq 0$, and in each cycle, this bit takes the values $\underbrace{0, 0, ..., 0}_{16}\underbrace{1, 1, ..., 1}_{16}$.

Given the internal state (S^t, N^t) of Fruit at time t, the keystream bit is generated as

$$z_t = h(s_{t+1}, s_{t+6}, s_{t+11}, s_{t+15}, s_{t+22}, s_{t+27}, s_{t+33}, s_{t+42}, n_{t+1}, n_{t+33}, n_{t+35})$$

$$\oplus\, s_{t+38} \oplus n_t \oplus n_{t+7} \oplus n_{t+13} \oplus n_{t+19} \oplus n_{t+24} \oplus n_{t+29} \oplus n_{t+36},$$

where $h(\cdot) = s_{t+15}\underline{n_{t+1}} \oplus s_{t+1}s_{t+22} \oplus s_{t+27}\underline{n_{t+35}} \oplus s_{t+11}\underline{n_{t+33}} \oplus s_{t+6}s_{t+33}s_{t+42}$ with the pseudo-linearity shown by underline. Fitting into the model, we have $P_{S^t} = \{s_{t+1}, s_{t+6}, s_{t+11}, s_{t+15}, s_{t+22}, s_{t+27}, s_{t+33}, s_{t+42}\}$, $Q_{S^t} = \{s_{t+38}\}$, $P_{N^t} = \{n_{t+1}, n_{t+33}, n_{t+35}\}$ and $Q_{N^t} = \{n_t, n_{t+7}, n_{t+13}, n_{t+19}, n_{t+24}, n_{t+29}, n_{t+36}\}$. It is clear that $h_{P_{S^t}}$ is a linear Boolean function with the input variables n_{t+1}, n_{t+33} and n_{t+35} for Fruit; accordingly, for any fixed value of the LFSR initial state of Fruit, the output keystream will depend linearly on the NFSR bits.

3 A General Description of Our Attack

We first present a high-level overview of our attack. The goal is to recover both the FSR state and the NFSR state at a fixed time instance which is consistent with the given keystream, and the round key bits within one repetition cycle.

The main idea is as follows. In the generic model depicted in Figure 2, the FSR is updated independently without the influence of the NFSR, the counter bits and the round key bits. For small state stream ciphers, the internal state size of the FSR cannot be too large, thus a suitable scale exhaustive search over all the possible values of the independently updated FSR is often feasible. Further, we could run the FSR forwards and backwards to obtain any value of its output and peel off the non-linearity of the involved h function. Combined with the pseudo-linearity of the h function, we could derive a random probabilistic linear system on the initial NFSR variables with a rather high bias. In fact, compared to the work in [3] which made the linear approximations of both the feedback function of the NFSR in Grain v0 and the output function a number of times, now we make the linear approximation of the feedback function of the NFSR in the model *only once*, without any linear approximation of the non-linear output filter function. Hence, the bias of the random probabilistic linear system is quite different from one half, which will facilitate the construction of the low-weight parity-checks to further reduce the dimension of the initial NFSR variables. Then instead of solving the parity-checks directly, we could just construct a distinguisher via the well-known FWT. The correct FSR candidate could be easily identified from the full Walsh spectrum of some derived function. Thus, the FSR is restored independently of the NFSR in the model, which results in a divide-and-conquer recovery of the whole internal state in presence of unknown round key bits. Finally, the internal state of the NFSR could be retrieved in a multi-pass manner [28] later with the complexity much lower than that of recovering the FSR. For the specific ciphers, one period of the round key bits and the original secret key could be derived with a much lower complexity according to the mechanism of the primitive and the definition of the round key function employed.

Formally, a high-level description of our attack is depicted in Algorithm 1.

Algorithm 1 Fast correlation attack on the generic model in Figure 2

```
Parameters: $m, m', D$
Input:  A keystream segment $\mathbf{z} = (z_0, z_1, \ldots, z_{D-1})$
1st phase: Prepare the parity-checks
for each possible value of $S^0$ do
    use the method in section 4.1 to derive the probabilistic system
    construct the parity-checks by the method in section 4.2
end for
2nd phase: Recover the full internal state matching with $\mathbf{z}$
for each possible value of $S^0$ do
    use the distinguisher in section 5.2 to check it
for each passed candidate of $S^0$ do
    recover the NFSR state part-by-part in section 5.3
for each candidate of the full internal state do
    check it and restore the secret key accordingly in section 5.4
```

In the following, we will interleave the generic idea of Algorithm 1 and the concrete attack on Fruit, i.e., each step of the generic idea will be followed by the corresponding procedure on Fruit, to demonstrate our attack in details.

4 Preparing the Parity-checks

As each fast correlation attack needs parity-checks, in this section, we show how to derive a random probabilistic linear system and construct the desirable parity-checks accordingly, based on the pseudo-linearity of the output function when combining the input variables, for both the generic model and the concrete Fruit case.

4.1 Degrading the System

Suppose the adversary somehow knows the initial state $S^0 = (s_0, s_1, ..., s_{m'-1})$ of the FSR and the Assumed properties (3.1) and (3.2) hold. Now the attacker can run the FSR forwards and backwards to remove its protection over the output keystream, the resultant system becomes a *linearly* filtered NFSR, involving the periodic round key bits. Given the NFSR state $N^t = (n_t, n_{t+1}, ..., n_{t+m-1})$ at time t, we rewrite the keystream bit z_t as

$$z_t = \bigoplus_{i=1}^{j_2} \psi_t^i \cdot n_{t+\beta_i} \oplus \bigoplus_{i=1}^{r_2} n_{t+\eta_i} \oplus \psi_t^0, \tag{2}$$

where the coefficients ψ_t^i, $i = 0, 1, ..., j_2$, depend on the FSR state at time t. For Fruit, for example, the keystream bit generated at time t can be written as

$$\begin{aligned} z_t =& (s_{t+15}\underline{n_{t+1}} \oplus s_{t+11}\underline{n_{t+33}} \oplus s_{t+27}\underline{n_{t+35}}) \\ & \oplus (\underline{n_t} \oplus \underline{n_{t+7}} \oplus \underline{n_{t+13}} \oplus \underline{n_{t+19}} \oplus \underline{n_{t+24}} \oplus \underline{n_{t+29}} \oplus \underline{n_{t+36}}) \\ & \oplus (s_{t+38} \oplus s_{t+1}s_{t+22} \oplus s_{t+6}s_{t+33}s_{t+42}) \end{aligned} \tag{3}$$

which corresponds to $\psi_t^0 = s_{t+38} \oplus s_{t+1}s_{t+22} \oplus s_{t+6}s_{t+33}s_{t+42}$, $\psi_t^1 = s_{t+15}$, $\psi_t^2 = s_{t+11}$, $\psi_t^3 = s_{t+27}$.

In the following, we will show that though there is the masking of the secret information, any internal state variable of the NFSR can be expressed as a linear combination of the NFSR state variable at a fixed time instance τ and of some keystream bits, under the condition that the FSR initial state S^0 is known, by extending the technique in [4]. These linear relations are derived by using the output function of Eq.(2) recursively. Here we only discuss the following two cases to illustrate the process, while the other cases can be handled analogously in a dynamic way by induction.

Case 1 (*Model*). Suppose $\eta_{r_2} > \beta_{j_2}$ holds. In this case, η_{r_2} is the highest index value of the initial NFSR variables $(n_t, n_{t+1}, \cdots, n_{t+m-1})$ involved in the keystream bit z_t. Let $\tau = 0$, we can express any internal state variable n_{m+i} $(i \geq 0)$ as a linear combination of the initial NFSR state variables $N^0 = (n_0, n_1, ..., n_{m-1})$ and of some keystream bits.

According to Eq.(2), $z_{m-\eta_{r_2}}$ is the first keystream bit dependent on n_m, we can write

$$z_{m-\eta_{r_2}} = n_m \oplus \bigoplus_{i=1}^{j_2} \psi_{m-\eta_{r_2}}^i \cdot n_{m-\eta_{r_2}+\beta_i} \oplus \bigoplus_{i=1}^{r_2-1} n_{m-\eta_{r_2}+\eta_i} \oplus \psi_{m-\eta_{r_2}}^0,$$

thus we have

$$n_m = z_{m-\eta_{r_2}} \oplus \bigoplus_{i=1}^{j_2} \psi_{m-\eta_{r_2}}^i \cdot n_{m-\eta_{r_2}+\beta_i} \oplus \bigoplus_{i=1}^{r_2-1} n_{m-\eta_{r_2}+\eta_i} \oplus \psi_{m-\eta_{r_2}}^0,$$

i.e., n_m is expressed as a linear combination of $N^0 = (n_0, n_1, ..., n_{m-1})$ and of a keystream bit $z_{m-\eta_{r_2}}$. Now we assume that for all $i: \ 0 \leq i < j$, all the bits n_{m+i} are expressed as a linear combination of the NFSR initial state variables and of keystream bits. Note that $z_{m+j-\eta_{r_2}}$ is the first keystream bit that depends on n_{m+j}, which indicates that

$$n_{m+j} = z_{m+j-\eta_{r_2}} \oplus \bigoplus_{i=1}^{j_2} \psi_{m+j-\eta_{r_2}}^i \cdot n_{m+j-\eta_{r_2}+\beta_i} \oplus \bigoplus_{i=1}^{r_2-1} n_{m+j-\eta_{r_2}+\eta_i} \oplus \psi_{m+j-\eta_{r_2}}^0.$$

That is, the variable n_{m+j} is expressed as a linear combination of a keystream bit $z_{m+j-\eta_{r_2}}$ and of the NFSR variables n_{m+i} with $i < j$. By induction, n_{m+j} can finally be expressed as a linear combination of the NFSR initial state variables $n_0, n_1, ..., n_{m-1}$ and of some keystream bits, under the condition that the FSR initial state S^0 is known.

Case 1 (*Fruit case*). For Fruit, we have $\eta_{r_2} = 36$ and $\beta_{j_2} = 35$, thus $\eta_{r_2} > \beta_{j_2}$ holds. Assume the initial state $S^0 = (s_0, s_1, ..., s_{42})$ of the LFSR is known, we will express each NFSR state variable n_i $(i \geq 37)$ as a linear combination of the NFSR initial state variables $N^0 = (n_0, n_1, ..., n_{36})$ and of some keystream bits.

First we consider how to express n_{37}. From Eq.(3), we have $z_1 = (s_{16}\underline{n_2} \oplus s_{12}\underline{n_{34}} \oplus s_{28}\underline{n_{36}}) \oplus (\underline{n_1} \oplus \underline{n_8} \oplus \underline{n_{14}} \oplus \underline{n_{20}} \oplus \underline{n_{25}} \oplus \underline{n_{30}} \oplus \underline{n_{37}}) \oplus (s_{39} \oplus s_2 s_{23} \oplus s_7 s_{34} s_{43})$, and z_1 is the first keystream bit dependent on n_{37}, thus we have

$$\begin{aligned} n_{37} =& z_1 \oplus (s_{16}\underline{n_2} \oplus s_{12}\underline{n_{34}} \oplus s_{28}\underline{n_{36}}) \oplus (\underline{n_1} \oplus \underline{n_8} \oplus \underline{n_{14}} \oplus \underline{n_{20}} \oplus \underline{n_{25}} \oplus \underline{n_{30}}) \\ & \oplus (s_{39} \oplus s_2 s_{23} \oplus s_7 s_{34} s_{43}). \end{aligned}$$

That is, we have expressed n_{37} as a linear combination of the NFSR initial state variables and of the keystream bit z_1. Next for n_{38}, we have $n_{38} = z_2 \oplus (s_{17}\underline{n_3} \oplus s_{13}\underline{n_{35}} \oplus s_{29}\underline{n_{37}}) \oplus (\underline{n_2} \oplus \underline{n_9} \oplus \underline{n_{15}} \oplus \underline{n_{21}} \oplus \underline{n_{26}} \oplus \underline{n_{31}}) \oplus (s_3 s_{24} \oplus s_8 s_{35} s_{44} \oplus s_{40})$, i.e., n_{38} depends on n_{37}. By a simple substitution, we get

$$\begin{aligned} n_{38} = & (z_2 \oplus s_{29} z_1) \oplus (s_{29} s_{16}\underline{n_2} \oplus s_{17}\underline{n_3} \oplus s_{29} s_{12}\underline{n_{34}} \oplus s_{13}\underline{n_{35}} \oplus s_{29} s_{28}\underline{n_{36}} \\ & \oplus s_{29}\underline{n_1} \oplus \underline{n_2} \oplus s_{29}\underline{n_8} \oplus \underline{n_9} \oplus s_{29}\underline{n_{14}} \oplus \underline{n_{15}} \oplus s_{29}\underline{n_{20}} \oplus \underline{n_{21}} \oplus s_{29}\underline{n_{25}} \\ & \oplus \underline{n_{26}} \oplus s_{29}\underline{n_{30}} \oplus \underline{n_{31}}) \oplus s_{29}(s_{39} \oplus s_2 s_{23} \oplus s_7 s_{34} s_{43}) \oplus s_{40} \oplus s_3 s_{24} \oplus s_8 s_{35} s_{44}. \end{aligned}$$

Note that in this process, the effects of the round key bits have been masked successfully. Thus if we carry on this procedure continually, we can get the desirable expressions for n_{37+2}, n_{37+3},...,$n_{37+(D-1)}$ from the keystream bits $z_1, z_2, ..., z_D$, where D is a given parameter.

Case 2 (*Model*). Suppose $\eta_1 < \beta_1$ holds, thus $\eta_1 < \beta_i$ for $2 \leq i \leq j_2$. In this case, η_1 is the lowest index value of the initial NFSR variables $(n_t, n_{t+1}, \cdots, n_{t+m-1})$ involved in the keystream bit z_t. For a fixed time instance τ, we will express any internal state variable $n_{\tau-j}$ $(j \geq 1)$ as a linear combination of the NFSR state variables $N^\tau = (n_\tau, n_{\tau+1}, ..., n_{\tau+m-1})$ and of some keystream bits.

According to Eq.(2), $z_{\tau-1-\eta_1}$ is the first keystream bit which is dependent on $n_{\tau-1}$, we rewrite it as

$$z_{\tau-1-\eta_1} = n_{\tau-1} \oplus \bigoplus_{i=1}^{j_2} \psi^i_{\tau-1-\eta_1} \cdot n_{\tau-1-\eta_1+\beta_i} \oplus \bigoplus_{i=2}^{r_2} n_{\tau-1-\eta_1+\eta_i} \oplus \psi^0_{\tau-1-\eta_1},$$

thus we have

$$n_{\tau-1} = z_{\tau-1-\eta_1} \oplus \bigoplus_{i=1}^{j_2} \psi^i_{\tau-1-\eta_1} \cdot n_{\tau-1-\eta_1+\beta_i} \oplus \bigoplus_{i=2}^{r_2} n_{\tau-1-\eta_1+\eta_i} \oplus \psi^0_{\tau-1-\eta_1},$$

i.e., $n_{\tau-1}$ is expressed as a linear combination of the NFSR state variables $N^\tau = (n_\tau, n_{\tau+1}, ..., n_{\tau+m-1})$ and of a keystream bit $z_{\tau-1-\eta_1}$.

Now we assume that for all $i : 1 \leq i < j$, all the bits $n_{\tau-i}$ are expressed as a linear combination of the NFSR state variables from N^τ and of some keystream bits. Note that $z_{\tau-j-\eta_1}$ is the first keystream bit that depends on $n_{\tau-j}$, and we have

$$n_{\tau-j} = z_{\tau-j-\eta_1} \oplus \bigoplus_{i=1}^{j_2} \psi^i_{\tau-j-\eta_1} \cdot n_{\tau-j-\eta_1+\beta_i} \oplus \bigoplus_{i=2}^{r_2} n_{\tau-j-\eta_1+\eta_i} \oplus \psi^0_{\tau-j-\eta_1}.$$

That is, the variable $n_{\tau-j}$ is expressed as a linear combination of a keystream bit $z_{\tau-j-\eta_1}$ and of the NFSR variables $n_{\tau-i}$ with $i < j$. By induction, $n_{\tau-j}$ can finally be expressed as a linear combination of the NFSR state variables from $N^\tau = (n_\tau, n_{\tau+1}, ..., n_{\tau+m-1})$ and of some keystream bits, under the condition that the FSR initial state S^0 is known.

Case 2 (*Fruit case*). For Fruit, we have $\eta_1 = 0$ and $\beta_1 = 1$, thus $\eta_1 < \beta_1$ holds. Similarly, assume the initial state $S^0 = (s_0, s_1, ..., s_{42})$ of the LFSR is known, we will express each NFSR state variable $n_{\tau-j}$ $(j \geq 1)$ as a linear combination of the NFSR state variables $N^\tau = (n_\tau, n_{\tau+1}, ..., n_{\tau+36})$ and of some keystream bits.

First we consider how to express $n_{\tau-1}$. From Eq.(3), we find that $z_{\tau-1}$ is the first keystream bit dependent on $n_{\tau-1}$, and $z_{\tau-1} = (s_{\tau+14}\underline{n_\tau} \oplus s_{\tau+10}\underline{n_{\tau+32}} \oplus s_{\tau+26}\underline{n_{\tau+34}}) \oplus (\underline{n_{\tau-1}} \oplus \underline{n_{\tau+6}} \oplus \underline{n_{\tau+12}} \oplus \underline{n_{\tau+18}} \oplus \underline{n_{\tau+23}} \oplus \underline{n_{\tau+28}} \oplus \underline{n_{\tau+35}}) \oplus (s_{\tau+37} \oplus s_\tau s_{\tau+21} \oplus s_{\tau+5}s_{\tau+32}s_{\tau+41})$, thus we have

$$\begin{aligned} n_{\tau-1} = z_{\tau-1} &\oplus (s_{\tau+14}\underline{n_\tau} \oplus s_{\tau+10}\underline{n_{\tau+32}} \oplus s_{\tau+26}\underline{n_{\tau+34}}) \\ &\oplus (\underline{n_{\tau+6}} \oplus \underline{n_{\tau+12}} \oplus \underline{n_{\tau+18}} \oplus \underline{n_{\tau+23}} \oplus \underline{n_{\tau+28}} \oplus \underline{n_{\tau+35}}) \\ &\oplus (s_{\tau+37} \oplus s_\tau s_{\tau+21} \oplus s_{\tau+5}s_{\tau+32}s_{\tau+41}) \end{aligned}$$

That is, we have expressed $n_{\tau-1}$ as a linear combination of the NFSR state variables $N^\tau = (n_\tau, n_{\tau+1}, ..., n_{\tau+36})$ and of the keystream bit $z_{\tau-1}$. Next for $n_{\tau-2}$, we have $n_{\tau-2} = z_{\tau-2} \oplus (s_{\tau+13}\underline{n_{\tau-1}} \oplus s_{\tau+9}\underline{n_{\tau+31}} \oplus s_{\tau+25}\underline{n_{\tau+33}}) \oplus (\underline{n_{\tau+5}} \oplus \underline{n_{\tau+11}} \oplus \underline{n_{\tau+17}} \oplus \underline{n_{\tau+22}} \oplus \underline{n_{\tau+27}} \oplus \underline{n_{\tau+34}}) \oplus (s_{\tau+36} \oplus s_{\tau-1}s_{\tau+20} \oplus s_{\tau+4}s_{\tau+31}s_{\tau+40})$, i.e., $n_{\tau-2}$ depends on $n_{\tau-1}$. By a simple substitution, we get

$$\begin{aligned} \underline{n_{\tau-2}} = (z_{\tau-2} \oplus s_{\tau+13}z_{\tau-1}) &\oplus (s_{\tau+13}s_{\tau+10}\underline{n_{\tau+32}} \oplus s_{\tau+9}\underline{n_{\tau+31}} \oplus s_{\tau+13}s_{\tau+26}\underline{n_{\tau+34}} \\ &\oplus s_{\tau+25}\underline{n_{\tau+33}} \oplus s_{\tau+13}s_{\tau+14}\underline{n_\tau} \oplus s_{\tau+13}\underline{n_{\tau+6}} \oplus \underline{n_{\tau+5}} \oplus s_{\tau+13}\underline{n_{\tau+12}} \oplus \underline{n_{\tau+11}} \\ &\oplus s_{\tau+13}\underline{n_{\tau+18}} \oplus \underline{n_{\tau+17}} \oplus s_{\tau+13}\underline{n_{\tau+23}} \oplus \underline{n_{\tau+22}} \oplus s_{\tau+13}\underline{n_{\tau+28}} \oplus \underline{n_{\tau+27}} \\ &\oplus s_{\tau+13}\underline{n_{\tau+35}} \oplus \underline{n_{\tau+34}}) \oplus s_{\tau+13}(s_{\tau+37} \oplus s_\tau s_{\tau+21} \oplus s_{\tau+5}s_{\tau+32}s_{\tau+41}) \\ &\oplus (s_{\tau+36} \oplus s_{\tau-1}s_{\tau+20} \oplus s_{\tau+4}s_{\tau+31}s_{\tau+40}), \end{aligned}$$

hence we have expressed $n_{\tau-2}$ as a linear combination of the NFSR state variables $N^\tau = (n_\tau, n_{\tau+1}, ..., n_{\tau+36})$ and of the keystream bits $z_{\tau-1}$ and $z_{\tau-2}$. We carry on this procedure continually, and finally can get the desirable expressions for $n_{\tau-3}$, $n_{\tau-4}$,...,$n_{\tau-D}$ from the keystream bits $z_{\tau-1}, z_{\tau-2}, ..., z_{\tau-D}$, where D is a given parameter.

Complexity. In [4], the time/memory complexities for expressing D NFSR state variables for a linearly filtered NFSR are given as $m \cdot D$ computations and $(m+1) \cdot D$ bits of memory, respectively. In our attack, we need to repeat the above process $2^{m'}$ times for all the possible initial states of the FSR. Thus the total time complexity is $2^{m'} \cdot m \cdot D$. When applied to Fruit, the time complexity of this step is $C_1 = 2^{43} \cdot 37 \cdot D = 2^{48.21} \cdot D$.

4.2 Building the Parity-checks

Now we discuss how to derive the desirable parity-check relations from the linear approximations of the NFSR feedback function g (or g^{-1}) and from the fact that the secret round key bits involved in the NFSR updating form a periodic sequence.

4.2.1 Expressing the NFSR variables

Assume there are R linearly independent linear approximations for g having the same largest bias $\epsilon > 0$ and let $\mathbf{a}^j = (\mathbf{a}_0^j, \mathbf{a}_1^j, \cdots, \mathbf{a}_{m-1}^j)$ be a binary vector of length m, then with the probability $\frac{1}{2} + \epsilon$, each linear approximation for g, corresponding to the linear mask $\mathbf{a}^j$ and a sign b_j, could be written as

$$g(N^t) = \mathbf{a}^j \cdot (N^t)^T \oplus b_j = \mathbf{a}^j \cdot (n_t, n_{t+1}, ..., n_{t+m-1})^T \oplus b_j, \text{ for } j = 1, 2, ..., R,$$

where the dot operator $\cdot$ between a row vector and a column vector represents the usual inner GF(2)-product and the operator $(\cdot)^T$ is the transpose of a row vector in the vectorial scenario. For the inverse process g^{-1} of the NFSR updating function, the corresponding linear approximation is

$$g^{-1}(N^t) = (\mathbf{a}^j \lll 1) \cdot (n_t, n_{t+1}, ..., n_{t+m-1})^T \oplus b_j, \text{ for } j = 1, 2, ..., R.$$

For simplicity, we only illustrate how to build the parity-checks for the above **Case 1 (*Model*)**, while a similar procedure can be carried out for **Case 2 (*Model*)** by using the linear approximations for g^{-1}. In addition to these two cases, other cases can be handled dynamically in a similar way.

Suppose $\eta_{r_2} > \beta_{j_2}$ holds. In this case, any internal state variable n_{m+i} $(i \geq 0)$ can be expressed as a linear combination of the NFSR initial state variables $(n_0, n_1, ..., n_{m-1})$ and of some keystream bits, under the condition that the FSR initial state S^0 is known. Suppose the keystream bits $z_{m-\eta_{r_2}+i}$, $i = 0, 1, ..., D-1$, are available, we first determine the expression of the D NFSR state variables n_{m+i}, $i = 0, 1, ..., D-1$, which can be accomplished in time $2^{m'} \cdot m \cdot D$. We represent the derived expressions in matrix form as

$$(n_0, n_1, \cdots, n_{m+D-1}) = N^0\mathbf{G} \oplus \chi \oplus v = (n_0, n_1, ..., n_{m-1})\mathbf{G} \oplus \chi \oplus v,$$

where the $m \times (m+D)$ matrix $\mathbf{G}$ is formed as $\mathbf{G} = [\mathbf{I}, \mathbf{g}_m, \cdots, \mathbf{g}_{m+D-1}]$ with the first m columns corresponding to the identity matrix $\mathbf{I}$ and $\mathbf{g}_i$ $(m \leq i \leq m+D-1)$ being the column vector, $\chi = (0, 0, \cdots, 0, \chi_m, \cdots \chi_{m+D-1})$ and $v = (0, 0, \cdots, 0, v_m, \cdots, v_{m+D-1})$ are $(m+D)$-bit vectors depending on the FSR initial state and the keystream bits $z_{m-\eta_{r_2}+i}$ for $0 \leq i \leq D-1$. Then for $j = m, ..., m+D-1$, we have

$$n_j = N^0 \cdot \mathbf{g}_j \oplus \chi_j \oplus v_j = (n_0, n_1, ..., n_{m-1}) \cdot \mathbf{g}_j \oplus \chi_j \oplus v_j, \tag{4}$$

where χ_j and v_j are the jth coordinates of χ and v, respectively.

Fruit case. We have $m = 37$, $m' = 43$, $\eta_{r_2} = 36$, $\beta_{j_2} = 35$, and $\eta_{r_2} > \beta_{j_2}$ holds. Applying the FWT to the feedback function g of the NFSR in Fruit, we have found that there are $R = 7$ linearly independent linear approximations for g having the same largest bias $\epsilon \triangleq 2^{-4.6}$, i.e., $n_t \oplus n_{t+10} \oplus n_{t+20}$, $n_t \oplus n_{t+10} \oplus n_{t+20} \oplus n_{t+12}$, $n_t \oplus n_{t+10} \oplus n_{t+20} \oplus n_{t+3}$, $n_t \oplus n_{t+10} \oplus n_{t+20} \oplus n_{t+14}$, $n_t \oplus n_{t+10} \oplus n_{t+20} \oplus n_{t+25}$, $n_t \oplus n_{t+10} \oplus n_{t+20} \oplus n_{t+8}$, $n_t \oplus n_{t+10} \oplus n_{t+20} \oplus n_{t+18}$. For $i = 0, 1, ..., 36$, denote by $\mathbf{I}_i$ the 37-bit ith row vector of the identity matrix $\mathbf{I}_{37\times37}$, where the i-th bit is 1 and all the other bits are 0. Then we have $b_j = 0$ for $j = 1, 2, ..., 7$, and

$$\begin{aligned}
\mathbf{a}^1 &= \mathbf{I}_0 \oplus \mathbf{I}_{10} \oplus \mathbf{I}_{20} \\
\mathbf{a}^2 &= \mathbf{I}_0 \oplus \mathbf{I}_{10} \oplus \mathbf{I}_{20} \oplus \mathbf{I}_{12} = \mathbf{a}^1 \oplus \mathbf{I}_{12} \\
\mathbf{a}^3 &= \mathbf{I}_0 \oplus \mathbf{I}_{10} \oplus \mathbf{I}_{20} \oplus \mathbf{I}_{3} = \mathbf{a}^1 \oplus \mathbf{I}_{3} \\
\mathbf{a}^4 &= \mathbf{I}_0 \oplus \mathbf{I}_{10} \oplus \mathbf{I}_{20} \oplus \mathbf{I}_{14} = \mathbf{a}^1 \oplus \mathbf{I}_{14} \\
\mathbf{a}^5 &= \mathbf{I}_0 \oplus \mathbf{I}_{10} \oplus \mathbf{I}_{20} \oplus \mathbf{I}_{25} = \mathbf{a}^1 \oplus \mathbf{I}_{25} \\
\mathbf{a}^6 &= \mathbf{I}_0 \oplus \mathbf{I}_{10} \oplus \mathbf{I}_{20} \oplus \mathbf{I}_{8} = \mathbf{a}^1 \oplus \mathbf{I}_{8} \\
\mathbf{a}^7 &= \mathbf{I}_0 \oplus \mathbf{I}_{10} \oplus \mathbf{I}_{20} \oplus \mathbf{I}_{18} = \mathbf{a}^1 \oplus \mathbf{I}_{18}.
\end{aligned}$$

For a given ω, let $D = 128(\omega - 1) + 1$, provided the keystream bits $z_0, z_1, ..., z_{D-1}$, the expressions of the NFSR variables n_{37+i}, $i = 0, 1, ..., D-1$, can be computed in time $C_1 = 2^{43} \cdot 37 \cdot D = 2^{48.21} \cdot D$ for all the possible LFSR initial states.

Next, we come back to the model and proceed to use the state updating function g of the NFSR to derive the probabilistic linear system. According to the NFSR updating in Eq.(1) and the linear approximation $(\mathbf{a}^j, b_j)$ for g, with the probability $\frac{1}{2} + \epsilon$, we have

$$n_{t+m} = k'_t \oplus c_t \oplus lin(S^t) \oplus \mathbf{a}^j \cdot (n_t, n_{t+1}, ..., n_{t+m-1})^T \oplus b_j, \text{ for } j = 1, 2, ..., R.$$

From the Assumed property (3.1), the round key bits k'_t and the counter bits c_t are unknown, and k'_t has a cycle of length p and c_t has a cycle of length q. Let d be the least common multiple of two integers p and q, i.e., $d = \texttt{lcm}(p, q)$, then $k'_{t+di} \oplus c_{t+di} = k'_t \oplus c_t$ for $t = 0, 1, ..., d-1$ and any $i \geq 0$. Accordingly with the same probability, we have

$$n_{t+m+di} \oplus \mathbf{a}^j \cdot (n_{t+di}, n_{t+1+di}, ..., n_{t+m-1+di})^T \oplus lin(S^{t+di}) = k'_t \oplus c_t \oplus b_j,\ j = 1, 2, ..., R.$$

Let $D = d(\omega - 1) + 1$, where ω is a parameter to be determined later. By choosing $t = 0$, we get

$$n_{m+di} \oplus \mathbf{a}^j \cdot (n_{di}, n_{1+di}, ..., n_{m-1+di})^T \oplus lin(S^{di}) = k'_0 \oplus c_0 \oplus b_j,$$

for $j = 1, 2, ..., R$ and any $i \geq 0$. For brevity, we define the following notations:

$$\begin{aligned}
\mathbf{u}_{i,j} &\triangleq \mathbf{g}_{m+di} \oplus (\mathbf{a}^j \cdot \mathbf{g}_{di}, \mathbf{a}^j \cdot \mathbf{g}_{di+1}, ..., \mathbf{a}^j \cdot \mathbf{g}_{di+m-1})^T,\\
Z_{i,j} &\triangleq \chi_{m+di} \oplus \mathbf{a}^j \cdot (\chi_{di}, \chi_{1+di}, ..., \chi_{m-1+di})^T,\\
v_{i,j} &\triangleq \upsilon_{m+di} \oplus \mathbf{a}^j \cdot (\upsilon_{di}, \upsilon_{1+di}, ..., \upsilon_{m-1+di})^T \oplus lin(S^{di}),
\end{aligned}$$

Substitute each n_j by (4) into the above equation, it can be further written as

$$N^0 \cdot \mathbf{u}_{i,j} \oplus Z_{i,j} \oplus v_{i,j} = k'_0 \oplus c_0 \oplus b_j,\ i = 0, 1, ..., \omega - 1, j = 1, 2, ..., R,$$

where $\mathbf{u}_{i,j}$ is a column vector in the inner GF(2)-product with $N^0 = (n_0, n_1, ..., n_{m-1})$, and $Z_{i,j}$ and $v_{i,j}$ are the derived binary values. Note that $\mathbf{u}_{i,j}$ and $v_{i,j}$ are totally determined by the FSR initial state and the corresponding linear approximation for g, $Z_{i,j}$ depends on the keystream information, the FSR initial state and the linear approximation employed. From the above, we actually achieve a noisy system with $\omega' \triangleq \omega R$ linear equations on the unknowns $(n_0, n_1, ..., n_{m-1})$, k'_0 and c_0, which can be rewritten as

$$N^0 \cdot \mathbf{u}_{i,j} \oplus Z_{i,j} \oplus v_{i,j} = k'_0 \oplus c_0 \oplus b_j \oplus e_{i,j},\ j = 1, 2, ..., R,\ i = 0, 1, ..., \omega - 1, \tag{5}$$

where $e_{i,j}$ is the random noise introduced by the corresponding linear approximation $(\mathbf{a}^j, b_j)$ for the NFSR state feedback function g satisfying $\Pr(e_{i,j} = 0) = \frac{1}{2} + \epsilon$ for all $j = 1, 2, ..., R$ and $i = 0, 1, ..., \omega - 1$. We notice that the complexity to construct the above system of equations is related to the Hamming weight of each $\mathbf{a}^j$, and is at most $m^2\omega'$. In our attack, we need to repeat the same process $2^{m'}$ times for all the possible initial states of the FSR. Thus the total time complexity is at most $2^{m'} \cdot m^2 \cdot \omega'$.

Fruit case. In Fruit, the counter bit c^{10}_t is known and has the period $q = 32$, while the round key bit k'_t has the period $p = 128$. For each possible LFSR state in Fruit, we can obtain a linear system in the form of Eq.(5) with $\omega' = 7 \cdot \omega$ linear equations, all holding with the bias $\epsilon = 2^{-4.6}$ and $b_j = 0$ for $1 \leq j \leq 7$, shown in the following Eq.(6).

$$(n_0, n_1, \cdots, n_{36}) \cdot \mathbf{u}_{i,j} \oplus Z_{i,j} \oplus v_{i,j} = k'_0 \oplus c^{10}_0 \oplus e_{i,j},\ j = 1, 2, ..., 7,\ i = 0, 1, ..., \omega - 1, \tag{6}$$

This can be accomplished in time $C_2 = 2^{43} \times [(1 \times 37 \times 3) + (6 \times 37 \times 2)] \cdot \omega = 2^{52.12} \cdot \omega$, which is derived from the concrete forms of the linear mask $\mathbf{a}^j$ $(1 \leq j \leq 7)$ for Fruit.

4.2.2 Constructing the Parity-checks

Now we are ready for constructing the parity-checks from the derived probabilistic system. As above, we first take a look at the generic model, and then the concrete Fruit case.

Let $\texttt{Low}_x(\mathbf{a})/\texttt{High}_x(\mathbf{a})$ be the value of the vector $\mathbf{a}$ on the least/most significant x positions. As in previous fast correlation attacks [7], now we try to find some linear combinations of columns which vanish on the lowest significant bits to reduce the secret dimension, i.e., we look for some κ-tuple of (usually $\kappa = 2$ or $\kappa = 4$ to cancel the secret information) column vectors $(\mathbf{u}_{i_1,j_1}, ..., \mathbf{u}_{i_\kappa,j_\kappa})$ satisfying $\texttt{Low}_{m-m_1}(\mathbf{u}_{i_1,j_1} \oplus ... \oplus \mathbf{u}_{i_\kappa,j_\kappa}) =$

$(0, ..., 0)'$. For $0 \leq i \leq \omega - 1$ and $j = 1, 2, \cdots, R$, we will regard the column vectors $\mathbf{u}_{i,j}$ in Eq.(5) as random vectors and $v_{i,j}$ as random variables. For the ω' column vectors $\mathbf{u}_{i,j}$, there is an expected number of $\Omega = \binom{\omega'}{\kappa} \cdot 2^{-(m-m_1)} \approx \frac{\omega'^{\kappa}}{\kappa! 2^{m-m_1}}$ such κ-tuples. We define the following notations to denote the indices of the t-th such column tuple for $t = 1, 2, ..., \Omega$,

$$\begin{aligned} &\mathcal{U}_t = \mathtt{High}_{m_1}\left(\bigoplus_{r=1}^{\kappa} \mathbf{u}_{i_r,j_r}^{(t)}\right),\ \mathcal{Z}_t = \bigoplus_{r=1}^{\kappa} Z_{i_r,j_r}^{(t)},\ \mathcal{V}_t = \bigoplus_{r=1}^{\kappa} v_{i_r,j_r}^{(t)}, \\ &\mathcal{B}_t = \bigoplus_{r=1}^{\kappa} b_{j_r}^{(t)},\ \mathcal{E}_t = \bigoplus_{r=1}^{\kappa} e_{i_r,j_r}^{(t)} \end{aligned}$$

and accordingly we obtain

$$(n_0, n_1, ..., n_{m_1-1}) \cdot \mathcal{U}_t \oplus \mathcal{Z}_t \oplus \mathcal{V}_t = \mathcal{B}_t \oplus \mathcal{E}_t,\ t = 1, 2, ..., \Omega \tag{7}$$

for $t = 1, 2, ..., \Omega$. Let $\epsilon_F = 2^{\kappa}\epsilon^{\kappa} > 0$, from the piling-up lemma [23], we have

$$\Pr(\mathcal{E}_t = 0) = \frac{1}{2} + 2^{\kappa-1}\epsilon^{\kappa} = \frac{1}{2}(1 + \epsilon_F).$$

Fruit case. Set the positive integer m_1 such that $0 < m_1 < 37$, we look for $\kappa = 2$ column vectors $(\mathbf{u}_{i_1,j_1}, \mathbf{u}_{i_2,j_2})$ satisfying $\mathtt{Low}_{m-m_1}(\mathbf{u}_{i_1,j_1} \oplus \mathbf{u}_{i_2,j_2}) = (0, ..., 0)'$. For the ω' column vectors $\mathbf{u}_{i,j}$, there are an expected number of $\Omega = \binom{\omega'}{2} \cdot 2^{-(m-m_1)} \approx \omega'^2 \cdot 2^{-(m-m_1+1)}$ such pairs. To fulfill this task, a sort-and-merge procedure is applied. First, these vectors $\mathbf{u}_{i,j}$ are sorted into 2^{m-m_1} equivalence classes according to their values on the least significant $m - m_1$ positions, thus any two vectors in the same equivalence class will have the same value on these positions. Then we look at each pair of vectors $(\mathbf{u}_{i_1,j_1}, \mathbf{u}_{i_2,j_2})$ in each equivalence class, deriving that $\mathtt{Low}_{m-m_1}(\mathbf{u}_{i_1,j_1} \oplus \mathbf{u}_{i_2,j_2}) = (0, ..., 0)'$. For each value of the LFSR initial state, we need to repeat this process. This can be finished in time $C_3 = 2^{43} \cdot (\omega' + \Omega)$.

Denote the t-th pair of columns by $(\mathbf{u}_{i_1,j_1}^{(t)}, \mathbf{u}_{i_2,j_2}^{(t)})$ for $t = 1, 2, ..., \Omega$. Similarly we define the notations that $\mathcal{Z}_t = Z_{i_1,j_1}^{(t)} \oplus Z_{i_2,j_2}^{(t)}$, $\mathcal{V}_t = v_{i_1,j_1}^{(t)} \oplus v_{i_2,j_2}^{(t)}$, $\mathcal{E}_t = e_{i_1,j_1}^{(t)} \oplus e_{i_2,j_2}^{(t)}$ and $\mathcal{U}_t = \mathtt{High}_{m_1}\left(\mathbf{u}_{i_1,j_1}^{(t)} \oplus \mathbf{u}_{i_2,j_2}^{(t)}\right)$, thus we derive $\Omega = \omega'^2 \cdot 2^{-(m-m_1+1)}$ equations as follows,

$$(n_0, n_1, ..., n_{m_1-1}) \cdot \mathcal{U}_t \oplus \mathcal{Z}_t \oplus \mathcal{V}_t = \mathcal{E}_t,\ t = 1, 2, ..., \Omega \tag{8}$$

Here $\Pr(\mathcal{E}_t = 0) = \frac{1}{2} + 2\epsilon^2 \triangleq \frac{1}{2}(1 + \epsilon_F)$, where $\epsilon = 2^{-4.6}$ and $\epsilon_F = 4\epsilon^2 = 2^{-7.2}$.

All together, for each value of the LFSR initial state of Fruit, we have derived a corresponding linear system of the form (8) which involves the first m_1 bits of the NFSR initial state.

5 A Divide-and-Conquer Fast Correlation Attack

In this section, we launch a divide-and-conquer fast correlation attack against the generic model and demonstrate it on Fruit itself. First, we provide a brief review on the multi-pass strategy exploited in our attack, then the detailed process is given with theoretical analysis.

5.1 The Multi-pass Strategy

After building the desirable parity-checks, we first make an independent recovery of the FSR initial state S^0. Conditioned on the restored FSR bits, we continue to retrieve the NFSR initial state N^0 part-by-part as follows. Precisely, we divide the NFSR initial state $N^0 = (n_0, n_1, ..., n_{m-1})$ into several smaller parts as shown below and try to recover them part-by-part in a sequential order.

$$(\underbrace{n_0, ..., n_{m_1-1}}_{m_1}, \underbrace{n_{m_1}, ..., n_{m_1+m_2-1}}_{m_2}, \underbrace{n_{m_1+m_2}, ...}_{...}, ..., n_{m-1})$$

We first recover the first m_1 bits of N^0, i.e., $(n_0, n_1, ..., n_{m_1-1})$ conditioned on both the FSR initial state candidates and the known keystream. Once we recovered the first $\sum_{j=1}^{i} m_j$ bits of the NFSR initial state, we enter the next pass to determine the next m_{i+1} bits of N^0 conditioned on the recovered information and the known keystream bits, which will have a much lower complexity than the recovery of the first m_1 bits. That is, we adopt a multi-pass strategy [28] to determine the whole NFSR initial state.

5.2 Independent Recovery of the FSR Initial State

As in the last section, we first show how to recover the FSR initial state of the model, and then present the illustration to Fruit. For any value of S^0 in the model, we first rewrite the parity-checks Eq.(7) for the model as

$$\begin{aligned}&(n'_0, n'_1, ..., n'_{m_1-1}) \cdot \mathcal{U}_t \oplus \mathcal{B}_t \oplus \mathcal{Z}_t \oplus \mathcal{V}_t\\&\quad = (n_0, n_1, ..., n_{m_1-1}) \cdot \mathcal{U}_t \oplus (n'_0, n'_1, ..., n'_{m_1-1}) \cdot \mathcal{U}_t \oplus \mathcal{E}_t,\ t = 1, 2, \cdots, \Omega\end{aligned}$$

where $(n'_0, n'_1, ..., n'_{m_1-1})$ is the guessed value of the first m_1-bit of the NFSR initial state, $\mathcal{U}_t$ and $\mathcal{V}_t$ are computed from the currently guessed value S^0 and the linear approximation for g, while $\mathcal{Z}_t$ is computed from the given keystream segment and the guessed value S^0. Here we introduce the target function $\Delta(i_1^{(t)}, j_1^{(t)}, ..., i_\kappa^{(t)}, j_\kappa^{(t)})$ as

$$\Delta(i_1^{(t)}, j_1^{(t)}, ..., i_\kappa^{(t)}, j_\kappa^{(t)}) = (n_0, n_1, ..., n_{m_1-1}) \cdot \mathcal{U}_t \oplus (n'_0, n'_1, ..., n'_{m_1-1}) \cdot \mathcal{U}_t \oplus \mathcal{E}_t.$$

It is obvious that if both the FSR initial state S^0 and the first m_1-bit $(n'_0, n'_1, ..., n'_{m_1-1})$ of the NFSR initial state are correctly guessed, $\mathcal{U}_t$ and $\mathcal{V}_t$ will take their true values consistent with the keystream, and $\Delta(i_1^{(t)}, j_1^{(t)}, ..., i_\kappa^{(t)}, j_\kappa^{(t)}) = \mathcal{E}_t$, thus $\Pr(\Delta = 0) = \frac{1}{2}(1+\epsilon_F)$, where $\epsilon_F = 2^\kappa \epsilon^\kappa$ (> 0). Besides, if S^0 is correctly guessed and $(n'_0, n'_1, ..., n'_{m_1-1})$ is wrongly guessed, then $\mathcal{U}_t$ and $\mathcal{V}_t$ will have their true values, and $\Delta(i_1^{(t)}, j_1^{(t)}, ..., i_\kappa^{(t)}, j_\kappa^{(t)}) = (n_0 \oplus n'_0, n_1 \oplus n'_1, ..., n_{m_1-1} \oplus n'_{m_1-1}) \cdot \mathcal{U}_t \oplus \mathcal{E}_t$. Since $\mathcal{U}_t = \mathtt{High}_{m_1}\left(\bigoplus_{r=1}^{\kappa} \mathbf{u}_{i_r,j_r}^{(t)}\right)$, then we further get

$$\begin{aligned}\Delta(i_1^{(t)}, j_1^{(t)}, ..., i_\kappa^{(t)}, j_\kappa^{(t)}) &= (n_0 \oplus n'_0, n_1 \oplus n'_1, ..., n_{m_1-1} \oplus n'_{m_1-1}) \cdot \mathcal{U}_t \oplus \mathcal{E}_t\\&= \bigoplus_{t: n_t \oplus n'_t = 1} \left(\bigoplus_{r=1}^{\kappa} \mathbf{u}_{i_r,j_r}^{(t)}\right) \oplus \mathcal{E}_t.\end{aligned}$$

As described above we have $\Pr(\mathbf{u}_{i_r,j_r}^{(t)} = 0) = \Pr(\mathbf{u}_{i_r,j_r}^{(t)} = 1) = 1/2$, then $\Delta(i_1^{(t)}, j_1^{(t)}, ..., i_\kappa^{(t)}, j_\kappa^{(t)})$ has the distribution $\Pr(\Delta = 0) = 1/2$. Finally, if S^0 is wrongly guessed, whatever the guess $(n'_0, n'_1, ..., n'_{m_1-1})$ is, $\Delta(i_1^{(t)}, j_1^{(t)}, ..., i_\kappa^{(t)}, j_\kappa^{(t)})$ will always have the distribution $\Pr(\Delta = 0) = 1/2$.

To fulfill the above observation, we could exhaustively search over all the possible guesses of the FSR initial state, and for each guess, we evaluate the parity checks to count the number of the vanishing Δs, for all the possible guesses $(n'_0, n'_1, ..., n'_{m_1-1})$. The straightforward method, as that in [7], has a time complexity of $2^{m'} 2^{m_1} \Omega$, which is obviously an inefficient attack. Instead our approach differs from [7] with the exploitation of FWT as proposed in [8]. For each possible value of S^0, we regroup the Ω parity-checks according to the pattern of $\mathcal{U}_t$ and define an integer-valued function h_{S^0} as

$$h_{S^0}(\mathbf{a}) = \sum_{t:\ \mathcal{U}_t = \mathbf{a}} (-1)^{\mathcal{Z}_t \oplus \mathcal{V}_t \oplus \mathcal{B}_t},$$

for all the patterns appearing in the Ω parity-checks; if a pattern does not occur, we just let $h_{S^0}(\mathbf{a}) = 0$ at that point. Thus $h_{S^0} : \mathrm{GF}(2)^{m_1} \to \mathbb{R}$ is a well-defined function and we could compute its Walsh transform as follows,

$$\begin{aligned}
W_{h_{S^0}}(n_0, n_1, ..., n_{m_1-1}) &= \sum_{\mathbf{a} \in GF(2)^{m_1}} h_{S^0}(\mathbf{a})(-1)^{\mathbf{a}\cdot(n_0,n_1,...,n_{m_1-1})'} \\
&= \sum_{\mathbf{a} \in GF(2)^{m_1}} \left(\sum\nolimits_{t:\ \mathcal{U}_t=\mathbf{a}} (-1)^{\mathcal{Z}_t \oplus \mathcal{V}_t \oplus \mathcal{B}_t}\right)(-1)^{\mathbf{a}\cdot(n_0,n_1,...,n_{m_1-1})'} \\
&= \sum_{t=1}^{\Omega} (-1)^{(n_0,n_1,...,n_{m_1-1})\cdot \mathcal{U}_t \oplus \mathcal{Z}_t \oplus \mathcal{V}_t \oplus \mathcal{B}_t} \\
&= \Pi_0 - \Pi_1,
\end{aligned}$$

where Π_0 and Π_1 are the number of 0s and 1s respectively, for the value of $\mathcal{Z}_t \oplus \mathcal{V}_t \oplus \mathcal{B}_t \oplus (n_0, n_1, ..., n_{m_1-1}) \cdot \mathcal{U}_t$. From this we have

$$\sum_{t=1}^{\Omega} (\Delta(i_1^{(t)}, j_1^{(t)}, ..., i_\kappa^{(t)}, j_\kappa^{(t)}) \oplus 1) = \frac{\Omega + W_{h_{S^0}}(n_0, n_1, ..., n_{m_1-1})}{2}.$$

Hence we only need to compute the Walsh transform of h_{S^0} to get the 2^{m_1} values for the number of vanishing Δs, corresponding to the 2^{m_1} guesses of $(n_0, n_1, ..., n_{m_1-1})$. This can be done efficiently by FWT in $\Omega + m_1 2^{m_1}$ time with 2^{m_1} memory. Note that for each guess of S^0, we should execute the above process once. Thus, the time complexity is $2^{m'}(\Omega + m_1 2^{m_1})$, which is greatly reduced compared to the complexity $2^{m'}2^{m_1}\Omega$ of the straightforward method. From the above, we define the statistic $\mathcal{F}$ as

$$\mathcal{F}(S^0) = \max_{(n_0,n_1,...,n_{m_1-1})} W_{h_{S^0}}(n_0, n_1, ..., n_{m_1-1}).$$

Denote $\mathbf{s}_c$ as the correct guess for S^0 and $\mathbf{s}_w$ otherwise. According to the central limit theorem, we have

$$\begin{aligned}
\frac{(\Omega + \mathcal{F}(\mathbf{s}_c))/2 - \Omega(1+\epsilon_F)/2}{\sqrt{\Omega(1+\epsilon_F)(1-\epsilon_F)}/2} &= \frac{\mathcal{F}(\mathbf{s}_c) - \Omega\epsilon_F}{\sqrt{\Omega(1-\epsilon_F^2)}} \sim \mathcal{N}(0,1), \\
\frac{(\Omega + \mathcal{F}(\mathbf{s}_w))/2 - \Omega/2}{\sqrt{\Omega}/2} &= \frac{\mathcal{F}(\mathbf{s}_w)}{\sqrt{\Omega}} \sim \mathcal{N}(0,1),
\end{aligned}$$

where $\mathcal{N}(\cdot,\cdot)$ is the normal distribution with the specified expectation and variance. From this we get $\mathcal{F}(\mathbf{s}_c) - \mathcal{F}(\mathbf{s}_w) \sim \mathcal{N}\left(\Omega\epsilon_F, \Omega(2-\epsilon_F^2)\right)$. Let Φ be the cumulative function of the standard normal distribution, i.e., $\Phi(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-\frac{1}{2}t^2}dt$, thus the probability that a wrong guess $\mathbf{s}_w$ has a better rank than the correct $\mathbf{s}_c$, i.e., $\mathcal{F}(\mathbf{s}_c) < \mathcal{F}(\mathbf{s}_w)$, is approximately $\Phi\left(-\sqrt{\Omega\epsilon_F^2/(2-\epsilon_F^2)}\right)$. Thus to identify the correct candidate of the FSR initial state with a high probability, the number Ω of parity-checks should satisfy $\Omega \geq 4m' \ln 2/\epsilon_F^2$. To have a flexible attack, we set a threshold value T of $\mathcal{F}(S^0)$ when choosing the FSR initial state candidates, i.e., all the values that result in $\mathcal{F}(S^0) \geq T$ will be chosen as candidates, otherwise will be filtered out. Denote by α the probability that the correct guess will be chosen as a candidate, and by β the probability that a wrong guess would be chosen as a candidate, then

$$\begin{aligned}
\alpha &= \Pr(\mathcal{F}(\mathbf{s}_c) \geq T) = 1 - \Phi\left(\frac{T - \Omega\epsilon_F}{\sqrt{\Omega(1-\epsilon_F^2)}}\right), \\
\beta &= \Pr(\mathcal{F}(\mathbf{s}_w) \geq T) = 1 - \Phi\left(\frac{T}{\sqrt{\Omega}}\right) \triangleq 2^{a'}.
\end{aligned}$$

In cryptanalysis, we expect to choose a T such that α is very close to 1 to assure a high passing probability for the correct guess, meanwhile β is very small to filter out all the wrong guesses, or to reduce the passing number of wrong guesses as much as possible.

Fruit case. Now we demonstrate the above procedure on Fruit to obtain some LFSR initial state candidates with the focus on the relations of the parameters originating in the attack.

Given the exact values of the parameters ω and m_1, we have $D = 128(\omega - 1) + 1$, $\omega' = 7\omega$ and $\Omega = \omega'^2 \cdot 2^{-((37-m_1)+1)} = \omega^2/2^{(32.39-m_1)}$. To identify the correct candidate of the LFSR initial state with a high probability, the number of parity-checks of Eq.(8) should be at least $\Omega = 4 \times 43 \times \ln 2/\epsilon_F^2 = 2^{21.30}$. The time complexity of this procedure for Fruit is $C_4 = 2^{43}(\Omega + m_1 2^{m_1})$.

To illustrate the basic idea of this stage, we show in Fig.3 and Fig.4 the Walsh Spectrum of the function $h_{\mathbf{s}_c}$ for a correct guess $\mathbf{s}_c$ and the function $h_{\mathbf{s}_w}$ for a randomly generated wrong guess $\mathbf{s}_w$, respectively. Precisely, set $\omega = 2^{11.3}$ and $D = 128(\omega - 1) + 1 = 2^{18.3}$, we first used the RC4 cipher to randomly generate one (K, IV) pair for Fruit, and then run Fruit to generate the initial state (S^0, N^0) and the corresponding keystream bits $\{z_i\}_{i=0}^{D-1}$. Next, we proceed as follows: we first fixed the values of the last 29 bits of N^0. Then with the correct guess $\mathbf{s}_c = S^0$, we expressed each NFSR state variable n_{37+i}, $i = 0, 1, ..., D - 1$, as a linear combination of the unfixed variables $(n_0, n_1, ..., n_7)$ with the known keystream bits $z_0, ..., z_{D-1}$. By using the 7 linearly independent linear approximations for g with the largest bias $2^{-4.6}$, we can construct $\omega' = 7 \cdot \omega = 2^{14.1} (> \frac{4 \times 43 \times \ln 2}{(2 \times 2^{-4.6})^2})$ parity checks only containing the variables $(n_0, n_1, ..., n_7)$. The Walsh spectrum can be computed by FWT for all the patterns of $(n_0, n_1, ..., n_7)$. Further, we randomly generated a wrong guess for S^0 and derived the corresponding Walsh spectrum for all the patterns of $(n_0, n_1, ..., n_7)$ with a similar process. Fig. 3 and Fig. 4 show that there is a peak in the Walsh spectrum derived for the correct guess of S^0, while it keeps in a steady state for a randomly generated wrong guess of S^0.

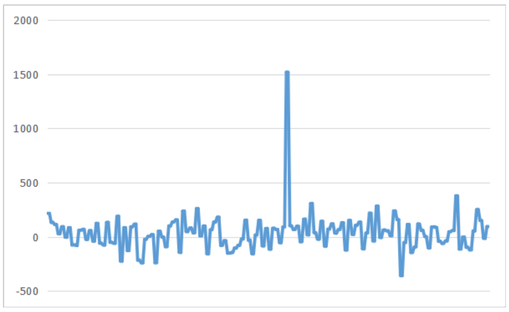

Figure 3: The Walsh Spectrum of the function $h_{\mathbf{s}_c}$ for the correct guess of S^0

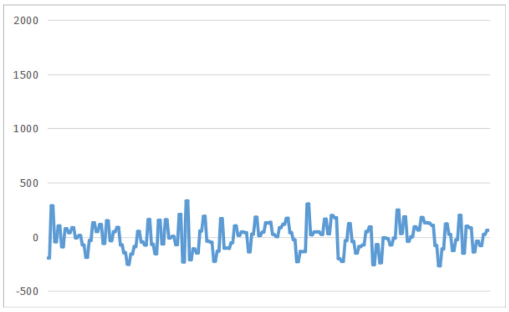

Figure 4: The Walsh Spectrum of the function $h_{\mathbf{s}_w}$ for a random wrong guess of S^0

The success probability of the attack depends essentially on the choice of the threshold T, which will be determined precisely according to various attack conditions in section 5.5. Naturally we expect to choose a T such that α is very close to 1, while β is very small, i.e., we let the correct candidate pass the statistical test with a high probability, while the wrong guesses will be reduced to a large extent.

5.3 Recovery of the NFSR Initial State

We first show how to recover the NFSR initial state of the model, and then focus on the concrete Fruit case.

We first recover the first m_1 bits of N^0, i.e., $(n_0, n_1, ..., n_{m_1-1})$. The target function is the sum $\sum_{t=1}^{\Omega_1}(\Delta(i_1^{(t)}, j_1^{(t)}, ..., i_\kappa^{(t)}, j_\kappa^{(t)}) \oplus 1)$, which will follow the binomial distribution $(\Omega_1, \frac{1}{2}(1+\epsilon_F))$ for the correct $(n_0, n_1, ..., n_{m_1-1})$, and otherwise this sum will have the binomial distribution $(\Omega_1, \frac{1}{2})$, where Ω_1 is the number of parity-checks needed for the recovery of $(n_0, n_1, ..., n_{m_1-1})$. To guarantee a high success rate, we set $\Omega_1 = 8m_1 \ln 2/\epsilon_F^2$, it is clear that the sum $\sum_{t=1}^{\Omega_1}(\Delta(i_1^{(t)}, j_1^{(t)}, ..., i_\kappa^{(t)}, j_\kappa^{(t)}) \oplus 1)$ should be maximum for the correct guess. We will use the distinguisher statistic $W_{h_{s_c}}(n_0, n_1, ..., n_{m_1-1})$ to characterize this property. Similarly we define an integer-valued function h_{s_c} by regrouping the Ω_1 parity-checks and compute the Walsh transform of $h_{\mathbf{s_c}}$ for all the possible guesses of $(n_0, n_1, ..., n_{m_1-1})$, then $W_{h_{s_c}}(\mathbf{n}_c) \sim \mathcal{N}(\Omega_1\epsilon_F, \Omega_1(1-{\epsilon_F}^2))$ and $W_{h_{s_c}}(\mathbf{n}_w) \sim \mathcal{N}(0, \Omega_1)$, where $\mathbf{n}_c$ denotes the correct guess and $\mathbf{n}_w$ otherwise. Accordingly, the statistic $W_{h_{s_c}}(n_0, n_1, ..., n_{m_1-1})$ should be maximum if $(n_0, n_1, ..., n_{m_1-1})$ is correct and the best candidate is

$$(n_0, n_1, ..., n_{m_1-1}) = \arg\max_{(n_0, n_1, ..., n_{m_1-1})} W_{h_{s_c}}(n_0, n_1, ..., n_{m_1-1}).$$

The time complexity of this procedure, conditioned on one candidate of the FSR initial state, lies in the calculation of the Walsh transform of $h_{\mathbf{s_c}}$ for all the 2^{m_1} possible values of the $(n_0, n_1, ..., n_{m_1-1})$, which is $\Omega_1 + m_1 2^{m_1}$.

Once the first m_1 bits of the NFSR initial state is recovered, we enter the next pass to determine the next m_2 bits of N^0 with a similar procedure conditioned on the recovered information, which will have a complexity much lower than the first step. Thus we can finally determine the whole NFSR initial state.

Fruit case. For Fruit, we set $\Omega_1 = 8m_1 \ln 2/\epsilon_f^2$. After we obtain the candidates of the LFSR initial state in section 5.2, we proceed to determine the first m_1-bit of the NFSR initial state following a similar method as above, conditioned on the LFSR state candidates and the available keystream. For one LFSR initial state candidate, the time complexity is computed as $C_5 = \Omega_1 + m_1 2^{m_1}$. After the recovery of $(n_0, n_1, ..., n_{m_1-1})$, the remaining $37 - m_1$ bits can be recovered with a similar method and a small-scale exhaustive search, conditioned on the state candidates of the LFSR initial state S^0, $(n_0, n_1, ..., n_{m_1-1})$, and the keystream, and the time complexity for one set of candidate is $C_6 = 2^{37-m_1} \cdot \frac{4}{(2\epsilon)^2} = 2^{37-m_1} \cdot \frac{1}{\epsilon^2}$.

5.4 Recovery of the Secret Information Bits Within one Cycle

Algorithm 2

```
Input:  a state candidate $(S^0, N^0)$.
Output: a flag representing the correctness of the state candidate,
        and output $k'_i \oplus c_i,\ i = 0, 1, ..., d-1$, for the correct one.
Create a $d$-bit vector $\zeta$;
for $i = 0, 1, ..., d-1$ do
    compute $n_{m+i}$ from $z_{m-\eta_{r_2}}, z_{m-\eta_{r_2}+1}, ..., z_{m-\eta_{r_2}+i}$ with the technique
     described in section 4.1;
    compute $k'_i \oplus c_i = n_{m+i} \oplus lin(S^i) \oplus g(n_i, n_{1+i}, ..., n_{m-1+i})$;
    store $k'_i \oplus c_i$ at the $i$-th position of the vector $\zeta$, i.e., $\zeta[i] = k'_i \oplus c_i$.
for $i = 0, 1, ..., d-1$ do
    compute $n_{m+d+i}$ from $z_{m+d-\eta_{r_2}}, z_{m+d-\eta_{r_2}+1}, ..., z_{m-\eta_{r_2}+d+i}$;
    compute $v_i \triangleq n_{m+d+i} \oplus lin(S^{d+i}) \oplus g(n_{d+i}, n_{1+d+i}, ..., n_{m-1+d+i})$;
    if $v_i = \zeta[i]$ then continue for next $i$;
    else output a flag that the state candidate is wrong and stop.
if $v_i = \zeta[i]$ for all $i = 0, 1, ..., d-1$
then output a flag that the state candidate is correct,
     and output the $d$ secret information bits, i.e., $\zeta[i]$, $i = 0, 1, ..., d-1$.
```

After identifying the candidates of the FSR initial state S^0 and the NFSR initial state N^0, we will carry out the Algorithm 2 to check whether a state candidate is correct, and if so, to further restore d consecutive secret information bits $k'_i \oplus c_i$ $(i = 0, 1, ..., d-1)$ within one cycle. For any state candidate, the average number of ticks for state checking is $d + (1 \cdot \frac{1}{2^0} + 2 \cdot \frac{1}{2} + 3 \cdot \frac{1}{2^2} + ... + d \cdot \frac{1}{2^{d-1}}) \approx d + 4$.

For Fruit, we have $m = 37$, $\eta_{r_2} = 36$, $m - \eta_{r_2} = 1$ and $d = 128$. Plugging them into Alg.2, we obtain the corresponding algorithm for recovering the 128 round key bits of Fruit, by combining the fact that the counter bits c_t^{10} are deterministic at any time t. The average number of ticks for state checking is 132.

5.5 Complexity Analysis

In the process of restoring the FSR initial state, a threshold T is introduced which can provide large flexibility when actually constructing a fast correlation attack. According to the value of T, the two probabilities α and $\beta = 2^{-a'}$ are computed, indicating the probabilities that the correct guess and a random wrong guess could be chosen as an initial state candidate of the FSR, respectively. The following four cases could be encountered according to the different values of α and $\beta = 2^{-a'}$.

I. $\alpha > 0.99$ and $\beta < 2^{-m'}$ (i.e., $a' > m'$). This means that the correct guess of the m'-bit FSR initial state will almost certainly be chosen as a state candidate, while none of the wrong ones could be chosen. In this case, we have only one state candidate of the FSR, thus it seems no need for the state checking step.

II. $\alpha > 0.99$ and $\beta > 2^{-m'}$ (i.e., $a' < m'$). This means that the correct guess of the FSR initial state will be chosen as a state candidate with a high probability, together with some wrong guesses. In other words, we will have to deal with some state candidates of the FSR and check the correctness for each of them. In this case, we need to carry out the state checking process.

III. $\alpha < 0.99$ and $\beta < 2^{-m'}$ (i.e., $a' > m'$). This means that the correct guess might not be chosen as a state candidate, but on the good side, none of the wrong guesses would be chosen. In this case, we might obtain no candidate, and thus have to repeat the whole attack process several times, denoted by λ, to guarantee a high success rate.

IV. $\alpha < 0.99$ and $\beta > 2^{-m'}$ (i.e., $a' < m'$). This means that some FSR initial state candidates will be obtained, among which the correct one might not exist. In this case, we will finally have to deal with some state candidates by checking the correctness for each of them, and if necessary, repeat the whole attack several times, denoted by λ, to guarantee a high success rate.

We have the following theorem on the various complexity aspects of the 2nd phase of Algorithm 1 on the generic model.

Theorem 1. *Let m_i $(i \geq 1)$ be the length of the divided pieces of the NFSR such that $\sum_i m_i \leq m$ and κ the weight of the constructed parity-checks to restore the first m_1 bits, then the data complexity is $D = \frac{d(\kappa! 2^{m-m_1})^{1/\kappa}}{R} \cdot (\frac{4m' \ln 2}{2^{2\kappa} \epsilon^{2\kappa}})^{1/\kappa}$, and the time complexity C is listed according to the above four cases:*

I. $2^{m'}(\Omega + m_1 2^{m_1}) + \sum(\Omega_i + m_i 2^{m_i}) + 2^{(m - \sum m_i)}/\epsilon^2$,

II. $2^{m'}(\Omega + m_1 2^{m_1}) + 2^{m'-a'} \left(\sum(\Omega_i + m_i 2^{m_i}) + 2^{(m - \sum m_i)}/\epsilon^2 + (d+4)\right)$,

III. $\lambda(2^{m'}(\Omega + m_1 2^{m_1}) + \sum(\Omega_i + m_i 2^{m_i}) + 2^{(m - \sum m_i)}/\epsilon^2)$,

IV. $\lambda(2^{m'}(\Omega + m_1 2^{m_1}) + 2^{m'-a'} (\sum(\Omega_i + m_i 2^{m_i}) + 2^{(m - \sum m_i)}/\epsilon^2 + (d+4)))$,

where $\beta = 2^{-a'}$ and Ω_i is the number of parity-checks utilized to restore the m_i $(i \geq 2)$ bits in the multi-pass phase.

Proof. We focus on the case **I**, other cases could be treated in a similar way.

For the data complexity, from $D = d(\omega - 1) + 1$ keystream bits, $\Omega = \frac{\binom{\omega'}{\kappa}}{2^{m-m_1}}$ and $\omega' = \omega \cdot R$, we have

$$\Omega := \frac{{\omega'}^{\kappa}}{\kappa! 2^{m-m_1}} = \frac{(\omega R)^{\kappa}}{\kappa! 2^{m-m_1}}.$$

On the other hand, as illustrated in section 5.2 and from the classical reasoning in correlation attacks [7, 18], to identify the correct candidate of the FSR initial state with a high probability, Ω is usually chosen to be $\Omega \geq \frac{(2-\epsilon_F^2)\cdot 2\cdot m'\cdot \ln 2}{\epsilon_F^2} \approx \frac{4\cdot m'\cdot \ln 2}{\epsilon_F^2}$. Thus we could safely set $\Omega = 4m' \ln 2/\epsilon_F^2$ with $\epsilon_F = 2^{\kappa}\epsilon^{\kappa}$. Accordingly, D is computed as

$$D := d \cdot \omega = d \cdot \frac{(\kappa \cdot 2^{m-m_1})^{\frac{1}{\kappa}}}{R} \cdot \Omega^{\frac{1}{\kappa}} = \frac{d(k! 2^{m-m_1})^{1/k}}{R} \cdot (\frac{4m' \ln 2}{\epsilon_F^2})^{1/k}.$$

For the time complexity in the case **I**, we first try all the possible values of the FSR initial state S^0. Based on the parity-checks of Eq.(7), we evaluate the distribution of $(n_0, n_1, ..., n_{m_1-1}) \cdot \mathcal{U}_t \oplus \mathcal{Z}_t \oplus \mathcal{V}_t \oplus \mathcal{B}_t$, $t = 1, 2, ..., \Omega$ for each possible value of S^0, by searching over the $(n_0, n_1, ..., n_{m_1-1})$ of the NFSR initial state, and record those possible FSR initial state values passing the statistical test in Section 5.2 as the candidates. It is expected that for the correct value of the FSR initial state S^0, there will be a peak in the Walsh spectrum distribution, otherwise the Walsh spectrum should have some uniform distribution, as depicted in Figures 3 and 4. The complexity cost during this process is counted precisely as follows. The preparation of $h_{S^0}(\mathbf{a})$ in Section 5.2 will take a complexity of Ω, while its FWT has a complexity of $m_1 2^{m_1}$. Recall that this routine has to be repeated for each guess of S^0, which gives $2^{m'}(\Omega + m_1 2^{m_1})$; the left summand is for the determination of other parts of the NFSR state in the multi-pass phase in a similar way as the first m_1 bits and the final correlation check. □

Though the weight κ of the parity-checks is not restricted in Theorem 1, it is expected that κ takes some small values such as 2 or 4 in the real attack. For the general value of κ, the desirable parity-checks could be constructed through the match-and-sort approach in [8] and the k-tree algorithm in [27].

Based on the theoretical framework established in this section, we have the following design criteria on Grain-like small state stream ciphers modelled in Fig.2.

1. The pseudo-linearity of the output function when combining the input variables should be avoided.

2. For l-bit security, there should exist no linear approximation with the bias ϵ for the state updating function g of the NFSR such that the resultant $D < 2^l$ and $C < 2^l$, where ϵ, D, C are the same notations as in Theorem 1.

Note that in [3], it has been realized that functions involved in Grain-like designs should have some correlation immunity. A design recommendation in [3] is to replace the feedback function g of the NFSR in Grain v0 by a 2-resilient function, which refers to correlation immunity rather than the nonlinearity (and there is a well-known tradeoff between them). We stress that in the version of Fruit as broken here, the feedback function of the NFSR follows this recommendation and is indeed 2-resilient, which clearly validates the necessity of the second design criterion. In addition, the second criteria actually depends on the concrete evaluation of the data and time complexities depicted in Theorem 1 for the primitive, which quantifies the amount of nonlinearity needed for a desirable security level by the complexities of a fast correlation attack.

Further, a new aspect here is that even in the presence of round keys and unknown counters, such correlations could be effectively exploited. In the following, we will analyze the complexities of the attack on Fruit by presenting the exact values of the involved parameters. Before this, we first illustrate how to restore the 80-bit secret key of Fruit from the 128 consecutive round key bits.

After we obtained the LFSR and the NFSR initial states in Fruit and the 128 consecutive round key bits $(k'_0, k'_1, ..., k'_{127})$, we continue to recover the 80-bit secret key as follows. Note that in Fruit, each round key bit is generated by combining 6 bits of the secret key, which is dependent on the value of C_r. As described in Section 2, C_r is only known in the first step (130 rounds) of the initialization. After the first 130 rounds, the initialization enters the second step (80 rounds) where C_r is fed from the LFSR and NFSR, thus is unknown anymore. Since we have known the initial state (S^0, N^0) and the round key bits in one cycle, we will proceed as follows. We run the inverse process of the second step of the initialization for 80 rounds and derive the internal state of the LFSR and the NFSR at time $t = -80$. As we know, all bits of C_r at time $t = -80$ equal to the corresponding LSBs of the NFSR except the last bit, accordingly we have only two guesses for the number of C_r. Recall that the round key bit k'_t is generated as

$$k'_t = k_{sv}k_{y+64} \oplus k_p k_{u+72} \oplus k_{q+32} \oplus k_{r+64},$$

where the indices sv, y, p, u, q, r are derived from C_r as $sv = c_t^0c_t^1c_t^2c_t^3c_t^4c_t^5$, $y = c_t^3c_t^4c_t^5$, $u = c_t^4c_t^5c_t^6$, $p = c_t^0c_t^1c_t^2c_t^3c_t^4$, $q = c_t^1c_t^2c_t^3c_t^4c_t^5$ and $r = c_t^3c_t^4c_t^5c_t^6$. Thus we obtain two systems of equations corresponding to the two guesses of the number of C_r with known 128 consecutive round key bits. Note that by guessing the values of $(k_{64}, k_{65}, ..., k_{79})$, we could linearize the round key function at the corresponding time instants and come up with 2^{16} systems with 128 linear equations and 64 unknowns, i.e., $(k_0, k_1, ..., k_{63})$. These systems can be solved by the Gauss elimination within at most $2^{16} \times 128^3 = 2^{37}$ basic operations. For the two guesses of the number of C_r, the complexity is at most 2^{38}.

As previously stated, we need to prepare the desirable parity-check relations before mounting the fast correlation attack on Fruit. As discussed in section 4.2.2, this can be finished in time $C_1 + C_2 + C_3$, where $C_1 = 2^{48.21} \cdot D$, $C_2 = 2^{52.12} \cdot \omega$ and $C_3 = 2^{43} \cdot (\omega' + \Omega) = 2^{43} \cdot (7\omega + 2^{21.30})$. Corresponding to the four cases at the beginning of section 5.5, for the concrete Fruit case, the following four cases are listed when analyzing the time complexity, which can be applied according to the different conditions and requirements of the attack.

I. $\alpha > 0.99$ and $\beta < 2^{-43}$ (i.e., $a' > 43$). The time complexity is $C_4 + C_5 + C_6 + 2^{38}$, where $C_4 = 2^{43}(2^{21.30} + m_1 2^{m_1})$, $C_5 = \Omega_1 + m_1 2^{m_1}$ and $C_6 = 2^{m_2} \cdot \frac{1}{\epsilon^2}$.

II. $\alpha > 0.99$ and $\beta > 2^{-43}$ (i.e., $a' < 43$). The time complexity is $C_4 + 2^{43-a'}(C_5 + C_6 + 132) + 2^{38}$.

III. $\alpha < 0.99$ and $\beta < 2^{-43}$ (i.e., $a' > 43$). The time complexity is $\lambda(C_4 + C_5 + C_6) + 2^{38}$.

IV. $\alpha < 0.99$ and $\beta > 2^{-43}$ (i.e., $a' < 43$). The time complexity is $\lambda(C_4 + 2^{43-a'}(C_5 + C_6 + 132)) + 2^{38}$.

Based on this classification, we list two sets of parameters. The first set of parameters are chosen as follows. Set $m_1 = 21$, i.e., we divide the NFSR into two parts of length 21 bits and $37 - 21 = 16$ bits, respectively. Let $\omega = 2^{16.35}$ and $D = 128(\omega - 1) + 1 = 2^{23.35}$. By using the 7 best linear approximations for g, we can construct $\omega' = 7 \cdot \omega = 2^{19.16}$ parity checks containing the full NFSR initial state variables, from which we can construct another $\Omega = \omega'^2 \cdot 2^{-(16+1)} = 2^{21.32} (> \frac{4\times43\times\ln 2}{(2^{-7.2})^2})$ parity checks containing only the first 21 variables of the NFSR initial state. We set a threshold $T = 2^{13.45}$ when recovering the LFSR initial state. In this case, we have $\alpha = 0.999978$ (> 0.99) and $\beta = 2^{-38.65}$ $(> 2^{-43})$, which accords with the above case **II**. The number of parity-checks needed for recovering the first 21-bit of the NFSR initial state is computed as $\Omega_1 = 8m_1 \ln 2/\epsilon_F^2 = 2^{21.26}$. Finally, the time complexity is $C \triangleq C_1 + C_2 + C_3 + C_4 + 2^{43-a'}(C_5 + C_6 + 132) + 2^{38} = 2^{71.56}$, equivalent to $\frac{2^{71.56}}{210+4} = 2^{63.82}$ Fruit encryptions. We also have another set of parameters:

$m_1 = 23$, $m_2 = 14$, $\omega = 2^{15.34}$, $D = 2^{22.34}$, $\omega' = 2^{18.15}$, $\Omega = 2^{21.30}(> \frac{4\times 43\times \ln 2}{(2^{-7.2})^2})$, $T = 2^{13.45}$, $\alpha = 0.999961$ (> 0.99) and $\beta = 2^{-39.20}$ ($> 2^{-43}$). With these parameters, the time complexity for recovering the 80-bit secret key of Fruit is $2^{70.55}$, equivalent to $2^{62.81}$ Fruit encryptions.

6 The Experimental Results

To validate the theoretical analysis of our attack, we have made practical experiments on a reduced version of Fruit. Similarly there are five parts involved: a 19-bit LFSR whose state at time t is denoted by $S^t = (s_t, s_{t+1}, ..., s_{t+18})$, a linked 18-bit NFSR whose state at time t is denoted by $N^t = (n_t, n_{t+1}, ..., n_{t+17})$, a 37-bit fixed key register, and two counter registers: a 6-bit counter $C_r = (c_t^0, ..., c_t^5)$ and a 7-bit counter $C_c = (c_t^6, ..., c_t^{12})$, allocated for the round key function and for the initialization/keystream generation, respectively. The 19-bit LFSR is updated independently and recursively as $s_{t+19} = s_t \oplus s_{t+3} \oplus s_{t+7} \oplus s_{t+17}$. The 18-bit NFSR is updated recursively by a non-linear feedback function g defined as $n_{t+18} = k'_t \oplus s_t \oplus c_t^9 \oplus g(N^t)$, where $g(N^t) = n_t \oplus n_{t+5} \oplus n_{t+10} \oplus n_{t+12}n_{t+3} \oplus n_{t+2}n_{t+13}n_{t+15}$, and c_t^9 is the 3-th LSB of the counter C_c. Define the values of sv, y, u, p, q, r from C_r as $sv = c_t^0c_t^1c_t^2c_t^3c_t^4$, $y = c_t^2c_t^3c_t^4$, $u = c_t^3c_t^4c_t^5$, $p = c_t^0c_t^1c_t^2c_t^3$, $q = c_t^1c_t^2c_t^3c_t^4$ and $r = c_t^2c_t^3c_t^4c_t^5$, then the round key bit k'_t is generated as $k'_t = k_{sv}k_{y+30} \oplus k_pk_{u+34} \oplus k_{q+15} \oplus k_{r+30}$. Given the internal state at time t, the filter function h produces $h_t = n_{t+1}s_{t+15} \oplus s_{t+1}s_{t+2} \oplus n_{t+15}s_{t+7} \oplus n_{t+13}s_{t+11} \oplus s_{t+6}s_{t+13}s_{t+18}$, and the keystream bit is generated as $z_t = h_t \oplus s_{t+16} \oplus n_t \oplus n_{t+4} \oplus n_{t+7} \oplus n_{t+10} \oplus n_{t+12} \oplus n_{t+17}$.

Our fast correlation attacks on the reduced version of Fruit have been fully implemented in C language on one core of a single PC, running with Windows 7, Intel Core i3-2120 CPU @ 3.30 GHz and 4.00GB RAM. In general, the experimental results match the theoretical analysis quite well.

There are 3 linearly independent linear approximations for g having the largest bias of $2^{-2.4}$, i.e., $n_t \oplus n_{t+5} \oplus n_{t+10}$, $n_t \oplus n_{t+5} \oplus n_{t+10} \oplus n_{t+12}$ and $n_t \oplus n_{t+5} \oplus n_{t+10} \oplus n_{t+3}$. In our experiments, we first verified the validity of the recovery of the LFSR initial state. To achieve this, we need to carry out a sort-and-merge procedure in theory to reduce the effective length of the NFSR initial state. Actually we have done some experiments on this issue which confirmed the fact that for x randomly generated vectors, there are an expected number of $x^2/2^{y+1}$ pairs which collide on some specified y positions. Thus we ran the experiments as follows. Set $\omega = 2^{6.94}$. Note that the round key function has a cycle of length 64, we first generated $D = 64(\omega - 1) + 1 = 2^{12.93}$ keystream bits from an initial state with the last 10 bits $(n_8, n_9, ..., n_{17})$ of the NFSR initial state being fixed and the 19 bits of LFSR initial state $(s_0, s_1, ..., s_{18})$ and the first 8 bits $(n_0, n_1, ..., n_7)$ of the NFSR initial state being randomly generated, using the RC4 cipher. Next, we exhaustively search over all the 2^{19} guesses of the LFSR initial state, and for each guess, we express each NFSR state variable n_{18+i}, $i = 0, 1, ..., D - 1$, as a linear combination of the variables $(n_0, n_1, ..., n_7)$ with the known keystream bits following the induction method in section 4.1. Thus, for each guess of the LFSR initial state, we constructed $3 \cdot \omega = 2^{8.52}(> \frac{4\times 19\times \ln 2}{(2\times 2^{-2.4})^2})$ parity checks only containing the variables $(n_0, n_1, ..., n_7)$. Further, for each possible 19-bit LFSR initial state, we applied FWT to evaluate the parity check equations for all the patterns of $(n_0, n_1, ..., n_7)$, and recorded the maximum value of Walsh transform denoted by $\mathcal{S}(s'_0, s'_1, ..., s'_{18})$. We set a threshold $T = 2^{6.3}$ and the guesses $(s'_0, s'_1, ..., s'_{18})$ are chosen as the LFSR initial state candidates when $\mathcal{S}(s'_0, s'_1, ..., s'_{18}) > 2^{6.3}$. The probability that the correct guess and the wrong guess is chosen as a candidate is $0.9996(> 0.99)$ and $2^{-15.65}$, respectively. Next, we proceed to recover the NFSR state using the obtained LFSR initial state candidates. For each candidate, we evaluate the parity check equations for all the patterns of $(n_0, n_1, ..., n_7)$, and the one with the maximum value of Walsh transform is chosen as the NFSR initial state candidate. On average, we obtained

10 state candidates for $(s_0, s_1, ..., s_{18})$ and $(n_0, n_1, ..., n_7)$. For each of these candidates, we checked the validity of the state and the recovery of the 64 consecutive round key bits by a method similar to Algorithm 2. The estimated time complexity of the above process is $2^{30.23}$. In the simulation, we finally identified the round key bits in a cycle within a few hours. From the round key function, the round key bit k'_t is generated as $k'_t = k_{sv}k_{y+30} \oplus k_p k_{u+34} \oplus k_{q+15} \oplus k_{r+30}$. By guessing the values of $(k_{30}, k_{31}, ..., k_{36})$, we come up with 2^7 systems with 64 linear equations and 30 unknowns, i.e., $(k_0, k_1, ..., k_{29})$. These systems can be solved using a non-optimized method with at most $2^7 \times 64^3 = 2^{25}$ basic operations, which could be safely ignored when comparing with the complexity in the previous steps.

7 Conclusions

In this paper, we have studied the security of Grain-like small state stream ciphers by fast correlation attacks, the classical cryptanalytic method against LFSR-based stream ciphers. A generalized model of such primitives is defined and a formal framework for fast correlation attacks utilizing the divide-and-conquer strategy on the model is presented with a thorough theoretical analysis. It is shown that if the non-linear combining function has some pseudo-linear property when combining the input variables from the cascaded internal state, then such an attack would be applicable in principle. This results in two general design criteria for such small state stream ciphers to achieve the desirable security. Both do hold irrespective of the specifics of round key generation. One is that the pseudo-linearity of the output function when combining the input variables should be *strictly* avoided; the other is to prevent the good linear approximation of the NFSR state updating function. As an application, we broke Fruit, a tweaked version of Sprout, in $2^{62.8}$ Fruit encryptions, given $2^{22.3}$ keystream bits for all the keys, which clearly violates the 80-bit security claim. Our results have been verified in experiments on a small-scale version of Fruit. Our attack becomes inefficient for Grain v1 because the length of the LFSR is already 80-bit and is not applicable to Plantlet and Lizard so far for the lack of the pseudo-linearity of the output functions in both. We believe that our work will be helpful in understanding the security of such small state primitives and useful for the upcoming designs.

Acknowledgements

We would like to thank the anonymous reviewers for very helpful comments. This work is supported by the National Key R&D Research program of China (Grant No. 2017YFB0802504), the program of the National Natural Science Foundation of China (Grant No. 61572482), National Cryptography Development Fund (Grant No. MMJJ20170107) and National Grand Fundamental Research 973 Programs of China (Grant No. 2013CB338002).

References

[1] Armknecht F. and Mikhalev V., On lightweight stream ciphers with shorter internal states, *Fast Software Encryption–FSE'2015*, LNCS vol. 9054, pp. 451-470, 2015.

[2] Banik S., Some results on sprout, *Progress in Cryptology–INDOCRYPT 2015*, LNCS vol. 9462, pp. 124-139, 2015.

[3] Berbain, C., Gilbert, H. and Maximov, A., Cryptanalysis of Grain, In M.J.B. Robshaw (Ed.), *Fast Software Encryption–FSE'2006*, LNCS vol. 4047, Springer-Verlag, pp. 15-29.

[4] Berbain, C., Gilbert, H. and Joux, A., Algebraic and correlation attacks against linearly filtered non linear feedback shift registers, In Avanzi R. M., Keliher L. and Sica F. (eds), *Selected Areas in Cryptography–SAC 2008*, LNCS vol. 5381, Springer-Verlag, pp. 184-198.

[5] Biryukov A. and Shamir A., Cryptanalytic time/memory/data tradeoffs for stream ciphers, In Okamoto T. (eds), *Advances in Cryptology–ASIACRYPT 2000*, LNCS vol. 1976, Springer-Verlag, pp. 1-13. 2008.

[6] Canteaut A. and Trabbia. M., Improved fast correlation attacks using parity-check equations of weight 4 and 5. In Preneel B. (eds), *Advances in Cryptology–EUROCRYPT 2000*, LNCS vol. 1807, pp. 573–588, Springer Berlin Heidelberg, 2000.

[7] Chepyzhov V. V., Johansson T. and Smeets B., A simple algorithm for fast correlation attacks on stream ciphers, In Goos G. Hartmanis J. van Leeuwen J. and Schneier B. (eds), *Fast Software Encryption–FSE'2000*, LNCS vol.1978, Springer-Verlag, pp. 181–195, 2000

[8] Chose P., Joux A. and Mitton M., Fast correlation attacks: an algorithmic point of view. In Knudsen L. R. (eds), *Advances in Cryptology–EUROCRYPT 2002*. LNCS vol. 2332, Springer Berlin Heidelberg, pp. 209-221, 2002.

[9] Courtois N. T., Weier. W., Algebraic attacks on stream ciphers with linear feedback, In Biham E. (eds), *Advances in Cryptology–EUROCRYPT'2003*, LNCS vol. 2656, Springer-Verlag, pp. 345–359, 2003.

[10] Courtois N. T., Fast algebraic attacks on stream ciphers with linear feedback, In Boneh D. (eds), *Advances in Cryptology–CRYPTO'2003*, LNCS vol. 2729, Springer-Verlag, pp. 176–194, 2003.

[11] Dey S. and Sarkar S., Cryptanalysis of full round Fruit, available at `https://eprint.iacr.org/2017/087.pdf`

[12] Esgin M. F. and Kara O., Practical cryptanalysis of full Sprout with TMD tradeoff attacks, *Selected Areas in Cryptography–SAC 2015*, LNCS vol. 9566, pp. 67-85, 2015.

[13] Ghafari V. A., Hu H. and Chen Y., Fruit: ultra-lightweight stream cipher with shorter internal state, available at `https://eprint.iacr.org/2016/355.pdf.`

[14] Golić J. D., On the security of nonlinear filter generators, In Gollmann D. (eds), *Fast Software Encryption–FSE 1996*. LNCS vol. 1039, pp. 173-188, 1996.

[15] Hamann M., Krause M. and Meier W., LIZARD - A lightweight stream cipher for power-constrained devices. *IACR Transactions on Symmetric Cryptology*, ISSN 2519-173X, vol. 2017, No. 1, pp. 45-79, `DOI:10.13154/tosc.v2017.i1.45-79`

[16] Hamann M., Krause M., Meier W., and Zhang B., Time-Memory-Data tradeoff attacks against small-state stream ciphers, available at `https://eprint.iacr.org/2017/384.pdf.`

[17] Hell M., Johansson T. and Meier W., Grain - a stream cipher for constrained environments, *International Journal of Wireless and Mobile Computing*, 2(1), pp. 86-93, 2007.

[18] Lu, Y. and Serge, V.: Faster correlation attack on Bluetooth keystream generator E0, in M. Franklin ed., *Advances in Cryptology–CRYPTO'2004*, LNCS vol. 3152, Springer-Verlag, pp. 407–425, (2004).

[19] Johansson T. and Jönsson F., Improved fast correlation attacks on stream ciphers via convolutional codes, In Stern J. (eds), editor, *Advances in Cryptology–EUROCRYPT'99*, LNCS vol. 1592, pp. 347–362, Springer Berlin / Heidelberg, 1999.

[20] Johansson T. and Jönsson F., Fast correlation attacks through reconstruction of linear polynomials, In Bellare M. (eds), *Advances in Cryptology–CRYPTO 2000*, LNCS vol. 1880, pp. 300-315, 2000.

[21] Lallemand V. and Naya-Plasencia M., Cryptanalysis of full Sprout, *Advances in Cryptology–CRYPTO 2015*, LNCS vol. 9215, pp. 663-682, 2015.

[22] Maitra S., Sarkar S., Baksi A. and Dey P., Key recovery from state information of Sprout: application to cryptanalysis and fault attack, available at `http://eprint.iacr.org/2015/236.pdf`

[23] Matsui M., Linear cryptanalysis method for DES cipher, *Advances in Cryptology–EUROCRYPT'93*, LNCS vol. 765, pp. 386-397, 1993.

[24] Meier W. and Staffelbach O. J., Fast correlation attacks on certain stream ciphers, *Journal of Cryptology*, vol. 1, issue 3, pp. 159-176, 1989.

[25] Mikhalev V., Armknecht F. and Müller C., On ciphers that continuously access the non-volatile key, *IACR Transactions on Symmetric Cryptology*, ISSN 2519-173X, vol. 2016, No. 2, pp. 52–79, `DOI:10.13154/tosc.v2016.i2.52-79.`

[26] Yarlagadda R. K. and Hershey J. E., Hadamard matrix analysis and synthesis with applications to communications and signal/Image Processing, pp. 17-22, Kluwer Academic, Dordrecht, 1997.

[27] Wagner. D., A generalized birthday problem. In Yung., M. (eds), *Advances in Cryptology–CRYPTO 2002*, LNCS vol. 2442, pp. 288–304. Springer Berlin Heidelberg, 2002.

[28] Zhang B. and Feng D., Multi-pass fast correlation attack on stream ciphers, *Selected Areas in Cryptography-SAC 2006*, LNCS vol. 4356, pp. 234-248, 2007.

[29] Zhang B. and Gong X., Another tradeoff attack on Sprout-like stream ciphers, *Advances in Cryptology-ASIACRYPT 2015*, LNCS vol. 9453, pp. 561-585, 2015.