An Attack on the LILLE Stream Cipher

Vahid Amin-Ghafari, Mohammad Ali Orumiehchiha, and Saeed Rostami

***Abstract*— A few small-state stream ciphers (SSCs) were proposed for constrained environments. All of the SSCs before the LILLE stream cipher suffered from distinguishing attacks and fast correlation attacks. The designers of LILLE claimed that it is based on the well-studied two-key Even-Mansour scheme and so is resistant to various types of attacks. This paper proposes a distinguishing attack on LILLE, the first attack since 2018. The data and time complexities to attack LILLE-40 are $2^{50.7}$ and $2^{41.2}$, respectively. We verified practically our attack on a halved version of LILLE-40. A countermeasure is suggested to strengthen LILLE against the proposed attack. We hope our attack opens the door to more cryptanalyses of LILLE.**

***Index Terms*— LILLE, stream cipher, lightweight encryption, distinguishing attack, time-memory-data trade-off attack, cryptography.**

I. Introduction

Stream ciphers, one of the main parts of symmetric encryption, play an important role in the information security industry. Thus, the security of stream ciphers should be carefully considered. The LILLE stream cipher was introduced as a robust lightweight cipher with the small-state stream cipher (SSC) concept in 2018 [1]. A few SSCs, such as Sprout [2], Fruit [3, 4], and Plantlet [5], were introduced before LILLE, but the designers of LILLE promised 80-bit security against distinguishing or key recovery attacks for LILLE. Time-memory-data trade-off (TMDTO) attacks and fast correlation attacks (FCA) were successfully applied to all SSCs published before LILLE [6-10]. Fruit-F was recently published, and the designers of Fruit-F claimed that it is resistant to FCA and TMDTO attacks [11].

The designers of LILLE claimed that the security of key/state recovery attacks on LILLE reduce to a two-key 6-round iterated Even-Mansour scheme, which is well-studied [12]. No attack has been presented to this cipher up until now. A distinguishing attack is introduced against LILLE in the current paper; that is the first attack.

A distinguishing attack is proposed based on the fact that the frequency of a special pattern in the keystreams of a cipher is different from that of truly random sequences, and this enables an attacker to distinguish between truly random sequences and keystreams of a stream cipher [13]. Distinguishing attacks are important and practical because identifying which sequences belong to a cipher (in massive data) is the first step for practical cryptanalysis. Furthermore, attacks (of any type) show weaknesses in the cipher and open the door to more attacks.

In the proposed distinguishing attack, an attacker should search for the same 40-bit. We will show that if the attacker has two tables of truly random 40-bit sequences and compares for collisions, she can find 2.34 collisions on average when she compares 600×2^{32} times. If the attacker does the same comparison on the specially selected keystreams of LILLE, she cannot find any collision. Thus, the attacker can win on the distinguishing attack and correctly recognize whether 40-bit sequences are from LILLE keystream or truly random sequences. The proposed attack is applicable on all versions of LILLE (i.e., LILLE-40, LILLE-60, and LILLE-80) with $2^{47.5}$ bits of the keystream, $2^{41.2}$ times comparing 40-bit sequences, and 20 gigabytes of memory.

We verified practically our attack on a halved version of LILLE-40, called *Shrunk LILLE* (see Appendix A for details). We will suggest our countermeasure to strengthen LILLE against the proposed attack.

The paper is organized as follows. Section 2 gives a brief description of the LILLE cipher. An observation on the internal state transition function of LILLE and a distinguishing attack on LILLE is presented in Section 3. Then, Section 4 presents a countermeasure to strengthen LILLE. Finally, Section 5 concludes the paper.

II. The LILLE Family of Stream Ciphers

LILLE is a stream cipher that accepts an 80-bit key and an IV to generate 40-bit keystream words. Key is divided into two parts, most significant (K_1) and least significant (K_2). The internal state of LILLE consists of 40-bit NFSR and L-bit LFSR. L introduces three versions of LILLE, i.e., LILLE-40, LILLE-60, and LILLE-80 for L equal to 40, 60, and 80 bits, respectively. For the initialization, all bits of NFSR (S_t) are equal to zeros, and all bits of LFSR are equal to zeros except the least significant bit, which is equal to one. Every 720 clocks, $ENC_{K_1,K_2,IV,L_r}(.)$ function accepts an 40-bit input and produces an 40-bit of keystream words as follows (initial value of X is $0x0000000000$). We refer to [1] for a full description.

This work has been submitted to the IEEE for possible publication.

Corresponding authors: Vahid Amin-Ghafari and Mohammad Ali Orumiehchiha.

Vahid Amin-Ghafari is with the School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, China (e-mail: vahidaming@cumt.edu.cn).

Mohammad Ali Orumiehchiha is with the Research Center for Developing Advanced Technology, Tehran, Iran (e-mail: orumiehchiha@rcdat.ir).

Saeed Rostami is with the Faculty of Engineering, University of Zanjan, Zanjan, Iran (e-mail: sae.rostami@gmail.com).

```
$ENC_{K_1,K_2,IV,L_r}(X)$:
        1-$X = X \oplus K_1$
        2-For $i = 0$ to 5 :
    If $i$ is even $RK = K_2$ else $RK = K_1$
    $X = P(X; IV; L_{r+120i}) \oplus RK$
        3-Output $X$
$P(S_0; IV; L_r)$:
        1-For $t = 0$ to 119 :
    $y_t = S_t[0] \oplus S_t[5] \oplus S_t[8] \oplus S_t[12] \oplus S_t[16] \oplus S_t[19]$
        $\oplus S_t[22] \oplus S_t[26] \oplus S_t[29] \oplus S_t[31]$
        $\oplus S_t[32] \oplus S_t[32] \cdot S_t[35] \oplus S_t[19] \cdot S_t[22]$
        $\oplus St[5] \cdot S_t[9] \oplus S_t[26] \cdot S_t[31] \cdot S_t[32]$
        $\oplus S_t[12] \cdot S_t[16] \cdot S_t[19] \oplus S_t[5] \cdot S_t[16]$
        $\cdot S_t[26] \cdot S_t[35] \oplus S_t[19] \cdot S_t[22] \cdot S_t[31]$
        $\cdot S_t[32] \oplus S_t[9] \cdot S_t[12] \cdot S_t[32] \cdot S_t[35]$
        $\oplus S_t[22] \cdot S_t[26] \cdot S_t[31] \cdot S_t[32] \cdot S_t[35]$
        $\oplus S_t[5] \cdot S_t[9] \cdot S_t[12] \cdot S_t[16] \cdot S_t[19]$
        $\oplus S_t[12] \cdot S_t[16] \cdot S_t[19] \cdot S_t[22] \cdot S_t[26]$
        $\cdot S_t[31] \oplus IV[t] \oplus L_{r+t}[0]$
    $S_{t+1} = S_t[1] \,||\, S_t[2] \,||\, \cdots \,||\, S_t[39] \,||\, y_t$
    $z_t = \begin{cases} L_{r+t}[0] \oplus L_{r+t}[5] \oplus L_{r+t}[15] \oplus L_{r+t}[20] \\ \quad \oplus L_{r+t}[25] \oplus L_{r+t}[34] \qquad for\ LILLE-40 \\ L_{r+t}[0] \oplus L_{r+t}[8] \oplus L_{r+t}[17] \oplus L_{r+t}[28] \\ \quad \oplus L_{r+t}[35] \oplus L_{r+t}[41] \qquad for\ LILLE-60 \\ L_{r+t}[0] \oplus L_{r+t}[13] \oplus L_{r+t}[23] \oplus L_{r+t}[38] \\ \quad \oplus L_{r+t}[51] \oplus L_{r+t}[62] \qquad for\ LILLE-80 \end{cases}$
    $L_{r+t+1} = \begin{cases} L_{r+t}[1] \,||\, \cdots \,||\, L_{r+t}[39] \,||\, z_t & for\ LILLE-40 \\ L_{r+t}[1] \,||\, \cdots \,||\, L_{r+t}[59] \,||\, z_t & for\ LILLE-60 \\ L_{r+t}[1] \,||\, \cdots \,||\, L_{r+t}[79] \,||\, z_t & for\ LILLE-80 \end{cases}$
        2-Output $S_{120}$
```

III. A Distinguishing Attack on LILLE

In this section, we explain how to apply a distinguishing attack on the LILLE. First, an observation on the internal state transition function of LILLE is presented. Second, the distinguishing attack is described based on the observation.

A. An Observation on the Internal State Transition Function of LILLE

As the designers of LILLE stated that the key-IV mixing algorithm is an injective function (one-to-one function) [1]. Also, the LFSR and NFSR functions of LILLE are injective[1], the internal state transition function of LILLE is injective. This means that two distinct internal states cannot produce the same internal state (Fig. 1). As keystream words (Z_i) are the values of NFSR after 120 clocks, the keystream generation algorithm is also injective. It means that if Z_1 and Z_c are equal, Z_2 and Z_{c+1} should be equal ($c \neq 1$). For the same reason, if Z_1 and Z_c are not equal, Z_2 and Z_{c+1} cannot be equal.

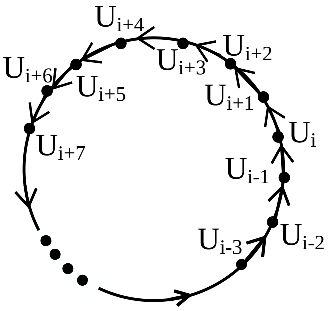

Fig. 1. Internal state transition of LILLE-40, U_j are internal states[2].

We verified the injectivity of the internal state transition function on Shrunk LILLE[3]. We produced keystream bits of Shrunk LILLE and saved 2^{13} 36-bit of them on a table. Then, 2^{13} 36-bit keystreams of Shrunk LILLE were produced (under the same key/IV) and searched for collisions[4]. As the internal state size of Shrunk LILLE is **32 bits**, if the internal state transition of Shrunk LILLE was not an injective function, the collisions should be based on the **32 bits** of internal state collision (not based on the collision of the **36-bit random** in 2^{26} searches). Our implementations showed that the probability of the collision is 0.00097, which is equal to the probability of finding two equal 36-bit random in 2^{26} searches. The probability of finding at least one collision in 2^{26} searches of 36-bit random is:

$$1-\left(1-\frac{1}{2^{36}}\right)^{2^{26}} = 0.00097. \qquad (1)$$

If the internal state transition was not injective, two different 32-bit internal states could arrive at the same internal state during forward clocking. Consequently, two different 32-bit internal states could produce the same keystreams if the internal state transition was not injective. In that situation, the probability of finding at least one collision in 2^{26} searches is[5]:

$$1-\left(1-\frac{1}{2^{32}}\right)^{2^{26}} = 0.015. \qquad (2)$$

Our implementation based on different internal state transition functions showed that the probability is 0.00097, and the function is injective.

[1] Because the NFSR function form is $S_{t+1}[39] = S_t[0] \oplus f(S_t[1], S_t[2], \ldots, S_t[39])$.

[2] We do not want to show that the internal state transition of LILLE-40 is necessarily a full period. Probably, the internal state transition contains more than one period.

[3] Shrunk LILLE has 16 bits LFSR and 16 bits NFSR, producing 16 keystream bits every 720 clocks.

[4] The total number of the search is $2^{13} \times 2^{13} = 2^{26}$.

[5] Note that the probability of equality of two random 32-bits is 2^{-32}.

B. A Distinguishing Attack on LILLE

This section proposes a distinguishing attack on LILLE-40, and then it is explained how to apply the distinguishing attack on LILLE-60 and LILLE-80. The designers of LILLE-40 have guaranteed that the period of the keystream sequences is at least $40 \times 48 \times (2^{40} - 1) = 2^{50.9}$ bits, and they recommend LILLE-40 can be used to encrypt 2^{50} bits using any pairs of key and IV [1]. As the period of LFSR is $(2^{40} - 1)$, and the *ENC* function needs 720 clocks to compute, the *ENC* function takes **the same initial state of LFSR** as its input after LCM[6]$(2^{40} - 1, 720)$ clocks (which is equal to $48 \times (2^{40} - 1)$ clocks[7]). As LILLE-40 produces every 720 clocks 40-bit of the keystream, we call $(48 \times (2^{40} - 1) \div 720) + 1$th word of keystream Z_d. Thus, the state of LFSR in Z_1 is equal to the state of LFSR[8] in Z_d. Moreover, the state of LFSR in Z_{1+i} is equal to the state of LFSR in Z_{d+i} for $i > 0$.

If we consider the keystream words of LILLE-40 on clocks Z_1 and Z_d under **the same key and IV**, the keystream words are equal together with probability 2^{-40} (This is because key, IV, and the internal states are the same on clocks Z_1 and Z_d except for 40 bits of NFSR). Thus, we have:

$$Probability\,(Z_{1+i} = Z_{d+i}) = 2^{-40} \quad for\ i \geq 0. \tag{3}$$

As the internal state transition function of LILLE-40 is an **injective function**[9], If we compare 2^{32} keystream words of LILLE-40 at the $d - 1$ clock intervals (i.e., Z_1 with Z_d, Z_2 with Z_{d+1}, ..., $Z_{2^{32}}$ with $Z_{d-1+2^{32}}$), the probability that we can find at least one equal keystream word of LILLE-40 is 2^{-40}. This is because if Z_1 is not equal to Z_d, Z_{1+j} will not be equal to Z_{d+j}, or in other words, the equality of two keystream words of LILLE-40 at the $d - 1$ clock interval is equivalent to the equality of all keystream words at the $d - 1$ clock intervals. Therefore, the probability of finding at least one collision during 2^{32} comparisons is 2^{-40}.

On another side if we compare 2^{32} **random words** that the size of each word is 40 bits, the probability of finding at least one collision during 2^{32} comparisons is:

$$1 - \left(1 - \frac{1}{2^{40}}\right)^{2^{32}} = 0.0039 = 2^{-8}. \tag{4}$$

Note that the probability of failure to find a collision in the first comparison is $1 - {}^{1}\!/_{2^{40}}$, and the probability of failure to find at least one collision in the 2^{32} comparisons is $\left(1 - \frac{1}{2^{40}}\right)^{2^{32}}$.

Therefore, our distinguishing attack works as follows:

1-We obtain two tables of 40-bit sequences from oracle (tables A and B); every table contains 2^{32} sequences[10].

2-We compare the first sequence of A with the first sequence of B to find collisions. Then we compare the second sequence of A with the second sequence of B to find collisions and continue to compare 2^{32} 40-bit sequences.

3-If we find one 40-bit from table A equal to a 40-bit of table B, we consider it a *WIN*.

4-We repeat this procedure 600 times[11].

5-If we have at least one *WIN*, 2^{32} 40-bit sequences are from a ***random*** sequence with high probability.

6-If we cannot find any *WIN*, 2^{32} 40-bit sequences are from the ***keystream*** of LILLE-40 with high probability.

According to Equation (4), we expect $600 \times 0.0039 = 2.34$ *WIN*s on average during our distinguishing attack on *random* sequences, while we expect $600 \times 2^{-40} = 2^{-30.7}$ *WIN*s on average for the *keystream* of LILLE-40. Thus, it is possible to distinguish between *random* sequences and *keystreams* of LILLE-40. The data complexity of the attack is $600 \times 40 \times (d - 1 + 2^{32}) = 24000 \times 2^{36.1} = 2^{50.7}$ bits (while only $600 \times 2^{32} \times 40 \times 2 = 2^{47.5}$ bits of the keystream is used during the attack). The time complexity is $600 \times 2^{32} = 2^{41.2}$ times comparing 40-bit sequences, and the memory complexity is $2^{32} \times 40$ bits or 20 gigabytes for distinguishing attack on LILLE-40. Note that Z_d is equal to $Z_{2^{36}}$, and the maximum required number of keystream bits ($Z_{d-1+2^{32}}$) is $2^{36.1}$ per key/IV of LILLE-40 (which is according to the legal usage of LILLE-40 as the designers of LILLE-40 stated that LILLE-40 can produce up to 2^{50} keystream bits per key/IV [1]).

The difference between the three variants of LILLE is back to their LFSRs. For LILLE-60, the LFSR length is 60 bits, and it is possible to apply a similar distinguishing attack on it. The period of LFSR is $(2^{60} - 1)$, and the LCM of $(2^{60} - 1, 720)$ equals $16 \times (2^{60} - 1)$. As LILLE-60 produces every 720 clocks 40-bit of the keystream, we call $(16 \times (2^{60} - 1) \div 720) + 1$th word of keystream $Z_{d'}$. If we compare 2^{32} keystream words of LILLE-60 at the $d' - 1$ clock intervals (i.e., Z_1 with $Z_{d'}$, Z_2 with $Z_{d'+1}$, ..., $Z_{2^{32}}$ with $Z_{d'-1+2^{32}}$), the probability that we can find at least one equal keystream word of LILLE-60 is 2^{-40}. Thus, we can apply a similar distinguishing attack on LILLE-60. The data complexity of the attack is $600 \times (d' - 1 + 2^{32}) \times 40 = 2^{69.1}$ bits (while only $600 \times 2^{32} \times 40 \times 2 = 2^{47.5}$ bits of the keystream is used during the attack). The time complexity is $600 \times 2^{32} = 2^{41.2}$ times comparing 40-bit sequences, and the memory complexity

[6] Least Common Multiple.

[7] In other words, the remainder of $(2^{40} - 1)$ divided by 720 is 15. It means that after 48 periods of LFSR, the *ENC* function will have the same initial state of LFSR as its input because of $48 \times 15 = 720$.

[8] It means that after 48 periods of LFSR, the LFSR which produce Z_d keystream word is equal to the LFSR of the first keystream word.

[9] In the previous section, we discussed that the internal state transition of LILLE-40 is an injective function.

[10] Tables A and B for LILLE-40 keystream should be prepared as follows: the first sequence of A and the first sequence of B are Z_1 and Z_d, the second sequence of A and the second sequence of B are Z_2 and Z_{d+1}, …, the last sequence of A and the last sequence of B are $Z_{2^{32}}$ and $Z_{d-1+2^{32}}$, respectively.

[11] It is clear that the probability of finding at least one collision during 2^{32} comparisons of LILLE-40 keystream words (i.e., 2^{-40}) is enough different from the probability of random words (i.e., 2^{-8}) for suggested distinguishing attack. However, 600 repeats are our suggestion based on our attack on Shrunk LILLE to achieve tangible results.

is $2^{32} \times 40$ bits or 20 gigabytes for distinguishing attack on LILLE-60. It is possible to apply a similar distinguishing attack on LILLE-80. The attack requires $600 \times 2^{32} \times 40 \times 2 = 2^{47.5}$ bits of the keystream and $600 \times 2^{32} = 2^{41.2}$ times comparing 40-bit sequences.

IV. A Countermeasure to Strengthen LILLE

The weakness of LILLE is to produce several keystream words from the same LFSR states under the same key and IV. For example, the state of LFSR while producing the first keystream word is the same as that of LFSR while producing the $(48 \times (2^{40} - 1) \div 720) + 1^{\text{th}}$ keystream word in LILLE-40. Thus, to prevent producing keystream words from the same LFSR states, the maximum number of keystream words should be $48 \times (2^{40} - 1) \div 720 = 2^{36}$ words. As every word of the keystream is 40 bits, LILLE-40 can produce $2^{41.3}$ bits safely. If the LILLE-40 produces more than 2^{36} keystream words under the same key and IV, it will be threatened by the proposed distinguishing attack. It seems that 2^{36} keystream words are too short for some applications. We suggest using $7 \times 98 = 686$ rounds instead of $6 \times 120 = 720$ rounds during the production of keystream words in LILLE-40. As the period of the LFSR is $(2^{40} - 1)$, and the *ENC* function needs 720 clocks to compute, LILLE-40 produces keystreams from the same state of LFSR after $\text{LCM}(2^{40} - 1, 720) = 48 \times (2^{40} - 1)$ clocks. If LILLE-40 is modified to use $7 \times 98 = 686$ rounds (instead of $6 \times 120 = 720$ rounds), it can clock $\text{LCM}(2^{40} - 1, 686) = 686 \times (2^{40} - 1)$ times before arriving at the same LFSR state to produce keystream words. It means that LILLE-40 can produce $686 \times (2^{40} - 1) \div 686 \approx 2^{40}$ keystream words or $2^{45.3}$ keystream bits safely under the same key and IV. In fact, the number of rounds should be chosen for LILLE so that they are coprime with the periods of LFSRs. For LILLE-40, 686 is coprime with $2^{40} - 1$. As $2^{60} - 1$ and $2^{80} - 1$ are coprime with 2, our suggestion for LILLE-60 and LILLE-80 is $8 \times 64 = 512$ rounds instead of $6 \times 120 = 720$ rounds. This modification not only does not impose overhead on cryptosystems but also increases keystream production speed. Note that reducing a few rounds of LILLE does not threaten the security of LILLE, according to the security analysis of LILLE. In addition, the designers of LILLE do not mention any clear reason for how to choose 720 rounds for it [1].

V. Conclusion

The designers of LILLE claimed that the security of LILLE reduces to a two-key 6-round iterated Even-Mansour scheme, and is resistant to various types of attacks. This paper presented the first attack on LILLE since 2018 when it was introduced as a robust small-state stream cipher. We proposed a distinguishing attack on LILLE and verified practically our attack on a halved version of LILLE-40. The proposed attack is applicable on all versions of LILLE with $2^{47.5}$ bits of the keystream, $2^{41.2}$ times comparing 40-bit sequences, and 20 gigabytes of memory. We hope this attack opens the door to other attacks on LILLE.

Appendix A
Shrunk LILLE

We verified practically our attack on a halved version of LILLE-40, called Shrunk LILLE. Shrunk LILLE has 16 bits LFSR and 16 bits NFSR, producing 16 keystream bits every 720 clocks. Its LFSR and NFSR are as follows:

$$\begin{aligned} y_t = {} & S_t\,[0] \oplus S_t\,[5] \oplus S_t[8] \oplus S_t[9] \oplus S_t[10] \oplus S_t[12] \\ & \oplus S_t[14] \oplus S_t[10] \cdot S_t[7] \oplus S_t[11] \\ & \cdot S_t[14] \oplus St\,[5] \cdot S_t[9] \oplus S_t[1] \cdot S_t[3] \\ & \oplus S_t[12] \cdot S_t[8] \oplus S_t[1] \cdot S_t[3] \\ & \cdot S_t[5] \oplus S_t[6] \cdot S_t[7] \cdot S_t[9] \oplus S_t[5] \\ & \cdot S_t[6] \cdot S_t[7] \cdot S_t[8] \oplus S_t[1] \cdot S_t[2] \\ & \cdot S_t[7] \cdot S_t[8] \cdot S_t[9] \cdot S_t[13] \oplus IV[t] \\ & \oplus L_{r+t}[0] \end{aligned}$$

$$S_{t+1} = S_t\,[1] \,||\, S_t\,[2] \,||\, \cdots \,||\, S_t\,[39] \,||\, y_t$$

$$z_t = L_{r+t}\,[0] \oplus L_{r+t}[1] \oplus L_{r+t}[3] \oplus L_{r+t}\,[12]$$

$$L_{r+t+1} = L_{r+t}\,[1] \,||\, \cdots \,||\, L_{r+t}\,[15] \,||\, z_t.$$

Shrunk LILLE accepts an 32-bit key and an 80-bit IV to generate 16-bit keystream words. Key is divided into two parts, most significant (K_1) and least significant (K_2). For the initialization, all bits of NFSR (S_t) are equal to zeros, and all bits of LFSR are equal to zeros except the least significant bit, which is equal to one. Every 720 clocks, $ENC_{K_1,K_2,IV,L_r}(.)$ function accepts an 16-bit and produces an 16-bit of keystream words, same as LILLE-40.

References

[1] S. Banik, T. Isobe, and M. Morii, “On design of robust lightweight stream cipher with short internal state,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences,* vol. 101, no. 1, pp. 99-109, 2018.

[2] Armknecht, and V. Mikhalev, “On lightweight stream ciphers with shorter internal states,” *International Workshop on Fast Software Encryption*. Springer, Berlin, Heidelberg, 2015, pp. 451-470.

[3] V. A. Ghafari, H. Hu, and Y. Chen, “Fruit-v2: ultra-lightweight stream cipher with shorter internal state,” *Cryptology ePrint Archive,* vol. 2016, pp. 355, 2016.

[4] V. A. Ghafari, and H. Hu, “Fruit-80: A Secure Ultra-Lightweight Stream Cipher for Constrained Environments,” *Entropy,* vol. 20, no. 3, pp. 180, 2018.

[5] V. Mikhalev, F. Armknecht, and C. Müller, “On ciphers that continuously access the non-volatile key,” *IACR Transactions on Symmetric Cryptology,* vol. 2016, no. 2, pp. 52-79, 2017.

[6] M. Hamann, M. Krause, W. Meier, and B. Zhang, “Design and analysis of small-state grain-like stream ciphers,” *Cryptography and Communications,* vol. 10, no. 5, pp. 803-834, 2018.

[7] Y. Todo, W. Meier, and K. Aoki, “On the Data Limitation of Small-State Stream Ciphers: Correlation Attacks on Fruit-80 and Plantlet,” *International Conference on Selected Areas in Cryptography (SAC*), Springer, Cham, 2019, pp. 365-392.

[8] S. Wang, M. Liu, D. Lin, and L. Ma, “On Grain-Like Small State Stream Ciphers Against Fast Correlation Attacks: Cryptanalysis of Plantlet, Fruit-v2 and Fruit-80,” *The Computer Journal*, 2022.

[9] B. Zhang, X. Gong, and W. Meier, “Fast Correlation Attacks on Grain-like Small State Stream Ciphers,” *IACR Transactions on Symmetric Cryptology,* vol. 2017, no. 4, pp. 58-81, 2017.

[10]B. Zhang, and X. Gong, “Another tradeoff attack on sprout-like stream ciphers,” *International Conference on the Theory and Application of Cryptology and Information Security*, Springer, Berlin, Heidelberg, 2015 pp. 561-585.

[11]V. A. Ghafari, F. Lin, and Z. Zhou, “A new idea in response to fast correlation attacks on small-state stream ciphers,” *Microprocessors and Microsystems*, pp. 104720, 2022.

[12] S. Even, and Y. Mansour, “A construction of a cipher from a single pseudorandom permutation” *International Conference on the Theory and Application of Cryptology*, Springer, Berlin, Heidelberg, 1993, pp. 210-224.

[13] V. A. Ghafari, and H. Hu, “A new chosen IV statistical distinguishing framework to attack symmetric ciphers, and its application to ACORN-v3 and Grain-128a,” *Journal of Ambient Intelligence and Humanized Computing,* vol. 10, no. 6, pp. 2393-2400, 2019.