PAPER

On Design of Robust Lightweight Stream Cipher with Short Internal State

Subhadeep BANIK[†], **Takanori ISOBE**[††], ***Nonmembers, and*** **Masakatu MORII**[†††], ***Senior Member***

SUMMARY The stream cipher Sprout with a short internal state was proposed in FSE 2015. Although the construction guaranteed resistance to generic Time Memory Data Tradeoff attacks, there were some weaknesses in the design and the cipher was completely broken. In this paper we propose a family of stream ciphers LILLE in which the size of the internal state is half the size of the secret key. Our main goal is to develop *robust* lightweight stream cipher. To achieve it, our cipher based on the two-key Even Mansour construction and thus its security against key/state recovery attacks reduces to a well analyzed problem. We also prove that like Sprout, the construction is resistant to generic Time Memory Data Tradeoff attacks. Unlike Sprout, the construction of the cipher guarantees that there are no weak key-IV pairs which produce a keystream sequence with short period or which make the algebraic structure of the cipher weaker and easy to cryptanalyze. The reference implementations of all members of the LILLE family with standard cell libraries based on the STM 90nm and 65 nm processes were also found to be smaller than Grain v1 while security of LILLE family depend on reliable problem in the symmetric cryptography.

key words: *Even-Mansour, lightweight, stream cipher, short internal state, TMD tradeoff.*

1. Introduction

1.1 Background

In the past few years, lightweight cryptography has become a very active research field, with a large deployment of network devices requiring security including resource-constrained devices such as sensor nodes and RFID tags. As a result, we have seen a number of candidates of lightweight primitives, e.g. block ciphers: PRESENT [14], KATAN [17], LED [28], Piccolo [44], TWINE [45], Midori [5], and PRINCE [15]. In any lightweight implementation, the register size consumes the most significant percentage of the total area. A stream cipher requires an internal state twice the size of the keylength to prevent the Biryukov-Shamir Time-Memory-Data Tradeoff attack [13]. This is one of the reasons why block ciphers are common in the lightweight cryptography world. However, due to the birthday limitation, it is undesirable to encrypt data of more than 2^{32} blocks (around few Gigabytes), with a key and IV pair in the CBC mode and a key in the counter mode. It is an undesirable property in real-world applications. On the other hand, a stream cipher does not have such a birthday limitation.

At FSE 2015, Armknecht and Mikhalev showed that even with a state smaller than twice the keylength, one can avoid the Biryukov-Shamir TMD attack. This was done by inserting some key information into the keystream generation function [1]. As an example, a stream cipher called Sprout was presented. Unfortunately, a lot of attacks immediately broke this construction. A state recovery attack based on a guess and determine attack was first proposed in [39]. This attack was faster than exhaustive search by a factor of about 2^{10} and had a memory complexity of 2^{46} bits. In [26], a TMD tradeoff attack was outlined using an online time complexity of 2^{33} encryptions and 770 TB of memory. The paper first observed that it was easy to deduce the secret key from the knowledge of the internal state and the keystream. The paper then made an observation on special states of Sprout that produced keystream without the involvement of the secret key. The pre-computation stage outlined a method to generate and store such states in tables. The online stage consisted of inspecting keystream bits, retrieving the corresponding state from the table, assuming of course that the state in question was a special state, and then computing the secret key. The process, if repeated a certain number of times, guaranteed that a special state would be encountered, from where the correct secret key was found. In [3], the author used a slide attack technique to distinguish the keystream produced by a single key and multiple randomly chosen IVs. The attack required 2^{40} keystream sequences and a memory of around 2^{48} bits. In the same paper the author showed that there were around 2^{70} key-IV pairs in Sprout that produced keystream of period 80. Furthermore, the author showed that for every key there existed around 2^{30} IVs for which the linear register used in Sprout became all zero after the key-IV setup phase. This weakened the algebraic structure of the cipher considerably and the author was able to perform a state recovery attack in the multiple IV setting using time equivalent to $2^{66.7}$ encryptions and negligible memory. Another TMD tradeoff attack using the normality of the output function was presented in [47]. Another state recovery attack using the Fibonacci to Galois transformation of shift registers was proposed in [40].

1.2 Motivation and Design Goal

After Sprout was broken, a very relevant question remained whether it was possible to construct ciphers with short internal states. Furthermore, another question was whether it was possible to guarantee the security of such ciphers against generic attacks. In this respect, a very natural tradeoff is be-

[†]The author is with the LASEC, École Polytechnique Fédérale de Lausanne, Switzerland.

[††]The author is with the Graduate School of Applied Informatics, University of Hyogo, Japan.

[†††]The author is with the Kobe University, Japan.

DOI: 10.1587/trans.E0.??.1

tween throughput and area. Shorter internal state usually implies that an attacker has to spend lesser effort in order to cryptanalyze the cipher via a state recovery attack. On the other hand, lower throughput implies that lesser information is made available to the attacker to perform any kind of cryptanalysis. It seems only natural that a cipher with shorter internal state, must in some sense compensate by lowering its throughput, in order to maintain the security margins. Such ciphers with small states are particularly useful in constrained environments like RFID tags where the major optimization thrust goes into lowering the area and power, even at the expense of throughput. Our second motivation was to find a lightweight solution for encrypting long messages. Most lightweight block ciphers have a blocksize of 64 bits, and due to the birthday constraint they cannot encrypt messages longer than 2^{32} blocks. Basically, IoT applications which do not require real-time operations between edge devices and a cloud server are our target. In this case, the edge device does sensing the environment including temperature, the pulse, locations, or records picture/movie information of surveillance camera, and send it to cloud server at fixed time, and the server gathers information from these from devices and analyzes it. In these applications, since a real-time processing is not required, throughput is not problem. It is a common use case of IoT.

The challenge therefore is to find a reliable lightweight stream cipher construction, secure against both generic and recently proposed attacks, while keeping the small state size requirement, and deliver a solution for encrypting a long message beyond 2^{32} blocks to lightweight cryptography field. Therefore, the design goals of the target stream cipher with respect to security and implementation targets are as follows.

Security

- Security against key and state recovery attacks that reduces to the security of a well-analyzed problem.
- Demonstrable security against all published Time-Memory-Data Tradeoff attacks [13, 25, 33].
- Guarantee a long period beyond 2^{32} blocks.

Implementation

- Smallest among *secure* stream ciphers with respect to the area, even if it requires lowering of throughput.

1.3 Our Contribution

In this paper we propose a family of stream ciphers LILLE, that has the smallest area among all secure stream ciphers offering 80 bit security. The cipher is based on the well known two-key Even Mansour construction. The design is put together in such a manner that no weak key-IV pairs exist, and given any key-IV pair we can guarantee a minimum period for the keystream produced by it. We demonstrate security of the cipher against a number of popular attack paradigms. Finally we present implementation results for the family of ciphers in ASIC using the standard cell libraries based on the STM 65nm and 90nm logic processes. We prove that all the three versions of the cipher occupy lesser area than 80-bit stream ciphers like Grain v1 [31] and Trivium [19].

1.4 Related work

1.4.1 Comparison with other lightweight stream ciphers:

Comparing with other stream ciphers offering 80 bit security, all the members of the LILLE family are smaller than Grain v1 and Trivium. In fact the size of all the members of this family are at least 1.5 times less than Trivium. It is slightly larger than Sprout, but since the cipher has been broken, we do not consider a direct comparison with it. For both Grain v1 and Sprout there exist a class of weak key-IV pairs [3, 48] that lead the LFSR in the design to the all zero state after the Key-IV setup phase. This not only causes the ciphers to produce keystream of lower period, but it also weakens their algebraic structure. In Sprout, this leads to a state recovery attack [3] in time equivalent to $2^{66.7}$ encryptions. In Grain v1, this leads to a distinguishing attack using $2^{44.2}$ keystream bits [48]. In LILLE, these drawbacks are absent due to the fact that the LFSR is initialized with a constant state that never enters the degenerate all zero state. Thus not only does the cipher guarantee a minimum period for the keystream sequence, there are also no weak key-IV pairs that weakens the algebraic structure of the cipher.

Very recently two lightweight stream ciphers: Lizard [29] and Plantlet [41] have been proposed. Plantlet suffers from the same distinguishing attack proposed in [3] in context of the stream cipher Sprout, and the authors of Plantlet admit this in [41, Section 5]. The authors of Lizard stipulate that no more than 2^{18} bits of keystream can be generated from a single Key-IV pair. This is not desirable for applications that need to encrypt large data streams. Furthermore, Lizard suffers from IV collision issues: it is possible to find a key for which there exist two different IVs which produce identical keystream segments. The authors of Lizard admit that such an IV collision can be found in $2^{60.5}$ queries. The authors also show a TMD Tradeoff Distinguisher that requires 2^{43} random IV queries. Lastly, it is also possible to find two different Key-IV pairs that generate the same keystream segment in only about 2^{28} queries. LILLE is free from such issues.

1.4.2 Comparison with block ciphers in CBC and counter mode:

Our construction looks like a combination of CBC mode and counter mode of block ciphers. As mentioned before, due to the birthday limitation, it is undesirable to encrypt data of more than 2^{32} blocks using 64-bit block cipher. On the other hand, our construction guarantees long message encryption by properly choosing the size of LFSR. Furthermore, the counter mode additionally requires the cost of a counter in addition to the area of the block cipher. Thus, we compare the gate size of our stream cipher with a lightweight block

cipher including a counter of appropriate length.

Due to birthday limitations, we can not encrypt more than $2^{N/2}$ blocks with a block cipher of blocksize N using conventional modes of operations like CBC and counter mode. Thus the maximum data an N bit block cipher can encrypt is $N2^{N/2}$. For $N = 64$, this is only around 2^{38} bits, so for larger dataset encryption we should make a comparison with ciphers with blocksize 96 or 128. Furthermore, the counter mode would require either a decimal counter or LFSR of $N/2$ bits. The cost of a 64-bit LFSR is estimated at around 300 GE (GE or gate equivalents is the area occupied by an equivalent number of 2-input NAND gates). The hardware cost of serialized implementations of well-known 128-bit block ciphers is around 2060 GE (AES 128), 1234 GE (Simon 128/128), 1280 GE (Speck 128/128) [4, 9]. Of block ciphers with 96 bit block size we have 955 GE (Simon 96/96), 1012 GE (Speck 96/96). As we will further see in Section 5.3, our construction is sufficiently smaller than the block cipher with counter modes.

There is no straightforward method to make a comaparison with the CBC mode of block ciphers. To implement any block cipher in any particular mode of operation like CBC would require some overhead (typically around 50 GE or more/less) and it depends on the particular architecture of the cipher in question. We can only remark that our construction is less in area than the core circuit of the 96/128 bit block cipher required to encrypt an equal amount of data.

2. Specification

The LILLE stream cipher has an 80 bit Secret Key and an 80 bit IV[†]. The most significant and least significant 40 bits of the Secret Key are denoted as K_1 and K_2 respectively. The cipher has a 40 bit internal state register and an ℓ bit Linear Feedback Shift Register (LFSR) of maximum period. We specify three variants of the cipher LILLE-40, LILLE-60 and LILLE-80 for which $\ell = 40, 60, 80$ respectively. The principal module in the LILLE stream cipher family is the $ENC_{K_1,K_2,IV,L_r}(\cdot)$ module that is a permutation on $\{0,1\}^{40} \rightarrow \{0,1\}^{40}$. The function is indexed by K_1, K_2, IV and L_r the current ℓ bit state of the LFSR.

In LILLE-40, the LFSR is initialized to the 40-bit state L_0 =`0x0000000001`. For LILLE-60, LILLE-80, the initial values are `0x000000000000001` and `0x0000000000000000001` respectively. The state is initialized to the 40-bit all zero vector. The keystream is produced due to the following rule (the process is described in Figure 1):

1. $r = 0$, $IV = IV|$`0x0000000000` (zero padded to 120 bits).
2. $L_0 = \begin{cases} \texttt{0x0000000001}, & \text{for LILLE-40,} \\ \texttt{0x000000000000001}, & \text{for LILLE-60,} \\ \texttt{0x00000000000000000001}, & \text{for LILLE-80.} \end{cases}$
3. Z_0 =`0x0000000000`.
4. **While** Keystream is required

[†]Although we use the term IV (Initial Vector) to denote this vector, it is used throughout in the encryption phases.

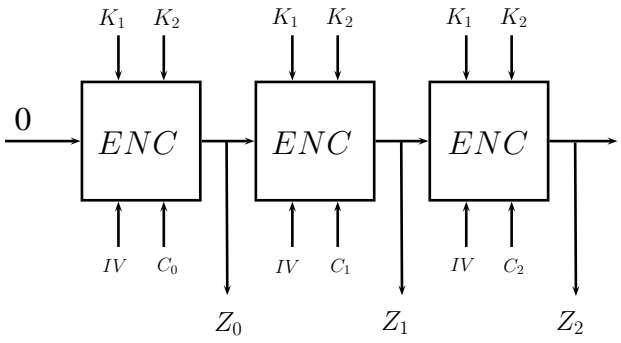


Fig. 1 Description of the LILLE family of stream ciphers. Note that $C_r = L_{720r}$, i.e. the $720r^{th}$ LFSR state.

- $Z_{r+1} = ENC_{K_1,K_2,IV,L_{720r}}(Z_r)$
- $r \leftarrow r + 1$

The successive 40 bit output values of the ENC module $Z_1, Z_2, Z_3 \ldots$ form the keystream sequence. The structure of the ENC module is given in Figure 2. We also provide a pseudocode as follows:

$\underline{ENC_{K_1,K_2,IV,L_r}(X)}$

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.

The module essentially alternates addition of K_1, K_2 to the current state in between the application of 6 instances of the permutation P. As we will see shortly, the permutation P takes as input the IV, and the current LFSR state L_r, the initial 40 bit state S_0 and outputs the 40 bit state S_{120} after 120 iterations. The permutation P and the LFSR are both implemented as shift registers. The functional description of the Permutation P is given as follows (note that $X[u]$ denotes the u^{th} bit of X, and || denotes the concatenation operation):

$\underline{P(S_0, IV, L_r)}$

1. For $t =$ 0 to 119

$$\begin{aligned} y_t = {} & S_t[0] \oplus S_t[5] \oplus S_t[8] \oplus S_t[12] \oplus S_t[16] \oplus S_t[19] \oplus S_t[22] \oplus \\ & S_t[26] \oplus S_t[29] \oplus S_t[31] \oplus S_t[32] \oplus S_t[32] \cdot S_t[35] \oplus \\ & S_t[19] \cdot S_t[22] \oplus S_t[5] \cdot S_t[9] \oplus S_t[26] \cdot S_t[31] \cdot S_t[32] \oplus \\ & S_t[12] \cdot S_t[16] \cdot S_t[19] \oplus S_t[5] \cdot S_t[16] \cdot S_t[26] \cdot S_t[35] \oplus \\ & S_t[19] \cdot S_t[22] \cdot S_t[31] \cdot S_t[32] \oplus S_t[9] \cdot S_t[12] \cdot S_t[32] \cdot \\ & S_t[35] \oplus S_t[22] \cdot S_t[26] \cdot S_t[31] \cdot S_t[32] \cdot S_t[35] \oplus \\ & S_t[5] \cdot S_t[9] \cdot S_t[12] \cdot S_t[16] \cdot S_t[19] \oplus S_t[12] \cdot S_t[16] \\ & \cdot S_t[19] \cdot S_t[22] \cdot S_t[26] \cdot S_t[31] \oplus IV[t] \oplus L_{r+t}[0] \end{aligned}$$

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

2. Output S_{120}.

In addition, the permutation P uses $L_{r+t}[0]$ which is the 0^{th} bit of the updated LFSR state L_{r+t} in each iteration. The LFSR state is updated in iteration (for all $t \geq 0$) as follows:

$$l_t = \begin{cases} L_{r+t}[0] \oplus L_{r+t}[5] \oplus L_{r+t}[15] \oplus L_{r+t}[20] \\ \oplus L_{r+t}[25] \oplus L_{r+t}[34], \quad \text{for LILLE-40,} \\ \\ L_{r+t}[0] \oplus L_{r+t}[8] \oplus L_{r+t}[17] \oplus L_{r+t}[28] \\ \oplus L_{r+t}[35] \oplus L_{r+t}[41], \quad \text{for LILLE-60,} \\ \\ L_{r+t}[0] \oplus L_{r+t}[13] \oplus L_{r+t}[23] \oplus L_{r+t}[38] \\ \oplus L_{r+t}[51] \oplus L_{r+t}[62], \quad \text{for LILLE-80.} \end{cases}$$

$$L_{r+t+1} = \begin{cases} L_{r+t}[1] \,||\, \cdots \,||\, L_{r+t}[39] \,||\, l_t, & \text{for LILLE-40,} \\ L_{r+t}[1] \,||\, \cdots \,||\, L_{r+t}[59] \,||\, l_t, & \text{for LILLE-60,} \\ L_{r+t}[1] \,||\, \cdots \,||\, L_{r+t}[79] \,||\, l_t, & \text{for LILLE-80.} \end{cases}$$

Since each permutation is computed in 120 clock cycles, the ENC module takes 720 rounds to compute. Also note that as given in Figure 2, the input LFSR states to the successive instances of P are $L_r, L_{r+120}, \ldots$.

3. Design Decision

We explain our design decisions vis-a-vis the design goals of Section 1.

3.1 Generic Construction:

- $ENC_{K_1,K_2,IV,L_r}(\cdot)$ is regarded as a 6-round iterated Even-Mansour with two alternating keys [27] for a fixed IV. The LILLE stream cipher produces a 40-bit keystream after a call of $ENC_{K_1,K_2,IV,L_r}(\cdot)$. This situation corresponds to the known plaintext setting in a block cipher. Thus, we can claim that *the securities of key/state recovery attacks reduces to a 6-round iterated Even-Mansour with two alternating keys in the known plaintext setting*.
- The basic component of the stream cipher is the $ENC_{K_1,K_2,IV,L_r}(\cdot)$ module which changes depending on the current LFSR state for a Key and IV. The cipher can guarantee a minimum period for the keystream due to the size of LFSR. As we will see shortly, a larger LFSR guarantees a larger period for the keystream sequence produced by the cipher. Hence, depending on user requirements, one can choose the LFSR as per the required period of a keystream.
- The internal permutation $ENC_{K_1,K_2,IV,L_r}(\cdot)$ takes both key and IV as inputs. This prevents generic TMD tradeoff attacks. We will describe the details in the next section.

3.2 Underlying Permutation:

The update function used in LILLE, is the same as the function f used in the hash function QUARK [2]. The function is of 13 variables, has a non-linearity of 3440, and algebraic degree 6, and is 3-resilient. One of the reasons that a function of high algebraic degree was chosen, was to thwart any algebraic advance via chosen IV attacks like cube attacks, dynamic cube attacks and conditional differential attacks. It is expected that after the 720 rounds of the ENC module the algebraic degree of the output bits would be close to 80. The function is surprisingly lightweight and occupies only around 52 GE using the standard cell library based on the STM 90nm process.

4. Security Analysis

We analyze the security of the LILLE stream cipher with respect to several attacks. First, we evaluate the generic construction assuming the underlying function $ENC_{K_1,K_2,IV,L_r}(\cdot)$ is a pseudo random function. Then, we evaluate the the underlying function $ENC_{K_1,K_2,IV,L_r}(\cdot)$.

4.1 General Construction

4.1.1 Key/State recovery attacks

Despite considerable cryptanalytic efforts over past twenty years, there is no efficient generic attacks on the more than 5-round iterated Even-Mansour with two alternating keys [23,24,34,42]. However, as mentioned in [24], there are polynomial-time advantage attacks on up to 8-round which improve over exhaustive search by a relatively-small factor [23]. If the user would like to also avoid this type of the attack, he has only to use a 10-round iterated Even-Mansour with two alternating keys as $ENC_{K_1,K_2,IV,L_r}(\cdot)$. Importantly, even if increasing the number of rounds, the additional cost is negligible.

Next, let us consider a multiple IV attack where the adversary is able to get keystream bits generated by different IVs. If the IV is different, an internal permutation $ENC_{K_1,K_2,IV,L_r}(\cdot)$ become a distinct one. Thus, in this case, the adversary has to attack different block ciphers at the same time, i.e. different 6-round iterated Even-Mansour with two alternating keys. Therefore, even in this case, there is no advantage over the single IV case.

4.1.2 Distinguishing attacks:

$ENC_{K_1,K_2,IV,L_r}(\cdot)$ takes different round constants from LFSR. Thus each of $ENC_{K_1,K_2,IV,L_r}(\cdot)$ is assumed to be independent PRP for a fixed Key-IV pair. Assuming that the input of each $ENC_{K_1,K_2,IV,L_r}(\cdot)$ is uniformly-distributed, a keystream block can be regarded as a pseudo random string.

4.1.3 Key and IV collision attack:

If either the key or IV has a difference δ, then the same difference δ is inserted into all $ENC_{K_1,K_2,IV,L_r}(\cdot)$ modules. Here, each of $ENC_{K_1,K_2,IV,L_r}(\cdot)$ is a independent function due to round constants. The probability that same differences are canceled out inside of all $ENC_{K_1,K_2,IV,L_r}(\cdot)$ at the same

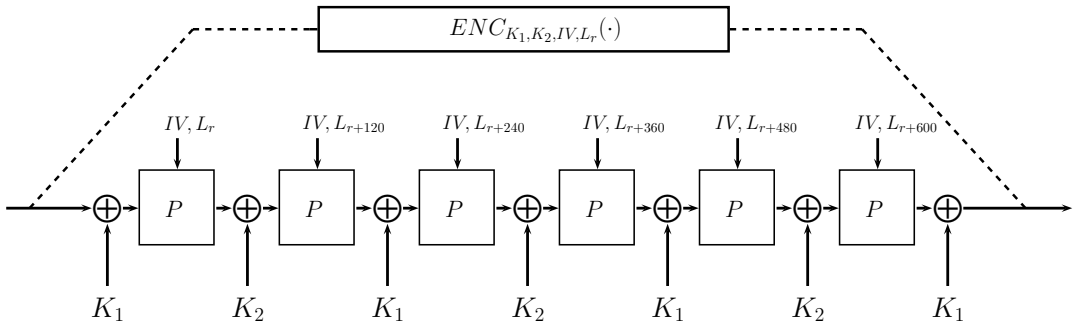


Fig. 2 Description of the $ENC_{K_1,K_2,IV,L_r}(\cdot)$ function

time is extremely low.

4.1.4 TMD TradeOff attacks

We will demonstrate that the LILLE stream cipher is secure against generic Time Memory Data (TMD) Tradeoff attacks. To do so let us recall three generic TMD attacks present in literature.

1. Biryukov-Shamir Attack [13] TMD tradeoff attacks aim to invert a one way function f at a single point in the range of function. The attack is probabilistic and the attacker may need access to to multiple points in the range of f. For stream ciphers, the one way function is typically the map between the internal state and the prefix of the keystream bits produced by the internal state. The attack outlined in [13] can be described thus:

- **a.** Let N denote the size of the set of internal states. The attacker chooses m, t, D so that $mt^2 = N$ and $t \geq D$.
- **b.** The attacker builds $\frac{t}{D}$ tables of size $m \times t$ in the following manner: he randomly chooses m initial states. For each initial state, he forms a chain of length t by iteratively applying the stream cipher function f and using the keystream as the state for the next point. For each table some unique reordering of the bits after applying the function f is used so that the tables do not store the same set of states.
- **c.** In the process, $mt \cdot \frac{t}{D} = \frac{N}{D}$ of the state space is covered by all the chains. This also happens to be the offline complexity of this stage. Also only the start and endpoints of each chain are stored in tables, and so $M = m \cdot \frac{t}{D}$ bits of memory is used.
- **d.** In the online phase, the attacker has access to D segments of keystream. For each target keystream segment y, he applies f on y upto t times checks if y is present as an endpoint in any table. If yes, he goes back to the starting point and retrieves the state just before y in the chain. The total time complexity is thus $T = D \cdot t \cdot \frac{t}{D} = t^2$. This gives us the tradeoff curve $TM^2D^2 = N^2$, with the limitation that $T \geq D^2$.

We claim that the above attack can not be applied on LILLE. First note that without the key, IV and current LFSR state a function can not map the internal state to the keystream. Hence the effective internal state of the cipher consists of not only the 40 bit internal state, but also the 80 bit secret key, 80 bit IV and the ℓ-bit LFSR state. Therefore, the effective value of $N = 2^{40+80+80+\ell} = 2^{200+\ell}$. This value is larger than 2^{240} for all three versions of LILLE. Considering the tradeoff curve $TM^2D^2 = N^2$, and the offline complexity $\mathcal{P} = \frac{N}{D}$ it is impossible to have both $\mathcal{P}$ and T less than 2^{80}.

2. Hong-Sarkar Attack [33] This attack is exactly the same as the Biryukov-Shamir attack, except that the definition of the underlying one way function is now changed. In this attack f maps the string consisting of the Key and IV to an equal length keystream bits. Thus if K and V refer to the size of the Key space and IV space respectively, then $N = KV$, and we will have the new tradeoff curve $TM^2D^2 = K^2V^2$ with the limitation that $T \geq D^2$. This attack becomes applicable if $V << K$, as in the case of the A5/3 cipher (in which the size of the secret key in 64 bits, and the size of the IV is 22 bits). In our case $K = V$, and $N = KV = 2^{160}$, and so as per the analysis presented in [35, Proposition 1] so this attack is again not feasible.

3. Dunkelman-Keller Attack [25] The Dunkelman-Keller TMD attack is a multiple IV attack, i.e. the attacker obtains keystream bits from multiple IVs and the same Key in order to perform the attack. The definition of the underlying one way function f is slightly different from the Hong-Sarkar attack. Given a fixed IV, the function f maps the secret key to the keystream sequence of equal length. The attacker chooses $\frac{V}{D}$ random IVs. For each IV he constructs t tables as before, by iterative application of the function f from m random starting points, with $mt^2 = K$. Again only the start and end points are stored and so for each IV the storage required is $M_{single} = mt$, and the total storage is therefore $M_{single} \cdot \frac{V}{D}$, and the total offline complexity $\mathcal{P} = K \cdot \frac{V}{D}$.

In the online phase, the attacker waits until he receives keystream for one of the $\frac{V}{D}$ IVs he had made tables for. This happens in roughly D IV resynchronizations. Once he gets such keystream from such an IV, he retrieves the t tables he had constructed for the particular IV and tries to find the inverse image of the keystream string in each of the tables. Therefore the online complexity is given by

$T = D + t^2 = D + \frac{K^2}{M^2_{single}} = D + \frac{K^2V^2}{M^2D^2}$ with the constraints $T \geq D, V \geq D$. In the case of LILLE, $KV = 2^{160}$ and again as per the analysis presented in [35, Proposition 1] so this attack is not feasible.

4. Esgin-Kara Attack [26] This attack is specific to the Sprout stream cipher. The authors take advantage of the non-linear key mixing in Sprout to enumerate a special class of internal states which produce keystream for around 40 cycles without any contribution from the secret key. All such states along with the produced keystream bits are listed in tables. The online stage consists of inspecting keystream bits, retrieving the corresponding state from the table, assuming of course that the state in question is a special state, and then computing the secret key. The process, if repeated a certain number of times, guarantees that a special state is encountered, from where the correct secret key is found. LILLE is immune from this style of attack primarily because it does not employ non-linear key mixing.

5. Zhang-Gong Attack [47] This attack is again specific to the Sprout stream cipher, and somewhat similar in structure to the Esgin-Kara attack, in so much that it takes advantage of the non-linear key mixing in Sprout to find special states that produce keystream without directly involving the secret key. The attack additionally uses a property of Boolean functions called k-normality. A function is called k-normal if it is constant over a k-dimensional subspace of its input variables. Using the normality of the output function used in Sprout, the attack further refines the definition of special states to mean those for which the output function is evaluated in the k-dimensional space for a given number of rounds, which are again listed in tables. Using this technique the TMD tradeoff attack they propose is around 2^{10} times faster than the Esgin-Kara attack. Again, since LILLE does not use non-linear mixing, this attack is not applicable.

4.1.5 Period:

In [3], it was shown that there are around 2^{70} Key-IV pairs in Sprout that produces keystream of period 80. Furthermore it was shown that for each key, there exists around 2^{30} IVs that lead to the LFSR being all zero during the keystream generating phase, for which keystream of period less than $80 \cdot 2^{40}$ is produced. However in LILLE we can guarantee a minimum period for the Keystream sequence produced by any Key-IV pair. The basic unit of the cipher is the ENC_{K_1,K_2,IV,L_r} module which behaves as a random permutation. The permutation depends on the current LFSR state. Since we use a maximum length LFSR for all three versions of the cipher, the ENC module will take any given LFSR state L_r as input only after $\mathsf{LCM}(2^\ell - 1, 720) = 48 \cdot (2^\ell - 1)$ iterations for LILLE-40, LILLE-80 and $16 \cdot (2^\ell - 1)$ iterations for LILLE 60. Since the ENC permutations themselves repeat after $48 \cdot (2^\ell - 1)$ iterations (for LILLE-40, LILLE-80), we can guarantee that the period of the keystream sequence for any key, IV pair is some integer multiple of $40 \cdot 48 \cdot (2^\ell - 1)$. For LILLE-60 this figure is $40 \cdot 16 \cdot (2^\ell - 1)$. This comes to around $2^{50.9}, 2^{69.3}, 2^{90.9}$ for LILLE-40, LILLE-60 and LILLE-80 respectively.

4.2 Underlying Function

4.2.1 Differential Attack:

The only way to insert a difference in the internal state is via the initial vector. A look at the structure of LILLE tells us that the best way to insert a difference is via the 80^{th} IV bit. In this case a difference gets inserted in the 80^{th} round of the ENC module. The resulting difference between states can be distinguished after 73 rounds, using around 2^{20} samples. We have

$$\mathsf{Pr}[S_{153}[0] \oplus S'_{153}[0] = 0] \approx \frac{1}{2} - 2^{-9},$$

where S_{153} and S'_{153} denote the two 153^{rd} round states initialized by a difference in the 80^{th} IV bit. The best possible differential attack we could find is on 153 rounds of the ENC module. This in turn means that the permutation P can be distinguished upto 73 of the 120 rounds using such a technique.

4.2.2 Linear Approximations:

In [12], it was shown that in Grain like constructions a linear approximation of some bias exists between a linear sum of the keystream bits and a linear sum of some of the LFSR bits. Using this approximation, the authors were able to apply a variant of the Fast Walsh transform to deduce the entire LFSR state of Grain v0. The authors of Grain identified this weakness and tweaked the keystream producing function in Grain v1 in a manner so that the above linear approximation held with much lower bias, which made a state recovery attack infeasible. In LILLE the keystream is produced by directly xoring the state with one half of the secret key. So any equation for linear approximation that utilizes the linear approximation of the update function in LILLE must also contain the unknown secret key. This makes any linear attack of the type described in [12] infeasible.

Furthermore the update function is 3-resilient which means that the affine approximation with least number of linear terms has a correlation coefficient of $3 \cdot 2^{-8}$. Omitting the publicly known IV and LFSR contributions to the update function, the best linear approximation for 80 out of the 120 rounds of the permutation P with the highest correlation coefficient we could find was

$$x_{80} = x_0 + x_7 + x_{15} + x_{16} + x_{18} + x_{26} + x_{36},$$

where $x_0, x_1, \ldots x_{39}$ and $x_0, x_1, \ldots x_{39}$ and $x_{80}, x_{81}, \ldots x_{119}$ are the input and output 40 bits for P reduced to 80 rounds.

The approximation has a correlation coefficient of $3^6 \cdot 2^{-48} \approx 2^{-38.5}$, which can not be distinguished in less than 2^{40}.

4.2.3 Conditional Differential Cryptanalysis:

The notion of Conditional Differential Cryptanalysis was first proposed by Ben-Aroya/Biham to attack Lucifer [10,11]. Knellwolf et al. in [37] used it attack reduced round versions of the Grain stream cipher. This attack paradigm is closely related to Cube and Dynamic Cube attacks [20,21] and Algebraic IV Differential Attacks [46]. Conditional Differential Cryptanalysis has been used to cryptanalyze reduced round versions of the Grain family [36,37], the stream cipher Trivium and the block cipher KATAN [38]. In a typical attack scenario, the attacker introduces some difference via a public variable like the IV/plaintext into the underlying cryptosystem and tries to construct a distinguisher that depends on the values of one or more Secret Key bits used in the system. The distinguisher is constructed in such a manner that the attacker is able to distinguish the output of the cryptosystem from an ideally distributed random variable if and only if he guesses the values of certain Secret Key bits/expressions correctly.

In LILLE the attacker, as before, introduces a difference in the 80^{th} IV bit. At rounds $t = 84, 87, 88$ the difference sits on the locations $35, 32, 31$ of the state register. These locations also happen to provide inputs to the update function of the state register. The attacker tries to prevent the propagation of the differential through the feedback function in these rounds by invoking certain algebraic conditions at these rounds. These typically involve assigning to 0 or 1 a function involving a secret key bit and some non linear function of the IV. A total of 8 such assignments are required. Thereafter as in [36], the IV space is divided into 2^8 non intersecting subsets so that all the 8 conditions are satisfied in only one these sets. If all the algebraic conditions are satisfied we have a bias in the 155^{th} round (which is observed in only one of the 2^8 subsets):

$$\mathsf{Pr}[S_{155}[0] \oplus S'_{155}[0] = 0] \approx \frac{1}{2} - 2^{-9}$$

This is the best possible distinguisher we could find for the permutation that takes less than 2^{30} samples. This again implies a distinguisher for 75 of the 120 rounds of P.

4.2.4 Cube Attacks:

Cube attacks were first introduced in [22], and have been used to cryptanalyze full versions of Grain-128 [20,21]. However we choose an update function of algebraic degree equal to 6 and the high number of rounds in the ENC module lead us to believe that the algebraic degree of the keystream bits in the secret key variables is close to 80. Hence we believe that cube attacks will be infeasible against all versions of LILLE.

4.2.5 Algebraic Attacks:

Algebraic attacks against LFSR-based keystream generators were introduced in [16]. The authors were able to attack ciphers like Toyocrypt and LILI-128, by lowering the algebraic degree of keystream equations by multiplying them by suitably chosen annihilator polynomials. These attacks work best if the algebraic degree of the bank of equations is more or less constant. But for ciphers like LILLE that are based on non-linear feedback shift registers, the algebraic degree of the state variables increase very rapidly and after 720 rounds of the ENC module, we expect the degree to be close to 80 in the secret key bits. As a result, algebraic attacks seem to be infeasible against LILLE.

4.2.6 Slide Attacks and Weak Key-IV pairs:

Slide attacks have been reported against Grain v1 in [18] to speed up exhaustive key search by a factor of 2. These attacks make use of the similarity of the state update routines in the Key-IV setup and the keystream generating phases of Grain v1. However, in LILLE any such attack is clearly defeated because of the presence of the LFSR which guarantees that the same version of the ENC permutation can not repeat before $48 \cdot (2^\ell - 1)$ iterations for LILLE 40,80 (and $16 \cdot (2^\ell - 1)$ for LILLE 60). Weak key-IV pairs that land the linear register in the all zero state after the key-IV setup and thus weaken the algebraic structure of the cipher has been reported against Grain v1 [48] and Sprout [3]. However this can not occur in LILLE because of the presence of the maximum length LFSR which never attains the all zero state irrespective of the key and IV used.

Recently, Hamann et al. have proposed a distinguishing attack on stream ciphers with a small internal state [30]. The attack exploits a collision of an initial state and a state in key generation phase. In LILLE, such a collision is never found, because a state of LFSR is always different in each clock. Thus, their attack is not applicable to LILLE.

4.2.7 Divide and Conquer Attacks:

We consider attacks in which the 40 bit subkey K_1 is completely known. In that event, the system reduces to a 2-round iterated Even-Mansour structure with a single key and known IV. Even in that case, there is no known method to find the second subkey efficiently [24].

4.2.8 Differential Fault Attack:

Fault attacks on stream ciphers is a well researched topic as is apparent from numerous papers in literature [7, 8, 32]. If the attacker is able to apply time and location synchronized bit flipping faults to either the state register or the LFSR, he may be able to determine the internal state of the cipher by comparing the faulty and fault-free keystream bits and formulating enough equations to solve for the internal state in the final round of the ENC module. Once the internal state is found, K_1 can be calculated, just by xoring the internal state with the 40 bit keystream block. However it is not clear how the attacker can use this method to deduce the value of

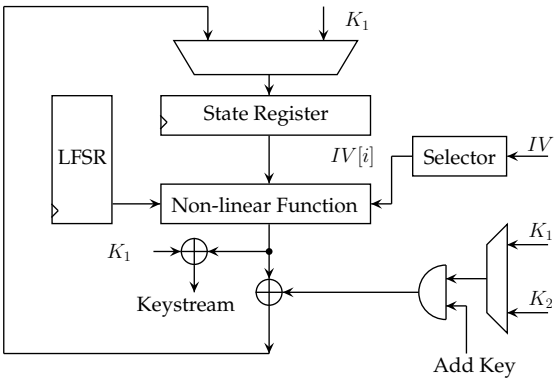


Fig. 3 Implementation of the LILLEStream cipher

K_2. Notwithstanding, we do not claim security from fault attacks.

4.2.9 Attack against Lizard:

In [6], the authors outline a number of cryptanalytic results against Lizard. First, they show that it is possible to get **a)** one key K and 2 IVs V_0, V_1 so that K, V_0 and K, V_1 produce the same keystream and **b)** key-IV pairs K_0, V_0 and K_1, V_1 so that K_0, V_0 and K_1, V_1 produce the same keystream. Both the attacks are possible due to the non-injective nature of the key-IV mixing function used in Lizard. This is not possible in LILLE because the key-IV mixing is one to one. Thereafter the authors propose a slide based distinguishing attack on Lizard. Slide attacks are not possible on LILLE because of the LFSR based round constants destroy similarity of the successive ENC modules.

Finally in [6], an impossible collision attack is mounted on reduced round Lizard, that gain takes help of IV pairs that with a given key, produce the same keystream. This is not possible in LILLE because no such pairs exist.

5. Implementation Results

In Figure 3, we present the circuit for the implementation of the LILLE stream cipher. The circuit consists of

1. A State Register,
2. A Linear Feedback Shift Register (LFSR),
3. A logic block named "Non-linear function" that implements the transformation $S_t \rightarrow S_{t+1}$,
4. A selector that takes the 80 bit IV and filters its i^{th} bit $IV[i]$ in each round,
5. A multiplexer that filters K_1, K_2 in respective rounds.
6. A bank of AND gates that helps filter the current round key only at the last (120th) iteration of the Permutation.
7. Xor gates for key addition, and other control logic.

In the first clock cycle the most significant 40 bits of the Secret Key K_1 is loaded onto the state register, and the state register is run for 120 cycles for computing the permutation P. The two halves of the secret key K_1 and K_2 are filtered in alternate rounds through a multiplexer and this output is used as the round key. In the 120th cycle of the permutation, the "Add Key" signal is set to 1, which filters the current round key for addition to the state. However, the round key is not added in the 6th and final permutation of the ENC cycle. This is because the addition of K_1 after the last permutation of the current ENC operation cancels out with the addition of K_1 before the first permutation round of the next ENC operation[†]. In the final round of the final permutation instance of the ENC operation, the keystream is made available by xoring K_1 with the state.

5.1 Simulation Results

We implemented the three ciphers LILLE-40, LILLE-60 and LILLE-80 using the standard cell libraries based on the STM 90nm and 65 nm logic process. The following design flow was adhered to. All the designs were initially implemented in VHDL and the functional verification was done using *Mentor Graphics ModelSim SE* software. The designs were then synthesized using the *Synopsys Design Compiler* for the Standard Cell library of the a) STM 90nm Logic Process: CORE90GPHVT v 2.1.a. and b) STM 65nm Logic Process: CORE65LPSVT v 5.1 The switching activity file was then generated by performing a timing simulation on the synthesized netlist using the *Synopsys VCS* Software. The power was then estimated with the *Synopsys Power Compiler* by using the switching activity file. In Table 1, we compare our implementation results with current state of the art hardware stream ciphers providing 80-bit security Grain v1, Trivium, Sprout. The area of the design is provided both in μm^2 and Gate Equivalents (GE). Although the sizes of Sprout and Plantlet are smaller than our construction, we have already pointed out the security issues in these ciphers in Section 1.4.

As an instructive example, in Figure 4, we present a break-up of the area shares taken by the various components of the circuit in the design of LILLE-40 using the standard cell library of the STM 90nm logic process. As can be seen, a major part of the circuit area (around 40%) is occupied just by the registers.

5.2 Discussion

We would like to point out that since the effective internal state is larger than 40 bits, the design is secure against generic TMD tradeoff attacks. However the Key and IV state do not require any additional storage in our construction. And so we require storage equal to only $40 + \ell$ bits, where ℓ is the size of the LFSR. The LFSR state is completely public, and mainly plays the role of a counter. Therefore the only variable internal state of the cipher is of 40 bits. Another aspect in which the design differs from traditional shift registers is the use of Key and IV. Generally, stream ciphers do not hold a secret key and an IV after their initialization processes, whereas LILLE does. However this method is used widely

[†]This does not create any security issue as far as we can see.

#	Cipher	State Size	STM 90nm			STM 65nm			Throughput (Mbps) @ 10 MHz	Max Data (Bits)
			Area		Power (in μW) @ 10 MHz	Area		Power (in μW) @ 10 MHz		
			(in μm^2)	(in GE)		(in μm^2)	(in GE)			
1	Grain v1	160	5072.0	1152.7	47.8	2791.9	1329.5	32.9	10	-
2	Trivium	288	8229.8	1870.5	78.4	4536.5	2160.2	54.3		2^{64}
3	Sprout	80	3367.4	765.3	30.2	1833.0	872.8	20.0		-
4	Plantlet	101	3897.6	885.8	35.4	2126.3	1012.5	23.7		-
5	Lizard	121	6516.5	1481.4	51.8	3463.2	1649.0	33.7		2^{18}
6	LILLE-40	40	4008.4	911.0	35.0	2137.7	1017.9	22.2	0.56	2^{50}
7	LILLE-60	40	4363.0	991.6	38.4	2322.8	1106.1	24.7		2^{69}
8	LILLE-80	40	4736.1	1076.4	42.8	2530.8	1205.1	28.2		2^{90}

Table 1 Implementation results

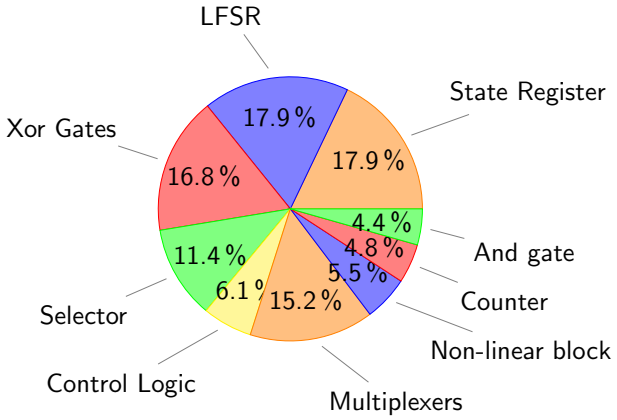


Fig. 4 Area shares for LILLE-40 in the STM 90nm logic process

in block ciphers like LED, Piccolo and Midori and even to some extent in the stream cipher Sprout. The third aspect we discuss is the issue of throughput. One can see that, the LILLE cipher family has considerably lower throughput when compared to Grain or Trivium, but this is to be expected as per the arguments outlined in Section 1.2.

5.3 Comparing with CTR mode with variable counter size

Since each of the LILLE versions have been catered to provide encryption for different data lengths, a fairer comparison can be made with block ciphers that operate in CTR mode with different counter sizes. For example we recommend LILLE-40 be used for encrypting at most 2^{50} bits using the same key-IV pair. However due to birthday limitations, the maximum number of different blocks a block cipher of block size of 64 bit can encrypt is 2^{32}. Thus we can only employ a 32 bit counter with a 64 bit block cipher. If we employ a 32 bit IV and let the counter constitute the remaining 32 bit input to the block cipher, the maximum number of databits the mode can encrypt is $64 * 2^{32} = 2^{38}$ bits. This is well short of 2^{50}. So any meaningful comparison can only be made with a 96 bit block cipher which can encrypt upto $96 * 2^{48} = 2^{54.6}$ bits or a 128 bit block cipher which can encrypt upto $128 * 2^{64} = 2^{71}$ bits. A combination of 43 bit counter and 128 bit block cipher **OR** 44 bit counter and 96 bit block cipher can encrypt 2^{50} bits. So for LILLE-60 that can accommodate 2^{69} bits, we could compare its hardware performance with a 128 bit block cipher using a 62 bit counter. LILLE-80 can encrypt 2^{90} bits, which can not be reached by any 96 or 128 bit block cipher in counter mode. In Table 2, we tabulate estimates for all block ciphers that provide at least 80 bits security. We choose serialized implementation for all the block ciphers since they have smaller hardware area. Note that we utilize the implementations in [4, 9] to arrive at the estimates.

#	Cipher	Key	Block	Counter Size	Area (GE)	TP (Mbps) @ 10 MHz
For max data = 2^{50} bits						
1	LILLE-40	80	-	-	911	0.56
2	AES-CTR	128	128	43	2260	5.20
3	Simon-CTR	96	96	44	1155	0.19
4	Speck-CTR	96	96	44	1212	0.18
For max data = 2^{69} bits						
1	LILLE-60	80	-	-	992	0.56
2	AES-CTR	128	128	62	2360	5.20
3	Simon-CTR	128	128	62	1534	0.14
4	Speck-CTR	128	128	62	1580	0.15

Table 2 Comparison with 64 bit block ciphers with variable counter size, (TP: Throughput)

6. Conclusion

We propose a family of three stream ciphers LILLE-40, LILLE-60 and LILLE-80. The ciphers are smallest in hardware area among all other stream ciphers that offer 80 bit security without any distinguishing or key recovery attacks. The design is such that there are no weak key-IV pairs and a minimum period for the keystream sequence is guaranteed. We also analyzed the security of the cipher against a number of existing attack paradigms. This makes it an attractive choice for application in constrained environments and/or in applications that require encryption of long messages of over 2^{32} blocks.

References

[1] F. Armknecht and V. Mikhalev. On Lightweight Stream Ciphers with Shorter Internal States. In FSE 2015, LNCS, Vol. 9054, pp. 451–470, 2015.

[2] J. P. Aumasson, L. Henzen, W. Meier, and M. Naya-Plasencia. QUARK : A Lightweight Hash. Journal of Cryptology, April 2013,

Volume 26, Issue 2, pp 313-339, 2013.

[3] S. Banik. Some results on Sprout. In INDOCRYPT 2015, LNCS, vol. 9462, pp. 124-139, 2015.

[4] S. Banik, A. Bogdanov, and F. Regazzoni. Atomic-AES v2.0. In IACR Cryptology ePrint Archive 2016: 1005. Available at http://eprint.iacr.org/2016/1005.

[5] S. Banik, A. Bogdanov, T. Isobe, K. Shibutani, H. Hiwatari, T. Akishita, F. Regazzoni. Midori: A Block Cipher for Low Energy. In ASIACRYPT 2015, LNCS, vol. 9453, pp. 411-436, 2015.

[6] S. Banik and T. Isobe. Some cryptanalytic results on Lizard. In IACR eprint archive. Available at http://eprint.iacr.org/2017/346.

[7] S. Banik, S. Maitra, and S. Sarkar. A Differential Fault Attack on the Grain Family of Stream Ciphers. In CHES 2012, LNCS, vol. 7428, pp. 122–139, 2012.

[8] S. Banik, S. Maitra, and S. Sarkar. A Differential Fault Attack on MICKEY 2.0. In CHES 2013, LNCS, vol. 8086, pp. 215–232, 2013.

[9] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, L. Wingers. The Simon and Speck Families of Lightweight Block Ciphers. In IACR eprint archive. Available at https://eprint.iacr.org/2013/404.pdf.

[10] I. Ben-Aroya, E. Biham. Differential Cryptanalysis of Lucifer. In CRYPTO 1993, LNCS, Vol. 773, pp. 187-199, 1993.

[11] I. Ben-Aroya, E. Biham. Differential Cryptanalysis of Lucifer. J. Cryptology vol. 9(1): pp. 21-34, 1996.

[12] C. Berbain, H. Gilbert, and A. Maximov. Cryptanalysis of Grain. In FSE 2006, LNCS, Vol. 4047, pp. 15–29, 2006.

[13] A. Biryukov and A. Shamir. Cryptanalytic Time/Memory/Data Tradeoffs for Stream Ciphers. In ASIACRYPT 2000, LNCS, Vol. 1976, pp. 1–13, 2000.

[14] A. Bogdanov, L. Knudsen, G. Leander, C. Paar, A. Poschmann, M. Robshaw, Y. Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In CHES 2007, LNCS, vol. 4727, pp. 450-466, 2007.

[15] J. Borghoff, A. Canteaut, T. Güneysu, E. B. Kavun, M. Knežević, L. R. Knudsen, G. Leander, V. Nikov, C. Paar, C. Rechberger, P. Rombouts, S. S. Thomsen, and T. Yalçin. PRINCE - A Low-Latency Block Cipher for Pervasive Computing Applications - Extended Abstract. In Asiacrypt 2012, LNCS, vol. 7658, pages 208-225, 2012.

[16] N. Courtois and W. Meier. Algebraic attacks on stream ciphers with linear feedback. In EUROCRYPT 2003, LNCS, vol. 2656, pp. 345–359, 2003.

[17] C. De Cannière, O. Dunkelman, and M. Knežević. KATAN and KTANTAN - a family of small and efficient hardware-oriented block ciphers. In CHES 2009, LNCS, vol. 5747, pp. 272-288, 2009.

[18] C. De Cannière, O. Küçük, and B. Preneel. Analysis of Grain's Initialization Algorithm. In AFRICACRYPT 2008, LNCS, Vol. 5023, pp. 276–289, 2008.

[19] C. De Cannière and B. Preneel. TRIVIUM -Specifications. eSTREAM, ECRYPT Stream Cipher Project Report, 2005. Available at http://www.ecrypt.eu.org/stream/p3ciphers/trivium/trivium_p3.pdf.

[20] I. Dinur, T. Güneysu, C. Paar, A. Shamir, and R. Zimmermann. An Experimentally Verified Attack on Full Grain-128 Using Dedicated Reconfigurable Hardware. In ASIACRYPT 2011, LNCS Vol. 7073, pp. 327–343, 2011.

[21] I. Dinur, and A. Shamir. Breaking Grain-128 with Dynamic Cube Attacks. In FSE 2011, LNCS, Vol. 6733, pp. 167–187, 2011.

[22] I. Dinur and A. Shamir. Cube Attacks on Tweakable Black Box Polynomials. In EUROCRYPT 2009, LNCS, Vol. 5479, pp. 278-299, 2009.

[23] I. Dinur, O. Dunkelman, N. Keller, and A. Shamir. Key Recovery Attacks on 3-round Even-Mansour, 8-step LED-128, and Full AES^2. In ASIACRYPT 2013(1), LNCS, Vol. 8269, pp. 337–356, 2013.

[24] I. Dinur, O. Dunkelman, N. Keller, and A. Shamir. Cryptanalysis of Iterated Even-Mansour Schemes with Two Keys. In ASIACRYPT 2014(1), LNCS, Vol. 8873, pp. 439–457, 2014.

[25] O. Dunkelman and N. Keller. Treatment of the initial value in Time-Memory-Data Tradeoff attacks on stream ciphers. Inf. Process. Lett., vol. 107 no. 5, pp. 133–137, 2008.

[26] M. F. Esgin and O. Kara. Practical Cryptanalysis of Full Sprout with TMD Tradeoff Attacks. In Selected Areas in Cryptography 2015, LNCS, Vol. 9566, pp.67–85, 2015.

[27] S. Even and Y. Mansour. A Construction of a Cipher from a Single Pseudorandom Permutation. In ASIACRYPT 1991, LNCS, Vol. 739, pp. 210–224, 1993.

[28] J. Guo, T. Peyrin, A. Poschmann, and M. J. B. Robshaw. The LED Block Cipher. In CHES 2011, LNCS, vol. 6917, pp. 326-341, 2011.

[29] M. Hamann, M. Krause and W. Meier. LIZARD - A Lightweight Stream Cipher for Power-constrained Devices. In IACR Transactions of Symmetric Cryptology. Volume 2017, Issue 1, pp. 45-79.

[30] M. Hamann, M. Krause, W. Meier and B. Zhang. Time-Memory-Data Tradeoff Attacks against Small-State Stream Ciphers In IACR Cryptology ePrint Archive 2017: 384. Available at http://eprint.iacr.org/2017/384.

[31] M. Hell, T. Johansson, and W. Meier. Grain - A Stream Cipher for Constrained Environments. ECRYPT Stream Cipher Project Report 2005/001, 2005. Available at http://www.ecrypt.eu.org/stream.

[32] M. Hojsik and B. Rudolf. Differential Fault Analysis of Trivium. In FSE 2008, LNCS, vol. 5086, pp. 158–172, 2008.

[33] J. Hong and P. Sarkar. New Applications of Time Memory Data Tradeoffs. In ASIACRYPT 2005, LNCS, Vol. 3788, pp. 353–372, 2005.

[34] T. Isobe and K. Shibutani. Security Analysis of the Lightweight Block Ciphers XTEA, LED and Piccolo. In ACISP 2012, LNCS, Vol. 7372, pp. 71–86, 2012.

[35] K. Khoo and C. Tan. New time-memory-data trade-off attack on the estream finalists and modes of operation of block ciphers. In 7th ACM Symposium on Information, Computer and Communications Security, ASIACCS, pp. 20-21, 2012.

[36] S. Knellwolf. Cryptanalysis of Hardware-Oriented Ciphers, The Knapsack Generator, and SHA-1. PhD Dissertation, 2012. Available at http://e-collection.library.ethz.ch/eserv/eth:5999/eth-5999-02.pdf

[37] S. Knellwolf, W. Meier, and M. Naya-Plasencia. Conditional Differential Cryptanalysis of NLFSR-based Cryptosystems. In ASIACRYPT 2010, LNCS, Vol. 6477, pp. 130–145, 2010.

[38] S. Knellwolf, W. Meier, and M. Naya-Plasencia. Conditional Differential Cryptanalysis of Trivium and KATAN. In SAC 2011, LNCS, Vol. 7118, pp. 200–212, 2011.

[39] V. Lallemand and M. Naya-Plasencia. Cryptanalysis of Full Sprout. In CRYPTO 2015, LNCS, Vol. 9215, pp. 663–682, 2015.

[40] G. Li and Y. Yarom, and D. C. Ranasinghe. Exploiting Transformations of the Galois Configuration to Improve Guess-and-Determine Attacks on NFSRs. Available at http://eprint.iacr.org/2015/1045.pdf

[41] V. Mikhalev, F. Armknecht, and C. Müller. On Ciphers that Continuously Access the Non-Volatile Key. In IACR Transactions of Symmetric Cryptology. Volume 2016, Issue 2, pp. 52-79.

[42] I. Nikolic, L. Wang, and S. Wu. Cryptanalysis of Round-Reduced LED. In FSE 2013, LNCS, Vol. 8424, pp. 112–129, 2013.

[43] C. Rolfes, A. Poschmann, G. Leander, C. Paar. Ultra-Lightweight Implementations for Smart Devices - Security for 1000 Gate Equivalents. In CARDIS 2008, LNCS, Vol. 5189, pp. 89-103.

[44] K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita, and T. Shirai. Piccolo: An Ultra-Lightweight Blockcipher. In CHES 2011, LNCS, vol. 6917, pp. 342-357, 2011

[45] T. Suzaki, K. Minematsu, S. Morioka, and E. Kobayashi TWINE: A Lightweight Block Cipher for Multiple Platforms. In SAC 2012, LNCS, vol. 7707, pp. 342-357, 2013.

[46] M. Vielhaber. AIDA Breaks BIVIUM (A&B) in 1 Minute Dual Core CPU Time. In IACR eprint archive. Available at http://eprint.iacr.org/2009/402.

[47] B. Zhang and X. Gong. Another Tradeoff Attack on Sprout-like Stream Ciphers. In ASIACRYPT 2015, LNCS Vol. 9453, pp. 561-585, 2015.

[48] H. Zhang and X. Wang. Cryptanalysis of Stream Cipher Grain Family. IACR Cryptology ePrint Archive 2009: 109. Available at `http://eprint.iacr.org/2009/109`.

Appendix A: Test Vectors

A. LILLE-40

1.
```
Key        : 00000 00000 00000 00000
IV         : 00000 00000 00000 00000
Keystream  : 8932b 7cc3e 3a9e7 23520
```

2.
```
Key        : 51084 ce6e7 3a5ca 2ec87
IV         : 687de d3b3c 85b3f 35b10
Keystream  : 572b1 c2227 0452c e6301
```

B. LILLE-60

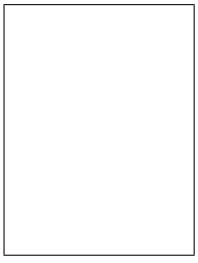
1.
```
Key        : 00000 00000 00000 00000
IV         : 00000 00000 00000 00000
Keystream  : 2f81e 66ae9 73452 4b334
```

2.
```
Key        : 51084 ce6e7 3a5ca 2ec87
IV         : 687de d3b3c 85b3f 35b10
Keystream  : 4c3ad 0fd80 ffc95 a46ea
```

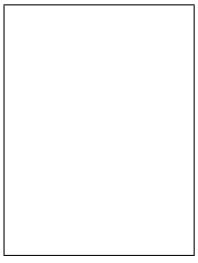
C. LILLE-80

1.
```
Key        : 00000 00000 00000 00000
IV         : 00000 00000 00000 00000
Keystream  : 8517f ffb61 0f062 79e8d
```

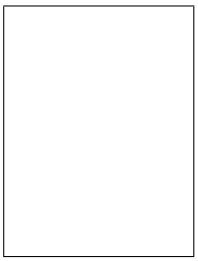
2.
```
Key        : 51084 ce6e7 3a5ca 2ec87
IV         : 687de d3b3c 85b3f 35b10
Keystream  : cd282 d508c ebb9d d21cc
```



Subhadeep Banik recieved B.Tech in Electronics and Electrical Communications Engineering and M. Tech in Automation and Computer Vision from Indian Institute of Technology Kharagpur in 2005. He recieved PhD in Computer Science from the Indian Statistical Institute, Kolkata in 2014. He is currently a Postdoc in École Polytechnique Fédérale de Lausanne. His current research interests include cryptography and digital design.



Takanori Isobe received the B.E., M.E., and Ph.D. degrees from Kobe University, Japan, in 2006, 2008, and 2013, respectively. He joined the Sony Corporation in 2008. His current research interests include information security and cryptography. He received the SCIS Paper Award and SCIS 2013 Innovation Paper Award from ISEC group of IEICE in 2008 and 2014, respectively. He also received the Best Paper Award from IEICE in 2015. He received the FSE 2011 Best Paper Award from IACR in 2011.



Masakatu Morii received the B.E. degree in electrical engineering and the M.E. degree in electronics engineering from Saga University, Saga, Japan, and the D.E. degree in communication engineering from Osaka University, Osaka, Japan, in 1983, 1985, and 1989, respectively. From 1989 to 1990 he was an Instructor in the Department of Electronics and Information Science, Kyoto Institute of Technology, Japan. From 1990 to 1995 he was an Associate Professor at the Department of Computer Science, Faculty of Engineering at Ehime University, Japan. From 1995 to 2005 he was a Professor at the Department of Intelligent Systems and Information Science, Faculty of Engineering at the University of Tokushima, Japan. Since 2005, he has been a Professor at the Department of Electrical and Electronics Engineering, Faculty of Engineering at Kobe University, Japan. His research interests are in error correcting codes, cryptography, discrete mathematics and computer networks and information security. He is a member of the IEEE.