# Multiple Sampling Fast Correlation Attack on Small State Stream Ciphers with Limited Round Key Period

Zhongzhi Zhou, Vahid Amin-Ghafari*, Hui Liu
* Corresponding author
E-mail addresses: Vahidaming@cumt.edu.cn

**Abstract**: The fast correlation attack (FCA) is a powerful cryptanalysis technique that targets stream ciphers based on linear feedback shift registers (LFSRs). Several FCAs were applied to small state stream ciphers (SSCs). In this paper, the idea of multiple sampling is proposed to use the available keystream bits more efficiently and decrease the data complexity of the attacks. This idea helps to overcome the limitation of SSCs on the number of output keystream bits. Moreover, we classify the parity check equations obtained from the different sampling rounds into different groups to ensure that the round keys used in these equations are the same. Our attack is applied to the Fruit-80 and reduces the data complexity from $2^{56.82}$ to $2^{49.82}$. This modified FCA can be applied to all SSCs with limited round key periods. Finally, we suggest a new design idea to strengthen SSCs against FCAs.

**Keywords:** Multiple sampling; Fast correlation attack; Small state stream cipher; Round key period; Parity check equation.

## 1 Introduction

Stream ciphers are an essential class of cryptography that operate as symmetric key encryption algorithms. The generation of keystream bits usually includes initialization phase and keystreams generation phase (Halevi et al., 2002). During the initialization phase, the key and the initialization vector (IV) are combined to generate the initial states of the cipher. In the keystreams generation phase, the initial states are constantly updated to produce keystream bits for encryption.

Mickey (Babbage et al., 2006), Trivium (Canniere, 2006), and the Grain family (Hell et al., 2006; Hell et al., 2007; Gren et al., 2011) are famous stream ciphers, which have internal states at least twice the security level to guarantee security against time-memory-data trade-off (TMDTO) attacks (Hellman et al., 1980). However, this rule inevitably increases the occupied area of the cipher hardware, which is unsuitable for resource-constrained environments such as IoT and RFID. The pursuit of designing stream ciphers with smaller internal states and resistance to TMDTO attacks has become a popular research direction. Sprout is recognized as the pioneer in SSCs and was designed to resist TMDTO attacks (Armknecht et al., 2015). Compared to the previous stream ciphers (i.e., conventional stream ciphers), it has a smaller internal state and introduced an additional function known as the round key function. This function generates the round keys to update the internal state of the nonlinear feedback shift register (NFSR) during the keystreams generation phase. Regrettably, Sprout exhibited inadequate resilience against TMDTO attacks, leaving it vulnerable to potential security breaches (Lallemand et al., 2015; Esgin et al., 2016; Kara et al., 2018). Nevertheless, it

laid the foundation for the development of other SSCs. Many SSCs, such as Plantlet (Mikhalev et al., 2016), Fruit-80 (Amin et al., 2018), and Atom (Banik et al, 2021), have been designed based on this idea and added some new functions to increase the security of ciphers. Due to their security and lower hardware requirements, SSCs have emerged as a leading research focus in cryptography.

Siegenthaler et al.(1984) proposed the correlation attack, an important cryptanalysis method targeting stream ciphers utilizing LFSRs. This approach leverages the correlation between the LFSR's internal states and the keystream sequence to validate guessed initial states. Correct guesses yield a significant bias, whereas incorrect guesses yield seemingly random bias statistics. While exhaustive searching of all cipher internal states incurs greater time complexity than correlation attacks, Fast Correlation Attacks (FCAs) have been devised to mitigate this. Various methods have been used to reduce the number of variables in the parity check equations and reduce the time complexity of FCAs. For example, Chose et al. (2002) treated the guessing and evaluation process of the FCAs as the Walsh Hadamard transformation. They used the fast Walsh Hadamard transformation to accelerate the FCAs process and reduce the time complexity of the attacks. Zhang et al. (2006) omitted to guess some internal state variables of LFSR by employing XOR operations on two different parity check equations. Todo et al. (2018) discovered the "commutative" property of multiplying a matrix with a vector in finite fields and used it to construct new parity check equations. When the cipher has multiple linear masks with high biases, a new false initial state assumption is proposed to identify multiple internal states with high biases. Considering the inverse matrices generated by the linear masks and the statistical distribution of the internal states, the correct internal state can be recovered without checking all internal states of the cipher.

Since round key functions are used to update the internal states of SSCs continuously, the round key bits are inevitably involved in parity check equations, considerably increasing the difficulty of the attacks. To solve this problem, Wang et al. (2019) took the period of the round key function as the sampling interval and obtained parity check equations with the same round key. They discovered that the round keys only affect the biases' direction without changing their absolute value. This method helps eliminate the influence of the round keys on the bias of statistics and resolve the bottleneck in attacks on SSCs.

However, as the round key period increases, the sampling interval for parity check equations also increases. The number of keystream bits required for the attacks is the number of parity check equations multiplied by the sampling interval. A larger sampling interval leads to a significant increase in data complexity. Meanwhile, many SSCs also limit the number of keystream bits generated by each key-IV pair, making it more challenging for attackers to obtain enough keystream bits for executing the attack algorithms. Another drawback of the above attacks is the wastage of obtained keystream bits, as only the sampled keystream bits are used in the attacks, and unsampled keystream bits are wasted. Then, improving the utilization of these unsampled keystream bits could reduce the data complexity of the attacks.

Time and data complexity are important indices for assessing the performance and

success of the attacks, which need to be considered comprehensively (Meier et al., 1989). It is essential to acknowledge that the obtained keystream bits from the cipher are generally limited to a restricted range due to the LFSR's finite period of the ciphers. Repeated use of the same internal state of LFSR reduces cipher security. If the data complexity of the attacks is too large, the number of keystream bits required for the attacks exceeds the maximum number of keystream bits that the cipher can provide, resulting in the attacks on the cipher failing. Therefore, under certain circumstances, reducing data complexity is more meaningful than reducing time complexity for SSCs.

This paper proposes an idea of multiple sampling on the parity check equations based on the round key period and applies this idea to FCA. Meanwhile, to enable the parity equations obtained from different rounds of sampling to be used together for attacks, we organize sampled parity check equations into different groups and add noise to these equations to ensure that all sampled parity equations have the same round key. Compared to previous single-round sampling, the improved attack with multiple sampling avoids increasing the data complexity of the attack with the growing period of the round key function, making better use of the obtained keystream bits to decrease data complexity. The data complexity of our multiple sampling FCA will gradually decrease with the increase in sampling rounds. According to the limitation of the cipher on the number of keystream bits, the number of sampling rounds can be adjusted to balance the time and data complexity better. To assess the feasibility of our attack, we apply it to analyze the Fruit-80. At the same time, we propose a new design principle to make SSCs resistant to our multiple sampling FCA and improve ciphers' security.

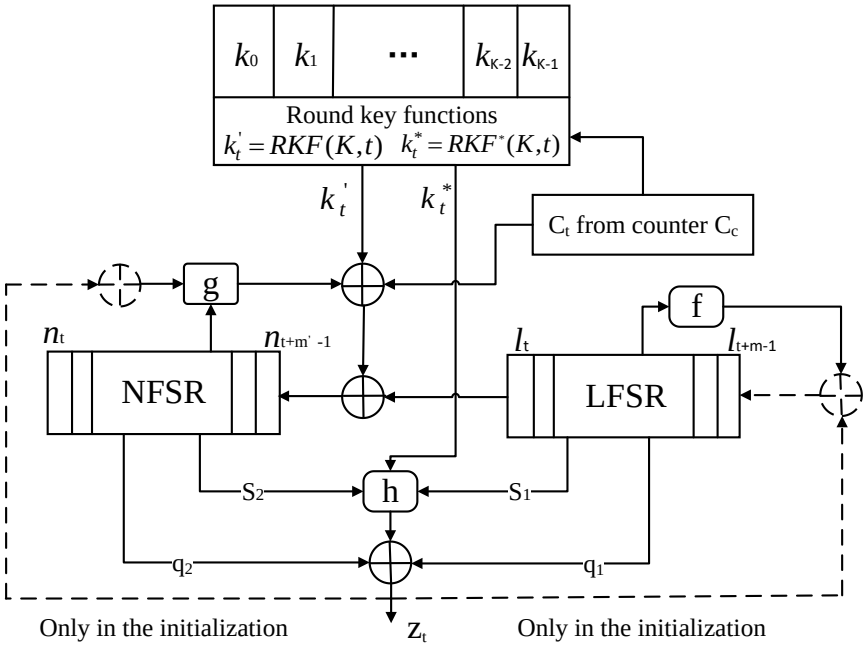
The paper is organized in the following way: Section 2 offers a concise overview of the general SSC model and the principle of FWT. Section 3 proposes an improved FCA based on multiple sampling. In Section 4, we introduce the Fruit-80 stream cipher and apply an improved attack on this cipher. Section 5 provides the countermeasures to strengthen SSCs. Finally, Section 6 concludes the paper.

## 2 Preliminaries

### 2.1 The general SSC model

To facilitate the expression of attack algorithms in the subsequent sections, we refer to the previous stream cipher model (Wang et al., 2019) and propose a more general and comprehensive SSC model that includes the fundamental characteristics and modules of the SSCs such as Plantlet, Sprout, the Fruit-80, and Atom. The model diagram of the SSC is illustrated in Figure 1.

**Figure 1** The general SSC model




LFSR: Consider $L^{(t)} = (l_t, ..., l_{t+m-1})$ as the internal state of the LFSR at time t and m as the length of LFSR. The LFSR's update from $L^{(t)}$ to $L^{(t+1)} = (l_{t+1}, ..., l_{t+m})$ is decided by a linear feedback function $f$ with $l_{t+m} = f(L^{(t)})$.

NFSR: Consider $N^{(t)} = (n_t, ..., n_{t+m'-1})$ as the internal state of the NFSR at time t and m' as the length of NFSR. The NFSR's update from $N^{(t)}$ to $N^{(t+1)} = (n_{t+1}, ..., n_{t+m'})$ is determined by a nonlinear feedback function $g^{'}$,

$$n_{t+m'} = g^{'}(N^{(t)}) = l_t \oplus k_t^{'} \oplus c_t \oplus g(N^{(t)}) \tag{2.1}$$

where $l_t$ represents the output of the LFSR at time t, $c_t$ denotes the counter bit at a fixed position of the counter $C_c$ at time t, and $C_c$ is a counter register with a known initial value and mode of operation. $k_t^{'}$ is the round key bit at time t, generated from the round key function.

Round key function: It is represented as RKF(K, t) and generating round keys $k_t^{'}$ to update the internal states of NFSR. Some SSCs, such as Fruit-80 and Atom, may employ another different round key function to generate additional round keys for updating the nonlinear filtering function h. Let $K = (k_0, ..., k_{\kappa-1})$ denote the cipher's keys, where $\kappa$ represents the security level of the cipher. Two different round keys can be represented as $k_t^{'} = RKF(K,t)$ and $k_t^{*} = RKF^{*}(K,t)$, respectively. The round key function essentially is a random selection or combination of keys at specific locations, and the indexes of the selected key are generally determined by the values of a counter or shift registers.

Output Function: The input parameters of the output function include LFSR bits, NFSR bits, and the output of the nonlinear Boolean function h. The output function usually can be defined as follows:

$$z_t = h(L_{T_{h,L}}^{(t)}, N_{T_{h,N}}^{(t)}) \oplus \bigoplus_{b_1 \in B_1} l_{t+b_1} \oplus \bigoplus_{b_2 \in B_2} n_{t+b_2} \tag{2.2}$$

where $L_{T_{h,L}}^{(t)} = (l_{t+\gamma_1},...,l_{t+\gamma_{s_1}})$ is a subset of $L^{(t)}$ with $0 \le \gamma_1 < ... < \gamma_{s_1} \le m-1$, $N_{T_{h,L}}^{(t)} = (n_{t+\delta_1},...,n_{t+\delta_{s_2}})$ is a subset of $N^{(t)}$ with $0 \le \delta_1 < ... < \delta_{s_2} \le m'-1$, $B_1 = (\sigma_1,...,\sigma_{q_1})$ represents the set of the LFSR taps with $0 \le \sigma_1 < ... < \sigma_{q_1} \le m-1$, $B_2 = (\eta_1,...,\eta_{q_2})$ represents the set of the NFSR taps with $0 \le \eta_1 < ... < \eta_{q_2} \le m'-1$, and $h(L_{T_{h,L}}^{(t)}, N_{T_{h,N}}^{(t)})$ is a nonlinear filtering function that is sometimes directly affected by the round keys $k_t^* = RKF^*(K,t)$.

Initialization phase: Let $IV = (iv_0,...,iv_{m+m'-1})$ be the initial value used to generate the initial state of the cipher. The m initial state bits of the LFSR are loaded with the first m bits of the IV, i.e., $l_i = iv_i$, $0 \le i \le m-1$, while m' LFSR state bits are loaded with the remaining bits of the IV, denoted as $n_{i-m} = iv_i, m \le i \le m+m'-1$. Compared to the keystream generation phase, the output function is fed back and XORed with the input of NFSR and LFSR in the initialization phase, i.e., $l_{t+m} = f(L^{(t)}) \oplus z_t$ and $n_{t+m'} = g'(N^{(t)}) = l_t \oplus k_t' \oplus c_t \oplus g(N^{(t)}) \oplus z_t$. Subsequently, the cipher is clocked g times without producing any keystream bits. Typically, the last few bits of the LFSR state are assigned a non-zero constant value to ensure that the output of the LFSR is not all 0.

2.2 Walsh-Hadamard Transform

The time complexity of basic FCA is $O(N2^m)$, where N is the number of parity check equations used for checking the correct initial state, and m is the size of the cipher's initial state. Chose et al. (2002) utilize the guess and evaluation procedure of the attacks as a Walsh-Hadamard transform (WHT) To decrease the FCAs' time complexity further. The fast Walsh-Hadamard transform (FWHT) can be applied to this procedure and accelerate recovery of the cipher's initial state. FWHT can be used to reduce the time complexity of FCA from $O(N2^m)$ to $O(N+m2^m)$. The following illustrates the transformation of the parity check equations used for the attacks into the form of WHT. Given a function $w: \{0,1\}^m \to Z$, the WHT of $w$ is defined as

$$\hat{w}(s) = \sum_{x \in \{0,1\}^m} w(x)(-1)^{<s,x>} \tag{2.3}$$

The general form of parity check equations is shown as $e_t = <s^t, \Gamma> \oplus z_t$, where $\Gamma$ is a linear mask, $s^t$ is the internal state of LFSR at time t, $z_t$ is keystream bit generated from the cipher at time t, and $e_t$ represents the noise introduced by the linear mask $\Gamma$. If the guessed initial state $s \in \{0,1\}^m$, the correlation $\sum_{t=0}^{N-1}(-1)^{e_t}$ is transformed into the following form,

$$\sum_{t=0}^{N-1}(-1)^{e_t} = \sum_{t=0}^{N-1}(-1)^{<s,\Gamma\times^T F^t>\oplus z_t}$$
$$= \sum_{x\in\{0,1\}^m}\Big(\sum_{t\in\{0,1,\ldots,N-1\mid\Gamma\times^T F^t=x\}}(-1)^{<s,x>\oplus z_t}\Big) \tag{2.4}$$
$$= \sum_{x\in\{0,1\}^m}\Big(\sum_{t\in\{0,1,\ldots,N-1\mid\Gamma\times^T F^t=x\}}(-1)^{\oplus z_t}\Big)(-1)^{<s,x>}$$

Consequently, the FWHT can be applied to the above parity check equations. When initial states $s^0$ are correctly guessed, statistic $\sum_{t=0}^{N-1}(-1)^{e_t}$ can exhibit a high bias.

## 3 Multiple Sampling FCA Algorithm

In this section, we first discuss the existing issue of the previous FCA on SSCs and subsequently explore how multiple sampling can efficiently solve this issue. Secondly, we propose an improved FCA algorithm by applying the idea of multiple sampling to FCA.

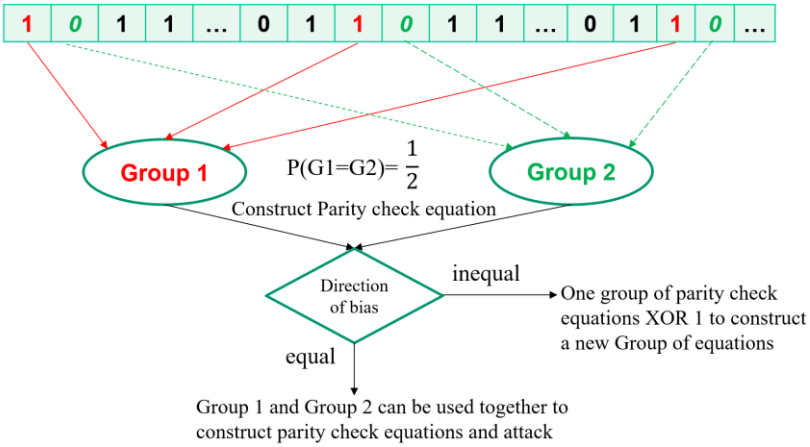
### 3.1 The idea of multiple sampling for parity check equations

Since the SSCs involve the round key function, obtaining suitable linear approximate equations without round keys is challenging. A previous FCA used the idea of sampling to eliminate the impact of the round keys on parity check equations. This method allows to construct the parity check equations without round keys, facilitating the recovery of the cipher's internal state (Wang et al., 2019). However, because FCA with sampling requires more keystream bits to construct enough parity check equations, this approach dramatically increases the data complexity of the attacks, resulting in significant data waste. We assert that the data complexity of an attack on a cipher should be determined by all keystream bits required for the attack rather than the number of discontinuous keystream bits used for the attack. Because attackers can't get only discrete keystream bits. In the case of a round key function with a period of 80, the previous FCA with single round sampling only selects 1 bit from every 80 keystream bits to construct parity check equations for the attack, leaving 79 keystream bits unused in each period of round key function. Although these 79 keystream bits have not been effectively used for the attack, we cannot ignore them when calculating data complexity. Thus, the total data complexity should include these 79 keystream bits in each period. Our method will result in a remarkably low data utilization rate and greater data complexity. As the round key period of the cipher increases, these problems will worsen.

To address the problems of FCA mentioned above, we consider increasing the number of sampling rounds to use more keystream bits that have yet to be sampled or used to construct parity check equations. In single-round sampling, keystream bits at a fixed order in each cycle are selected to ensure that the equations constructed with these keystream bits have the same round key, eliminating the influence of the round keys on the bias absolute value of the statistical test. Thus, these parity check equations acquired during each sampling period have the same round key. Since the value of the round key is either 0 or 1, the probability of the parity check equations obtained from two sampling positions of different rounds with the same round key is 1/2. An intuitive explanation

of the multiple sampling process is provided in Figure 2.

**Figure 2** Multiple sampling of parity-check equation




The values in the boxes represent the values of round keys at clocks, which are cycled continuously to participate in the parity check equations. The first and second positions' values of each round key period are sampled separately, and the parity check equations with the sampled round keys are employed to recover the initial state of the cipher. We put the parity check equations from the first round of sampling into Group 1 and the parity check equations from the second round of sampling into Group 2 to ensure that parity check equations in the same group share the same round key. Considering the probability of 1/2 that two groups of parity check equations share the same round key. If both groups of parity check equations involve the same round key, these equations can be directly used to restore the correct initial state of LFSR by hypothesis testing. However, if the two groups of parity check equations involve different round keys, two groups of equations cannot be directly employed together to attack the cipher.

Parity check equations obtained from different rounds of sampling may involve different round keys. However, when the round key is a fixed constant, it can only affect the direction of the bias within parity check equations without altering the absolute value of the bias. Therefore, by assessing the directions of biases, we can determine whether the round key values in Group 1 and Group 2 are the same. Considering the situation that the round keys within one group of parity check equations differ from those in the other group, to eliminate the round keys' influence on the biases of sampled parity check equations, we can XOR all parity check equations of Group 2 with 1 to get new Group 3 of equations. We can see that if the biases' directions of the equations in Group 1 and Group 2 are different, the directions of those in Group 1 and Group 3 must be the same. Consequently, we can get two groups of equations with the same bias value, and then these two groups of equations can be used together to verify the correct initial state.

Sampling the parity check equations in multiple rounds has advantages over single-round sampling. Our method optimizes the utilization of the keystream bits at different locations in each period, efficiently reducing data complexity by increasing the number of sampling rounds. For instance, the cipher has a round key period of 80, and an attack requires at least 1000 parity check equations to analyze this cipher

successfully. The data complexity in a previous FCA with single-round sampling is 80×1000. However, our improved FCA with two-round sampling uses two keystream bits in each period of the round key function to construct the parity check equations. Then, the data complexity is reduced to 80×500. Increasing the number of sampling rounds allows more keystream bits to be effectively employed in constructing parity check equations. This approach can significantly reduce the data complexity of the attack.

### 3.2 Improved FCA based on multiple sampling

This section introduces how to use the improved multiple sampling FCA algorithm to construct enough parity check equations and apply it to the general model of SSC.

Most SSCs are designed to enhance the security of ciphers against FCA by improving nonlinear filtering functions and integrating additional functions into ciphers. It is a challenge to find the correlation between individual internal state bits and corresponding keystream bits. Therefore, this paper searches for the high correlation between the sum of some internal state bits and corresponding keystream bits. This approach can increase the number of available linear masks and the probability of discovering linear masks with high biases. Firstly, we conduct linear approximation on the sum of keystream bits and assume the index values of the sum of SSC bits are on set $T_z$, set $T_z$ will be determined later. Then, we can derive the following equation,

$$\bigoplus_{i\in T_z} z_{t+i} = \bigoplus_{i\in T_z}(h(L_{T_{h,L}}^{t+i}, N_{T_{h,N}}^{t+i}) \oplus \bigoplus_{b_1\in B_1} l_{t+i+b_1} \oplus \bigoplus_{b_2\in B_2} n_{t+i+b_2}) \tag{3.1}$$

To eliminate state bits of the NFSR from equations and independently recover the initial state of the LFSR, an appropriate linear approximation is required for the update function of the NFSR. The approximate equation of NFSR with bias $\varepsilon_{g^*}$ is depicted below.

$$n_{t+m'} \approx k_t^{'} \oplus c_t \oplus l_t \oplus \bigoplus_{i\in I_g} n_{t+i} \tag{3.2}$$

where $I_g$ represents the set comprising the indices of the linear terms for the NFSR update function. Then, we select a suitable $T_z = I_g \cup \{m'\}$ to ensure that $\bigoplus_{i\in T_z} n_{t+b_2+i}$ has a high bias while reducing the number of the NFSR state bits in equations. Then, the sum of NFSR bits can be denoted as

$$\begin{aligned}\bigoplus_{i\in T_z} n_{t+b_2+i} &= \bigoplus_{i\in I_g} n_{t+b_2+i} \oplus n_{t+b_2+m'} \\ &= k_{t+b_2}^{'} \oplus c_{t+b_2} \oplus l_{t+b_2} \oplus g^*(N^{(t+b_2)})\end{aligned} \tag{3.3}$$

where $g^*(N^t) = \bigoplus_{i\in I_g} n_{t+i} \oplus g(N^t)$ and its bias is $\varepsilon_{g^*}$.

Then, we consider the linear approximation of the nonlinear filtering function $h(L_{T_{h,L}}^{(t+i)}, N_{T_{h,N}}^{(t+i)})$ with bias $\varepsilon_{h,i}(a_i)$. Let $a_i \in \{0,1\}^{s_1+s_2}$ denote the linear masks of function $h(L_{T_{h,L}}^{(t+i)}, N_{T_{h,N}}^{(t+i)})$ at the clock $t+i$, i.e., $a_i = (a_i[1], \ldots, a_i[s_1+s_2])$,

$$\begin{aligned}&h(L_{T_{h,L}}^{(t+i)}, N_{T_{h,N}}^{(t+i)}) \approx a_i \cdot h(L_{T_{h,L}}^{(t+i)}, N_{T_{h,N}}^{(t+i)})^T \\ &= (a_i[1], \ldots, a_i[s_1]) \cdot (L_{T_{h,L}}^{(t+i)})^T \oplus (a_i[s_1+1], \ldots, a_i[s_1+s_2]) \cdot (N_{T_{h,N}}^{(t+i)})^T\end{aligned} \tag{3.4}$$

The dot operator $\cdot$ between a column vector and a row vector represents the inner product in GF(2). There are $|T_z|$ h functions that require approximation. Thus, the total linear masks for all linear approximations of $|T_z|$ connected h can be represented as $a_{T_z} \in \{0,1\}^{(s_1+s_2)\times|T_z|}$. As the piling-up lemma, the total biases of linear masks $a_{T_z}$ are calculated as $\varepsilon_{h,T_z}(a_{T_z}) = 2^{|T_z|-1} \times \prod_{i\in T_z} \varepsilon_{h,i}(a_i)$.

To present different types of terms in the linear approximation equation more clearly, we organize the equation into the following form,

$$\begin{aligned}\bigoplus_{i\in T_z} z_{t+i} \approx & \bigoplus_{i\in T_z}\Big(\bigoplus_{b_1\in B_1} l_{t+i+b_1}\Big) \oplus \bigoplus_{b_2\in B_2} l_{t+b_2} \oplus \bigoplus_{i\in T_z}\big((a_i[1],\ldots,a_i[s_1])\cdot (L_{T_{h,L}}^{(t+i)})^T\big) \\ & \oplus \bigoplus_{b_2\in B_2} k'_{t+b_2} \oplus \bigoplus_{b_2\in B_2} c_{t+b_2} \\ & \oplus \Big(\bigoplus_{i\in T_z}\big((a_i[s_1+1],\ldots,a_i[s_1+s_2])\cdot (N_{T_{h,N}}^{(t+i)})^T\big) \oplus \bigoplus_{b_2\in B_2} g^*(N^{(t+b_2)})\Big)\end{aligned} \tag{3.5}$$

The part of the equation involving the LFSR's internal states and keys needs to be guessed in our attack, and $c_t$ is the counter bit at a fixed position of the counter $C_c$, which is known. Therefore, if we can ensure the term that involves the internal state of the NFSR has a high bias, our improved FCA can be adopted. Let

$$\varepsilon_{g^*,B_2}(a_{T_z}) = \Pr\Big[\bigoplus_{i\in T_z}\big((a_i[s_1+1],\ \ldots,\ a_i[s_1+s_2])\cdot (N_{T_{h,N}}^{(t+i)})^T\big) \oplus \bigoplus_{b_2\in B_2} g^*(N^{(t+b_2)}) = 0\Big] - \frac{1}{2} \tag{3.6}$$

and this bias is independent of $(a_i[1],\ \ldots,\ a_i[s_1])$. If $\varepsilon_{g^*,B_2}(a_{T_z})$ is high enough for fixed $a_{T_z}$, we can derive the following linear approximation equation with bias value $2\times\varepsilon_{h,T_z}(a_{T_z})\times\varepsilon_{g^*,B_2}(a_{T_z})$,

$$\begin{aligned}\bigoplus_{i\in T_z} z_{t+i} \approx & \bigoplus_{i\in T_z}\Big(\bigoplus_{b_1\in B_1} l_{t+i+b_1}\Big) \oplus \bigoplus_{b_2\in B_2} l_{t+b_2} \oplus \bigoplus_{i\in T_z}\big((a_i[1],\ldots,a_i[s_1])\cdot (L_{T_{h,L}}^{(t+i)})^T\big) \\ & \oplus \bigoplus_{b_2\in B_2} k'_{t+b_2} \oplus \bigoplus_{b_2\in B_2} c_{t+b_2}\end{aligned} \tag{3.7}$$

The internal state of LFSR at time t can be denoted as $L^{(t)} = L^{(0)} \times F^t$, where $L^{(0)}$ represents the initial internal state of the LFSR, and F is the state transition matrix of the LFSR, which is determined by the taps of the feedback function of the LFSR. Then, the above linear approximation equation can be simplified into the following form :

$$\bigoplus_{i\in T_z} z_{t+i} \approx L^{(0)}\cdot(F^t \times U(a_{T_z})) \oplus \bigoplus_{b_2\in B_2} k'_{t+b_2} \oplus \bigoplus_{b_2\in B_2} c_{t+b_2} \tag{3.8}$$

where

$$U(a_{T_z}) = \bigoplus_{i\in T_z}\Big(\bigoplus_{b_1\in B_1} g_{i+b_1} \oplus \bigoplus_{j\in\{1,\ldots,s_1\}} a_i[j]\cdot g_{i+T_{h,L}[j]}\Big) \oplus \bigoplus_{b_2\in B_2} g_{b_2} \tag{3.9}$$

where $g_q$ is the first column of the matrix $F^q$, $T_{h,L}[j]$ is the jth element of $T_{h,L}$. By using the above equations, we can derive the linear approximation equation with a linear mask u,

$$\bigoplus_{i\in T_z} z_{t+i} \approx L^{(0)}\cdot(F^t \times u) \oplus \bigoplus_{b_2\in B_2} k'_{t+b_2} \oplus \bigoplus_{b_2\in B_2} c_{t+b_2} \tag{3.10}$$

where $u \in \{0,1\}^m$ is a column vector of length m. If different $a_{T_z}$ derive the same

linear mask u through U(.), the bias of u is the sum of all corresponding biases,

$$\varepsilon_u = \sum_{\{a_{T_z} | U(a_{T_z})=u\}} 2 \times \varepsilon_{h,T_z}(a_{T_z}) \times \varepsilon_{g^*,B}(a_{T_z}) \tag{3.11}$$

Let $\hat{z}_t = \bigoplus_{i \in T_z} z_{t+i}$, $\hat{k}_t = \bigoplus_{b_2 \in B_2} k'_{t+b_2}$, $\hat{c}_t = \bigoplus_{b_2 \in B_2} c_{t+b_2}$. If r different linear masks with high biases can be discovered, we can obtain r distinct linear approximation equations with the guessed variables $L^{(0)} = (l_0, ..., l_{m-1})$ and $\hat{k}_t$, as depicted below :

$$L^{(0)} \cdot (F^t \times u_j) \oplus \hat{z}_t \oplus \hat{c}_t \oplus \hat{k}_t = e_{t,j} \quad t \geq 0, j = 1, ..., r \tag{3.12}$$

where $e_{t,j}$ is the random noise introduced by linear approximation on $\hat{z}_t$ with linear mask $u_j$ and satisfies $\Pr[e_{t,j} = 0] = \frac{1}{2} + \varepsilon_j, \varepsilon_j \approx \varepsilon_u$.

According to the "communitive" property in the finite field (Todo et al., 2018), we can get $L^{(0)} \cdot (F^t \times u_j) = (L^{(0)} \times F_{u_j}) \cdot g_t$. Then, we construct parity check equations using the linear mask $g_{dt}$ instead of $F^{dt} \times u_j$,

$$(L^{(0)} \times F_{u_j}) \cdot g_t \oplus \hat{z}_t \oplus \hat{c}_t \oplus \hat{k}_t = e_{t,j} \quad t \geq 0, j = 1, ..., r \tag{3.13}$$

where $F_{u_j} = [u_j, F^1 \times u_j, ..., F^{m-1} \times u_j]$, m is the length of the initial state $L^{(0)}$, and $g_t$ is the first column vector of the matrix $F^t$.

When multiple linear masks with high bias are available, obtaining more initial states with high bias statistical characteristics is possible. The parity check equations, constructed using the "communitive" property, break the conventional false initial state hypothesis. This approach facilitates recovery of the correct initial state of the cipher without the need to guess all initial state bits of LFSR. We can ignore $\beta$ bits of the initial state and only guess remaining $m - \beta$ bits. Due to the existence of multiple initial states with high biases, even if some initial states are omitted, the correct initial state can still be obtained. Flexible selection of $\beta$ can effectively balance time and data complexity of attack.

Assuming that the round key function of the cipher is periodic and the period of round keys is d, denoted as $k'_{t_0+dt'} = k'_{t_0}$ for $t_0 = 0, ..., d-1$ and $t' \geq 0$. We can generate many parity check equations with unknown variables $L^{(0)} = (l_0, ..., l_{m-1})$ and $\hat{k}_0$, along with r linear mask,

$$(L^{(0)} \times F_{u_j}) \cdot g_{t_0+dt'} \oplus \hat{z}_{t_0+dt'} \oplus \hat{c}_{t_0+dt'} \oplus \hat{k}_{t_0} = e_{t_0+dt',j} \tag{3.14}$$
$$t' = 0, ..., \Omega - 1, \ j = 1, ..., r, t_0 = 0, ..., d-1$$

where $\Omega$ represents the number of parity check equations required for getting the correct initial state of LFSR with a high probability. The range of quantities for $\Omega$ is obtained from the Skellam distribution, $\Omega \geq \frac{\pi 2^{\beta+2}(m+1)\ln 2}{r\varepsilon^2}$, where $\beta$ is the number of omitted variables of the initial state (Wang et al., 2019). The period of round key function is used as the period of sampling to collect parity check equations. Although this method can obtain parity check equations with the same round key, it inevitably

leads to the wastage of keystream bits and increased data complexity of the attack. Consequently, we adopt multiple rounds of sampling to construct the parity check equations. When $t_0$=0 or 1, the following two groups of equations can be derived; we consider them as Group 1 and Group 2 separately.

$$(L^{(0)} \times F_{u_j}) \cdot g_{dt'} \oplus \hat{z}_{dt'} \oplus \hat{c}_{dt'} \oplus \hat{k}_0 = e_{dt',j} \text{ t'=0,...,} \frac{\Omega-1}{2}, j=1,...,r \tag{3.15}$$

$$(L^{(0)} \times F_{u_j}) \cdot g_{dt'+1} \oplus \hat{z}_{dt'+1} \oplus \hat{c}_{dt'+1} \oplus \hat{k}_1 = e_{dt'+1,j} \text{ t'=0,...,} \frac{\Omega-1}{2}, j=1,...,r \tag{3.16}$$

The values $\hat{k}_0$ and $\hat{k}_1$ are constant and only influence the bias directions without changing the absolute value of the bias. When employing two groups of parity check equations simultaneously to verify the correct initial state, it is imperative to ensure that $\hat{k}_0$ and $\hat{k}_1$ are equal. Suppose the round keys of two groups of parity check equations differ, the biases calculated by the correct initial state for the two groups of parity check equations will be opposite, causing the total bias approach to be 0. Then, the initial state of the cipher cannot be restored successfully. To resolve this issue, we construct Group 3 of equations, derived from all equations in Group 1 XOR 1. The Group 3 of equations is as follows,

$$(L^{(0)} \times F_{u_j}) \cdot g_{dt'} \oplus \hat{z}_{dt'} \oplus \hat{c}_{dt'} \oplus \hat{k}_0 \oplus 1 = e_{dt',j} \oplus 1, \text{ t'=0,...,} \frac{\Omega-1}{2}, j=1,...,r \tag{3.17}$$

Equations from Groups 1 and 2 are employed together for statistical analysis, while equations from Groups 2 and 3 are utilized similarly. Consequently, two groups of parity check equations that display biases in opposite directions are discarded. Conversely, the remaining two groups of parity equations exhibit the bias in the same direction, making them suitable for being used together to determine the correct initial state. Assuming equations from Groups 1 and 2 have different round keys, and we can XOR 1 with equations from Groups 1 to get equations of Group 3, then equations from Groups 2 and 3 have the same round key, i.e., $\hat{k}_1 = \hat{k}_0 \oplus 1$.

$$\begin{cases} (L^{(0)} \times F_{u_j}) \cdot g_{dt'+1} \oplus \hat{z}_{dt'+1} \oplus \hat{c}_{dt'+1} \oplus \hat{k}_1 = e_{dt'+1,j} & \text{t'=0,...,} \frac{\Omega-1}{2}, j=1,...,r \\ (L^{(0)} \times F_{u_j}) \cdot g_{dt'} \oplus \hat{z}_{dt'} \oplus \hat{c}_{dt'} \oplus \hat{k}_0 \oplus 1 = e_{dt',j} \oplus 1 & \text{t'=0,...,} \frac{\Omega-1}{2}, j=1,...,r \end{cases} \tag{3.18}$$

We can merge and simplify the above two groups of equations as follows,

$$(l'_0,...,l'_{m-1}) \cdot g_t \oplus \hat{z}_t \oplus \hat{c}_t \oplus \hat{k}, \text{ t=0,...,} \Omega-1 \tag{3.19}$$

where $L'^{(0)} = (l'_0,...,l'_{m-1})$ represents the guessed value of $L^{(0)} \times F_{u_j}$. We introduce an indicator for all parity check equations as

$$\Delta_t(l'_0,...,l'_{m-1}) = (l'_0,...,l'_{m-1}) \cdot g_t \oplus \hat{z}_t \oplus \hat{c}_t \oplus \hat{k} \tag{3.20}$$

If the value of $L'^{(0)}$ is guessed as $L^{(0)} \times F_{u_j}$ and the round key $\hat{k}$ is guessed correctly, then we can get $\Delta_t(L'^{(0)}) = e_{t,j}$, $j=1,...r$ and $\Pr[\Delta_t(L'^{(0)})=0] = \frac{1}{2} + \varepsilon_j$. If the value of $L'^{(0)}$ is guessed as $L^{(0)} \times F_{u_j}$ and $\hat{k}$ is guessed wrongly, then $\Delta_t(L'^{(0)}) = e_{t,j} \oplus 1$, $j=1,...,r$ and $\Pr[\Delta_t(L'^{(0)})=0] = \frac{1}{2} - \varepsilon_j$. If the guessed initial state $L'^{(0)}$ does not

belong to the set $\{L^{(0)} \times F_{u_j}, j=1,...,r\}$, then $\Delta_t(L'^{(0)})$ is generally assumed to perform randomly and $\Pr[\Delta_t(L'^{(0)})=0]=\frac{1}{2}$. Given $S_h=\{L^{(0)} \times F_{u_j}, j=1,...,r\}$ and $S_l=\{0,1\}^m \setminus S_h$, we define the statistic $\varepsilon$ of bias of indicator as

$$\mathcal{E}(l'_0,...,l'_{m-1})=\sum_{t=0}^{\Omega-1}(-1)^{\Delta_t(l'_0,...,l'_{m-1})} \tag{3.21}$$

According to the central limit theorem and value of $\hat{k}$, we can obtain representations of the statistic,

$$\mathcal{E}(L'^{(0)}) \sim N(\Omega\varepsilon, \Omega(1-\varepsilon^2)) \approx N(\Omega\varepsilon, \Omega) \tag{3.22}$$

or

$$\mathcal{E}(L'^{(0)}) \sim N(-\Omega\varepsilon, \Omega) \tag{3.23}$$

when $L'^{(0)} \in S_h$, where $N(\cdot,\cdot)$ represents the normal distribution with variance and expectation parameters, $\varepsilon^2$ is small enough to be disregarded, and

$$\mathcal{E}(L'^{(0)}) \sim N(0, \Omega) \tag{3.24}$$

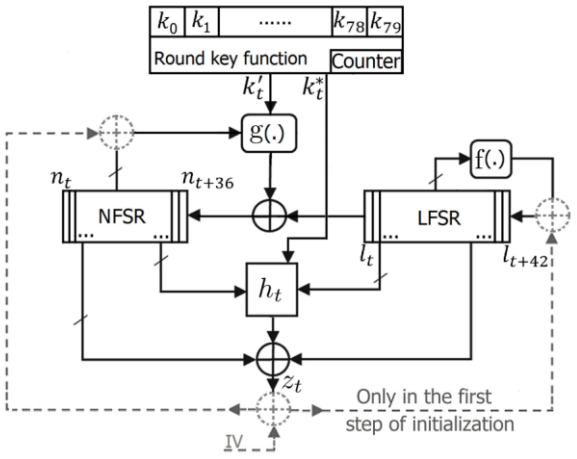
when $L'^{(0)} \in S_l$. To further reduce the time complexity of the attack, we apply FWHT to the guessing and evaluation process of the cipher's initial state. The details of FWHT are introduced in Section 2.

## 4 Application of the attack on Fruit-80

### 4.1 Brief description of Fruit-80

Fruit-80 is a bit-oriented SSC with an 80-bit key $K=(k_0,...,k_{79})$ and a 70-bit known initial vector $IV=(iv_0,...iv_{69})$. It includes four main parts: a 43-bit LFSR whose state at time t is denoted as $L^{(t)}=(l_t,...,l_{t+42})$, a 37-bit NFSR whose state at time t is denoted as $N^{(t)}=(n_t,...,n_{t+36})$, an 80-bit register used for storing fixed key, and a 7-bit counter register denoted as $C_{c=}(c_t^0,...,c_t^6)$ used for the update of round key functions.

**Figure 3** The Block Diagram of Fruit-80



Update function of LFSR:

$$l_{t+43} = f(L^{(t)}) = l_t \oplus l_{t+8} \oplus l_{t+18} \oplus l_{t+23} \oplus l_{t+28} \oplus l_{t+37} \tag{4.1}$$

Update function of NFSR:

$$\begin{aligned} n_{t+37} &= k_t^{'} \oplus l_t \oplus g(N^{(t)}) \\ &= k_t^{'} \oplus l_t \oplus n_t \oplus n_{t+10} \oplus n_{t+20} \oplus n_{t+12}n_{t+3} \\ &\oplus n_{t+14}n_{t+25} \oplus n_{t+5}n_{t+23}n_{t+31} \\ &\oplus n_{t+8}n_{t+18} \oplus n_{t+28}n_{t+30}n_{t+32}n_{t+34} \end{aligned} \tag{4.2}$$

Round key functions:

$$k_t^{'} = RKF(K,t) = k_r k_{p+16} k_{q+48} \oplus k_r k_{p+16} \oplus k_{p+16} k_{q+48} \oplus k_r k_{q+48} \oplus k_{p+16} \tag{4.3}$$

$$k_t^{*} = RKF*(K,t) = k_r k_{p+16} \oplus k_{p+16} k_{q+48} \oplus k_r k_{q+48} \oplus k_r \oplus k_{p+16} \oplus k_{q+48} \tag{4.4}$$

where $r = (c_t^0 c_t^1 c_t^2 c_t^3)$, $p = (c_t^1 c_t^2 c_t^3 c_t^4 c_t^5)$, $q = (c_t^2 c_t^3 c_t^4 c_t^5 c_t^6)$.

Nonlinear filtering function:

$$\begin{aligned} h(L_{T_{h,L}}^{(t)}, N_{T_{h,N}}^{(t)}, k_t^{*}) &= k_t^{*}(n_{t+36} \oplus l_{t+19}) \oplus l_{t+6}l_{t+15} \\ &\oplus l_{t+1}l_{t+22} \oplus n_{t+35}l_{t+27} \oplus n_{t+1}n_{t+24} \\ &\oplus n_{t+1}n_{t+33}l_{t+42} \end{aligned} \tag{4.5}$$

where $L_{T_{h,L}}^{(t)} = (l_{t+1}, l_{t+6}, l_{t+15}, l_{t+19}, l_{t+22}, l_{t+27}, l_{t+42})$, $N_{T_{h,N}}^{(t)} = (n_{t+1}, n_{t+24}, n_{t+33}, n_{t+35}, n_{t+36})$.

Keystream output function:

$$\begin{aligned} z_t &= h(L_{T_{h,L}}^{(t)}, N_{T_{h,N}}^{(t)}, k_t^{*}) \oplus l_{t+38} \oplus n_t \oplus n_{t+7} \oplus n_{t+19} \\ &\oplus n_{t+29} \oplus n_{t+36} \end{aligned} \tag{4.6}$$

The initialization phase:

It connects 1000000000 in front of the IV bits to extend the length of the IV to 80 bits, i.e., $IV' = (1000000000 iv_0 iv_1 ... iv_{68} iv_{69})$. Then keys are separately loaded into the LFSR and NFSR, i.e., ( $k_0 \to n_0, k_1 \to n_1, ... k_{36} \to n_{36}, k_{37} \to l_0, k_{38} \to l_1, ..., k_{79} \to l_{42}$ ). Additionally, the counter $C_c$ denoted as $(c_0^0 c_0^1 ... c_0^5 c_0^6)$ are set to 0. Firstly, the XOR of the output bits and IV bits are fed into the NFSR and LFSR ( $z_i \oplus v_i^{'}, 0 \le i \le 79$ ), and then the cipher is clocked 80 times. The counter $C_c$( $c_{80}^0 = n_{80}, c_{80}^1 = n_{81}, ..., c_{80}^5 = n_{85}, c_{80}^6 = l_{80}$ ) is updated using the NFSR and LFSR, and the $l_{80}$ is set to 1. Finally, the cipher is clocked 80 times without any feedback during the LFSR and NFSR update process. The output generated during the initialization phase is not used as keystream bits to encrypt information. The initialization phase aims to provide a secure internal state for the cipher.

## 4.2 Attack on Fruit-80

The overall round key period of Fruit-80 is determined by its round keys $k_t^{*}$ and $k_t^{'}$, which is the least common multiple (LCM) of the two round key periods. The period of two round keys can be calculated individually. Due to periods of $k_t^{*}$ and $k_t^{'}$ are

both 128, the overall round key period of Fruit-80 is 128. Then, we can analyze the internal structure of Fruit-80 and apply the improved FCA in Section 3 to it.

The linear approximation of the NFSR's update function is presented below, and its bias is calculated as $2^{-4.6}$.

$$n_{t+37} \approx k_t^{'} \oplus l_t \oplus n_t \oplus n_{t+10} \oplus n_{t+20} \oplus c_t^3 \tag{4.7}$$

To eliminate the NFSR terms in the parity check equations as much as possible and make the bias higher, we choose the set of keystream indexes as $T_z = \{0,10,20,37\}$. Then, the keystream bits of Fruit-80 can be expressed as

$$\begin{aligned} \bigoplus_{i\in T_z} z_{t+i} = & \bigoplus_{i\in T_z} l_{t+i+38} \oplus \bigoplus_{b\in B} l_{t+b} \bigoplus_{i\in T_z} h(L_{T_{h,L}}^{(t+i)}, N_{T_{h,N}}^{(t+i)}, k_t^*) \\ & \oplus \bigoplus_{b\in B} g^*(N^{(t+b)}) \oplus \bigoplus_{b\in B} k_{t+b}^{'} \oplus \bigoplus_{b\in B} c_{t+b}^3 \end{aligned} \tag{4.8}$$

where B={0,7,19,29,36}, $g^*(N^{(t)}) = n_t \oplus n_{t+10} \oplus n_{t+20} \oplus g(N^{(t)})$ and $\Pr[g^*(N^{(t)} = 0)] = \frac{1}{2} + 2^{-4.6}$.

Let $a_i = (a_i[0],...,a_i[11])$ denote the linear mask of the h function at time t+i. The linear approximation of the h function can be expressed as

$$h(L_{T_{h,L}}^{(t+i)}, N_{T_{h,N}}^{(t+i)}) \approx (a_i[0],...,a_i[6]) \cdot (L_{T_{h,L}}^{(t+i)})^T \oplus (a_i[7],...,a_i[11]) \cdot (N_{T_{h,N}}^{(t+i)})^T \tag{4.9}$$

with the biases $\varepsilon_{h,i}(a_i) = \pm 2^{-6}, \pm 2^{-7}$ or 0. Due to the $\bigoplus_{i\in T_z} z_{t+i}$ having 4 h functions, each function needs to be approximated separately. We denote $a_{T_z} = (a_0, a_{10}, a_{20}, a_{37})$ as the linear mask, connected with four linear masks of length 12. According to the pilling-up lemma, the bias of 4 h functions' connected approximations can be calculated as

$$\varepsilon_{h,T_z}(a_{T_z}) = 2^3 \times \prod_{i\in T_z} \varepsilon_{h,i}(a_i) \tag{4.10}$$

To exclude the NFSR terms in the parity check equations, the bias $\varepsilon_{g^*,B}(a_{T_z})$ for all NFSR terms in these equations is computed.

$$\varepsilon_{g^*,B}(a_{T_z}) = \Pr[\bigoplus_{i\in T_z} ((a_i[7],...,a_i[11]) \cdot (N_{T_{h,N}}^{(t+i)})^T) \oplus \bigoplus_{b\in B} g^*(N^{(t+b)}) = 0] - \frac{1}{2} \tag{4.11}$$

This bias remains independent on $(a_i[0],\ldots,a_i[6])$ for $i \in T_z$. Then, we refer to the linear masks $a_{T_z}$ with non-zero biases $\varepsilon_{g^*,B}(a_{T_z})$ found in (Wang et al., 2019) and present a portion of them in the following Table 1, where * represents 0 or 1.Then, linear approximation equations of $\bigoplus_{i\in T_z} z_{t+i}$ can be derived for any $a_{T_z}$,

$$\bigoplus_{i\in T_z} z_{t+i} \approx \bigoplus_{i\in T_z} l_{t+i+38} \oplus \bigoplus_{b\in B} l_{t+b} \bigoplus_{i\in T_z} (a_i[0],...,a_i[6]) \cdot (L_{T_{h,L}}^{(t+i)})^T \oplus \bigoplus_{b\in B} k_{t+b}^{'} \tag{4.12}$$

and its bias is calculated as $2 \times \varepsilon_{h,T_z}(a_{T_z}) \times \varepsilon_{g*,B}(a_{T_z})$.The above linear approximation equation can be simplified to :

$$\bigoplus_{i\in T_z} z_{t+i} \approx L^{(0)} \cdot (F^t \times u) \oplus \bigoplus_{b\in B} k_{t+b}^{'} \tag{4.13}$$

where $u \in \{0,1\}^{43}$ are linear masks. If different $a_{T_z}$ are transformed by $U(\cdot)$ into the

same linear mask u, the biases of these $a_{T_z}$ should be added to obtain the bias of linear mask $u$. The bias $\varepsilon_u$ can be expressed as

$$\varepsilon_u = \sum_{\{a_{T_z} | u=U(a_{T_z})\}} 2\times\varepsilon_{h,T_z}(a_{T_z})\times\varepsilon_{g^*,B}(a_{T_z}) \tag{4.14}$$

where

$$\begin{aligned} U(a_{T_z}) = \bigoplus_{i\in T_z}((a_i[0]\cdot g_{i+1} \oplus a_i[1]\cdot g_{i+6} \oplus a_i[2]\cdot g_{i+15} \oplus a_i[3]\cdot g_{i+19} \\ \oplus a_i[4]\cdot g_{i+22} \oplus a_i[5]\cdot g_{i+27} \oplus a_i[6]\cdot g_{i+42}) \oplus g_{i+38}) \oplus \bigoplus_{b\in B} g_b \end{aligned} \tag{4.15}$$

By exhaustively searching all $a_{T_z}[0,...,6]$, we can get r=$2^{20}$ $u$ with absolute values of biases greater than $th = \frac{\varepsilon}{2} = 2^{-31.62}$.

**Table 1** Biases of linear masks when $a_i[7:11],\ i\in T_z$ are fixed

| $a_0[8]$ | $a_0[9]$ | $a_0[10]$ | $a_{10}[8]$ | $a_{10}[9]$ | $a_{20}[7]$ | $a_{20}[8]$ | $a_{20}[9]$ | $a_{37}[7]$ | $a_{37}[8]$ | $a_{37}[9]$ | $\varepsilon_u$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $+2^{-13.28}$ |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $+2^{-17.80}$ |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $+2^{-13.28}$ |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $+2^{-17.80}$ |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $+2^{-14.86}$ |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $+2^{-19.39}$ |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $+2^{-14.86}$ |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $+2^{-19.39}$ |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | $+2^{-13.28}$ |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | $+2^{-17.80}$ |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | $-2^{-13.28}$ |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | $-2^{-17.80}$ |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | $+2^{-14.86}$ |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | $+2^{-19.39}$ |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | $-2^{-14.86}$ |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | $-2^{-19.39}$ |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | $+2^{-15.26}$ |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | $+2^{-18.06}$ |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | $+2^{-15.26}$ |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | $+2^{-18.06}$ |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | $+2^{-15.99}$ |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | $+2^{-18.80}$ |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | $+2^{-15.99}$ |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | $+2^{-18.80}$ |
| … | … | … | … | … | … | … | … | … | … | … | … |
| * | * | 1 | * | * | * | * | * | * | * | * | 0 |
| * | * | * | * | * | * | * | * | 1 | * | * | 0 |

From the analysis presented in Section 2, we require at least $\Omega = \frac{\pi 2^{\beta+2}(m+1)\ln 2}{r\varepsilon^2}$ parity check equations to recover the correct LFSR's initial state with high probability. The probability of a wrong state considered as the correct initial state is smaller than $2^{-m-1}$. According to Theorem 3(Wang et al., 2019), the data and time complexity are separately $\overline{D} = k\times\frac{\pi 2^{\beta+2}(m+1)\ln 2}{r\varepsilon^2}$ and $\overline{T} = \frac{\pi k 2^{\beta+2}(m+1)\ln 2}{r\varepsilon^2} + rk2^{m+1-\beta}p$ in FCA

with single round sampling, where $p = Q((\pi r^{-1} 2^{\beta}(43+1)\ln 2)^{\frac{1}{2}})$, $k$ is period of round key function and $\beta$ represents the number of LFSR's initial state bits that can be omitted. $Q(x)$ is the tail distribution function of the standard normal distribution, i.e., $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\frac{y^2}{2}} dy$. In our multiple sampling FCA, assuming the number of sampling rounds is denoted as n. The data complexity of our attack is shown as

$$D = \frac{1}{n}\overline{D} = \frac{k}{n} \times \frac{\pi 2^{\beta+2}(m+1)\ln 2}{r\varepsilon^2} \tag{4.16}$$

Because our attack can use n keystream bits to construct the parity check equation in each round key cycle. In contract, single round sampling only uses one keystream bit per cycle, the number of keystream bits required for our attack is only 1/n of the single round sampling. The time complexity of our attack is shown as

$$T = \frac{2(n-1)+1}{n}\overline{T} = (2-\frac{1}{n}) \times (\frac{\pi k 2^{\beta+2}(m+1)\ln 2}{r\varepsilon^2} + rk2^{m+1-\beta} p) \tag{4.17}$$

The parity check equation required for our FCA with n rounds of sampling comprises n groups of parity check equations obtained from different rounds of sampling, and the number of equations in each group is $\frac{\Omega}{n}$. Considering that different groups of equations may have different round keys, we use the round key of the first group of equations as a reference and construct additional (n-1) groups of equations by performing the XOR operation between 1 and the remaining (n-1) groups of equations, to ensure the existence of $\Omega$ parity check equations with the same round key. Therefore, the time complexity of n round sampling should include calculation time for $\frac{2(n-1)+1}{n} \times \Omega$ equations compared to single round sampling.

The time and data complexity of our attack on Fruit-80 for different $\beta$ and n are presented below. The data provided in the corresponding positions of the following two tables denote the time and data complexity of the same attack.

**Table 2** Time complexity with different $n$ and $\beta$

| $n/\beta$ | 0 | 1 | 2 | 3 | Source |
|---|---|---|---|---|---|
| 1 | $\mathbf{2^{69.99}}$ | $2^{68.99}$ | $2^{67.98}$ | $2^{66.98}$ | (Wang et al. ,2016) |
| 2 | $2^{70.57}$ | $2^{69.57}$ | $2^{68.56}$ | $2^{67.56}$ | Our work |
| 3 | $2^{70.72}$ | $2^{69.72}$ | $2^{68.71}$ | $2^{67.71}$ | Our work |
| 4 | $2^{70.80}$ | $\mathbf{2^{69.80}}$ | $2^{68.79}$ | $2^{67.79}$ | Our work |

**Table 3** Data complexity with different $n$ and $\beta$

| $n/\beta$ | 0 | 1 | 2 | 3 | Source |
|---|---|---|---|---|---|
| 1 | $\mathbf{2^{56.82}}$ | $2^{57.82}$ | $2^{58.82}$ | $2^{59.82}$ | (Wang et al. ,2016) |
| 2 | $2^{55.82}$ | $2^{56.82}$ | $2^{57.82}$ | $2^{58.82}$ | Our work |
| 3 | $2^{55.23}$ | $2^{56.23}$ | $2^{57.23}$ | $2^{58.23}$ | Our work |
| 4 | $2^{54.82}$ | $\mathbf{2^{55.82}}$ | $2^{56.82}$ | $2^{57.82}$ | Our work |

In Table 2 and Table 3, the first rows of data are related to the time and data complexities of previous attacks (Wang et al., 2019), respectively. Other rows are our

improved attacks' results for different numbers of sampling rounds. We compared the results of our multiple sampling attacks with data in the first row; when the time complexity approximates $2^{70}$, the data complexity of our attack is half of the previous attack; the corresponding data in the table are marked in bold. Meanwhile, the minimum data complexity of our attack with 128 rounds of sampling can be reduced from $2^{56.82}$ to $2^{49.82}$ compared with the previous attack. When the available keystream bits are very limited, the reduction in the data complexity of the attack is more meaningful than the reduction in the time complexity. There are limitations over the number of produced keystream bits under a fixed key and IV on every stream cipher, and this point is more restricted in SSCs. Thus, our idea implements FCA over SSCs whenever enough keystream is unavailable. Meanwhile, our multiple sampling FCA can better balance time and data complexity by adjusting the number of sampling rounds and omitted states. This method can ensure that the lower time complexity of the attack can be obtained with enough keystream bits.

## 5 A countermeasure to strengthen SSCs

The fundamental requirement for implementing our improved attack is that the cipher's round key period is finite. This characteristic is typically found in most Grain-like SSCs. Therefore, SSCs with limited round key periods will lead to security vulnerabilities related to our multiple sampling FCA. To overcome this problem, we recommend increasing the round key function's period of SSCs

Due to the sampling interval of our attack for the parity check equations being equal to the period of the round key function, when the period of the round key function is large enough, more keystream bits are needed to implement this attack successfully. If the number of keystream bits used in the attack is more than that the cipher can provide, we believe that this attack is invalid for the cipher. We assume that the round key period of Fruit-80 is increased to t, and the data complexity of multiple sampling FCA is $D = \frac{t}{n} \times \frac{\pi 2^{\beta+2}(m+1)\ln 2}{r\varepsilon^2}$. For the number of omitted internal state bits $\beta$ and sampling rounds n of the attack are 0 and 3, if the period of the round key is more than $2^{32}$, the data complexity of multiple sampling FCA on Fruit-80 will increase to $2^{80}$, making the multiple sampling FCA ineffective.

Therefore, we propose a new design principle of SSCs that the round key function period should be set large enough. By observing the periods of the two round key functions of Fruit-80, we found that their periods are both 128, and their LCM is themselves, which leads to a very small total round key period. Therefore, we consider changing the period of one of the round key functions to increase the total round key period. For example, one of the round key functions selects 79 key bits in a loop. Although the period of the single round key function is reduced, the overall round key period is greatly increased because the LCM of 79 and 128 is 79 ×128. The other representative design idea originated from Fruit-F. Since the LFSR and NFSR of the ciphers have large periods, the internal state bits of them are used to update the round key function. The round key bits are obtained from several bits of the NFSR and the LFSR (Amin et al., 2023). For Fruit-80, the period of the LFSR is $2^{43}$-1, and the period of NFSR is the multiple of $2^{43}$-1. Thus, the period of the new round key function is at least $2^{43}$-1, which is much larger than the previous period. Since most stream ciphers contain the parts of LFSR and NFSR, this design idea can apply to other SSCs to

increase the round key function period, improving SSCs' security against multiple sampling FCA.

## 6 Conclusion

This paper proposed a new idea of employing multiple sampling for parity check equations in FCA, overcoming the SSCs' limitation on the number of output keystream bits. The proposed attack can use available keystream bits more efficiently and decrease the data complexity of the FCAs. Compared to the previous attack on Fruit-80, the suggested attack with four rounds of sampling reduced data complexity from $2^{56.82}$ to $2^{55.82}$ over the similar time complexity. Our improved FCA with 128 rounds of sampling can reduce the data complexity to $2^{49.82}$, which is the minimum data complexity of all attacks on Fruit-80. Meanwhile, our idea of multiple sampling can also be applied to other FCAs and SSCs. Furthermore, to ensure the SSCs' security, we suggested increasing the round key period of SSCs to resist the described attacks and improve ciphers' security.

## References


Amin-Ghafari, V., Hu, H. and Xie, C. (2016) 'Fruit-v2: ultra-lightweight stream cipher with shorter internal state', IACR Cryptology ePrint Archive 2016, https://eprint.iacr.org/2016/355.

Amin-Ghafari, V., Hu, H. (2018) 'Fruit-80: a secure ultra-lightweight stream cipher for constrained environments', Entropy, Vol. 20, No. 3, pp.180-189, doi: 10.3390/e20030180.

Amin-Ghafari, V., Lin, F., and Zhou, Z. (2023) 'A new idea in response to fast correlation attacks on small-state stream ciphers', Microprocessors and Microsystems, Vol.96, No. 10, pp.07-20, doi: 10.1016/j.micpro.2022.104720.

Armknecht, F., Mikhalev, V. (2015) 'On lightweight stream ciphers with shorter internal states', In Fast Software Encryption: 22nd International Workshop, FSE 2015, Vol.90, No. 54, pp. 451-470, doi: 10.1007/978-3-662-48116-5_22.

Babbage, S., Dodd, M. (2006) 'The stream cipher MICKEY 2.0. ECRYPT Stream Cipher', Estream Ecrypt Stream Cipher Project, pp.191-209.

Banik, S., Caforio, A., Isobe, T., Liu, F., Meier, W., Sakamoto, K., and Sarkar, S. (2021) 'Atom: a stream cipher with double key filter', IACR Transactions on Symmetric Cryptology, pp.5-36, doi: 10.46586/tosc.v2021.i1.5-36.

Canniere, C. (2006) 'Trivium: A stream cipher construction inspired by block cipher design principles', In International Conference on Information Security, pp.171-186.

Chose, P., Joux, A., and Mitton, M. (2002) 'Fast correlation attacks: An algorithmic point of view', International Conference on the Theory and Applications of Cryptographic Techniques, Vol.23, No.32, pp.209-221.

Esgin, M., Kara, O. (2016) 'Practical cryptanalysis of full Sprout with TMD tradeoff attacks', International Conference on Selected Areas in Cryptography, pp. 67-85.

Gren, M., Hell, M., Johansson, T., and Meier, W. (2011) 'Grain-128a: a new version of Grain-128 with optional authentication', International Journal of Wireless and

Mobile Computing, Vol.5, No.1, pp.48-59, doi: 10.1504/IJWMC.2011.044106.
Hell, M., Johansson, T., and Meier, W. (2007) 'Grain: a stream cipher for constrained environments', International journal of wireless and mobile computing, Vol.2, No.1, pp.86-93.
Hell, M., Johansson, T., Maximov, A., and Meier, W. (2006) 'A stream cipher proposal: Grain-128', IEEE International Symposium on Information Theory, doi: 10.1109/ISIT.2006.261549
Hellman, M. (1980) 'A cryptanalytic time-memory trade-off', IEEE transactions on Information Theory,Vol.26, No.4, pp.401-406, doi:10.1109/tit.1980.1056220.
Kara, O., Esgin, M. (2018) 'On analysis of lightweight stream ciphers with keyed update', IEEE Transactions on Computers, Vol.68, No.1, pp. 99-110.
Lallemand, V., Naya-Plasencia, M. (2015) 'Cryptanalysis of full Sprout', Advances in Cryptology, Vol.92, No.15, pp.663-682, doi: 10.1007/978-3-662-47989-6_32.
Meier, W., Staffelbach, O. (1989) 'Fast correlation attacks on certain stream ciphers', Journal of cryptology, Vol.68, No.1, pp.159-176.
Mikhalev, V., Armknecht, F., and Muller, C. (2016) 'On ciphers that continuously access the non-volatile key', IACR Transactions on Symmetric Cryptology, pp.52-79.
Siegenthaler, T. (1984) 'Correlation-immunity of nonlinear combining functions for cryptographic applications', IEEE Transactions on Information theory, Vol.30, No.5, pp.776-780.
Todo, Y., Isobe, T., Meier, W., Aoki, K., and Zhang, B. (2018) 'Fast correlation attack revisited: cryptanalysis on full grain-128a, grain-128, and grain-v1', Advances in Cryptology–CRYPTO 2018: 38th Annual International Cryptology Conference, Vol.109, No.92, pp. 129-159.
Wang, S., Liu, M., Lin, D., and Ma, L. (2019) 'Fast correlation attacks on grain-like small state stream ciphers and cryptanalysis of plantlet, Fruit-v2 and fruit-80', Journal of computer, Vol.66, No.6, pp.1376-1399, doi: 10.1093/comjnl/bxac016.
Zhang, B., Feng, D. (2006) 'Multi-pass fast correlation attack on stream ciphers', Journal of the Korea Institute of Information Security, Vol.43, No.56, pp.234-248.