A New Construction Method for Keystream Generators

Çağdaş Gül and Orhun Kara

***Abstract*— We introduce a new construction method of diffusion layers for Substitution Permutation Network (SPN) structures along with its security proofs. The new method can be used in block ciphers, stream ciphers, hash functions, and sponge constructions. Moreover, we define a new stream cipher mode of operation through a fixed pseudorandom permutation and provide its security proofs in the indistinguishability model. We refer to a stream cipher as a Small Internal State Stream (SISS) cipher if its internal state size is less than twice its key size. There are not many studies about how to design and analyze SISS ciphers due to the criterion on the internal state sizes, resulting from the classical tradeoff attacks. We utilize our new mode and diffusion layer construction to design an SISS cipher having two versions, which we call DIZY. We further provide security analyses and hardware implementations of DIZY. In terms of area cost, power, and energy consumption, the hardware performance is among the best when compared to some prominent stream ciphers, especially for frame-based encryptions that need frequent initialization. Unlike recent SISS ciphers such as Sprout, Plantlet, LILLE, and Fruit; DIZY does not have a keyed update function, enabling efficient key changing.**

***Index Terms*— Block cipher, stream cipher, keystream, diffusion layer, lightweight ciphers, truncated pseudorandom permutations, tradeoff attacks.**

I. Introduction

IN THIS work, we study how to design stream ciphers whose internal state sizes are less than twice their key sizes, which we call Small Internal State Stream (SISS) ciphers. Our methodology is based on the efficient construction of diffusion layers for Substitution Permutation Network (SPN) structures, and the implementation of these SPN structures within a new stream cipher mode, thereby offering a level of security that can be rigorously demonstrated.

The demand for secure communication among ubiquitous handheld devices with constrained resources has been on the rise in the field of telecommunications. Given the limited battery capacity of these lightweight portable devices, it is most likely that they provide the confidentially through frame-based encryption by using either block ciphers in stream cipher modes such as Kasumi in GSM, Keeloq of rolling code systems, and AES in WPA2, or stream ciphers. In general, stream ciphers consume more energy than block ciphers in encrypting short frames [1].

In terms of security, the threshold for recovering an internal state should not be larger than the threshold for recovering a key through tradeoff attacks in a stream cipher. This comparison suggests that the internal state size should be at least 4k/3 bits when the key length is k bits. However, to provide resistance against tradeoff attacks, most contemporary stream ciphers have internal state sizes of at least 2k bits. Despite the persistent increase in the utilization of SISS ciphers in the industry over the past two decades, there has been a scarcity of research in the literature on the secure design of such ciphers due to the strict limitations imposed on state sizes. Conversely, there are several SISS ciphers that are widely used in practical applications. However, a significant majority of them have been demonstrated to possess substantial vulnerabilities. There are practical attacks on A5/1 or A5/2 of GSM [2], E0 of Bluetooth [3], Crypto1 and Hitag2 [4] of immobiliser systems, Keeloq [5], Myfire Classics [6], [7], and RC4 of WEP [8], [9].

Armknecht and Mikhalev present a methodical approach for the development of SISS ciphers through the introduction of keyed state update functions [10], and prove that a cipher with a keyed state update provides at least k-bit security against the internal state recovery tradeoff attacks given in [11], [12], and [13]. Basically, four modern SISS ciphers with keyed updates have been designed since then. The first cipher, Sprout [10], is analyzed intensively in [14], [15], [16], and [17]. Although Plantlet is introduced to enhance the security of Sprout [18], several attacks are published [19], [20], [21]. Fruit [22] and LILLE [23] are the other recent SISS ciphers with keyed updates. Fruit-80 [24] inherits some weaknesses [19], [20] and there is a recent attack on LILLE [25]. But, no weaknesses have been reported on the latest version of Fruit yet.

In recent years, there has been a proliferation of designs for ultra-lightweight block ciphers such as [26], [27], [28], [29], and [30]. However, the literature on SISS ciphers without keyed updates is relatively scarce. One example is Lizard having a 120-bit key and 121-bit internal state [31]. Nevertheless, a new attack recovers internal states with an overall complexity of 2^{54} [32]. The main motivation of this work is to study the question of *how to design a secure SISS cipher without a keyed update.*

Manuscript received 6 July 2022; revised 13 February 2023 and 25 May 2023; accepted 13 June 2023. Date of publication 21 June 2023; date of current version 28 June 2023. The work of Orhun Kara was supported in part by TÜBİTAK 1001 Project under Grant 121E228. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Hossein Pishro-Nik. *(Corresponding author: Orhun Kara.)*

Çağdaş Gül is with the TÜBİTAK BİLGEM Informatics and Information Security Research Center, 41470 Kocaeli, Turkey, and also with the Computer Engineering Department, Marmara University, 34722 Istanbul, Turkey.

Orhun Kara is with the Department of Mathematics, İzmir Institute of Technology (IZTECH), Urla, 35430 İzmir, Turkey, and also with TÜBİTAK BİLGEM, 41470 Kocaeli, Turkey (e-mail: orhunkara@iyte.edu.tr).

Digital Object Identifier 10.1109/TIFS.2023.3287412

We investigate the utilization of Truncated Pseudorandom Permutations (TPP) as keystream generators. The studies on the statistical divergence of TPPs date back to the 1970s as an old combinatorics problem introduced by Stam [33]. Some early works have presented loose bounds independently [34], [35]. Then, the cryptography community has realized that the Stam's results are quite tight and a number of works have verified his results [35], [36], [36], [38]. Gilboa et al. have studied the question "How many queries are required to distinguish a TPP from a random function?" [39]. This question is already addressed in our case due to the restriction on the amount of data to be encrypted. A tight bound is given by Mennick [40] recently which we utilize in our security proofs.

A. Our Contributions

The primary contributions are threefold: The proposal of a new stream cipher mode with a formal proof of security, the development of a diffusion layer construction with detailed security analysis, and the design of a secure SISS cipher.

Our stream cipher mode uses a truncated pseudorandom permutation that we call the TPP mode. There are several stream cipher modes making use of pseudorandom permutations such as GCM, GCM-SIV, EDM, EDMD, PMAC1, CENC, AES-PRF, FastPRF [41], [42], [43], [44], [45], [46], [47], all of which utilize key schedules. However, the TPP mode does not require a key schedule, which enables us to construct SISS ciphers by saving the cost of key registers. Furthermore, we provide the security proofs of our mode in Section III.

The TPP mode necessitates the use of a pseudorandom permutation, specifically one with a large block size given that the permutation is fixed. However, the block sizes of conventional block ciphers are often inadequate for this mode. Therefore, we propose a new method of constructing diffusion layers for SPN ciphers that possess larger block sizes. We show that these layers provide fast diffusion for the SPN ciphers having relatively large numbers of S-boxes (Substitution-boxes), in terms of the numbers of differentially and linearly active S-boxes by Theorem 6 and Theorem 4 in Section IV. Furthermore, our proposed diffusion layers effectively integrate matrix multiplications and bitwise permutations, resulting in a reduction of XOR (exclusive-or) operations in hardware. We introduce two examples, the 120-bit and 160-bit diffusion layers, which consist of 72 and 96 XOR operations, respectively. The proposed method can be applied to a variety of cryptographic designs, including block ciphers, hash functions, and sponge functions.

To demonstrate the feasibility of our new construction method, we introduce the DIZY SISS cipher in Section V. DIZY comes in two variations, each with a different key length. The first variant, DIZY-80, uses an 80-bit key, while the second variant, DIZY-128, uses a 128-bit key. We give their security analyses in Section VI with their implementation results on Field Programmable Gate Array (FPGA) and Application-Specific Integrated Circuit (ASIC) platforms in Section VII. We compare our performance results with some of the well-known lightweight ciphers in Table IV.

II. Motivation and Problem Statement

Tradeoff attacks form a class of generic attacks that are typically divided into two phases: An offline phase, during which tables are constructed containing input/output pairs of a given one-way function, and an online phase, during which the outputs of the one-way function are searched for within the previously constructed tables.

The internal state size of a stream cipher is supposed to be at least twice as large as its security level to counter tradeoff attacks where the security level is typically equated with the length of the key used in the stream cipher [11], [12], [13]. In this case, the time complexity of the online phase of a generic tradeoff attack to recover an internal state is at least 2^k for a k-bit key. However, if we define our oneway function as the function whose input is the main key and the output is a keystream part for a chosen and fixed IV, we can recover the main key in about $2^{2k/3}$ calls with $2^{2k/3}$ memory in the online phase by the Hellman tradeoff attack [50], [51]. That is, the Hellman attack is applicable to any symmetric cipher as a k-bit oneway function. Therefore, when evaluating the security of any tradeoff attack, it is essential to compare it with the Hellman attack instead of simply measuring it against the exhaustive search. As a result, the default security threshold of the online phase of any tradeoff attack for an internal state recovery must be $2^{2k/3}$ rather than 2^k. Hence, the internal state size is enough to be at least $4k/3$ which makes the design of SISS ciphers possible [52], [53].

Finding preimages of a oneway function of the identical input-output sizes is known as the Hellman inverting problem with a default solution proposed by Hellman as $T^2M = N^2$ in a domain of N elements with the offline complexity $P = N$. Here T and M represent the time and the memory complexities of the online phase [50]. The Biryukov-Shamir curve is $M^2D^2T = N^2$ with $D^2 \leq T$, where N is the number of all the internal states [11]. It is enough to invert one of D keystream segments. Moreover, $PD = N$ in the offline phase. If the internal state size is more than $4k/3$ then the online phase of the Biryukov-Shamir attack is at least $2^{2k/3}$, which is achieved when we have $2^{k/3}$ data [11]. The offline cost is at least 2^k. So, if the number of encryptions is restricted to be at most $2^{k/3}$ then the Biryukov-Shamir inverting problem for the internal states will be as hard as the Hellman inverting problem for the main key with a fixed IV. A similar argument is valid for the Babbage-Golić attack [12], [13].

Almost all the modern key stream generators without keyed updates have larger internal states than $4k/3$. The security against divide-and-conquer or guess-and-determine attacks is usually provided by means of large states of keystream generators such as 288-bit internal state of Trivium with 80-bit key [54]. So, it is unknown in the literature how to design secure SISS ciphers. We address this comprehensive open problem.

III. A New Stream Cipher Mode

Let P be a pseudorandom permutation used as a state update function such that $P(x_{i-1}) = x_i$. Define the output function as the least significant w bits $y_i = TPP(x_{i-1})$ of x_i where

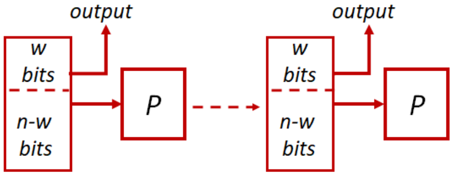

Fig. 1. TPP mode.

TPP is the composition of P and the truncation function. We describe two games, Game A and Game B, to prove the indistinguishability of the TPP mode. There are two entities in a game who are the challenger and the distinguisher. Only the challenger has access to the permutation P and the distinguisher has to ask the challenger for the corresponding outputs of the inputs he/she can query since P is unknown to the distinguisher. The challenger publishes either the corresponding outputs or random outputs with probability ½. The objective of the distinguisher is to guess the source with the highest possible probability by exploiting his/her infinite computational power.

Game A entails a powerful assumption that the distinguisher can choose the truncated part of the input in the internal state. We can nevertheless provide a bound for his/her advantage to distinguish the outputs of the TPP mode from the outputs of a random function. Game B simulates the real use case scenario of the mode where the distinguisher does not query anything but tries to distinguish a keystream sequence generated through the TPP mode from a random sequence.

a) Game A: Let a distinguisher queries distinct $z_1, \dots, z_q \in GF(2)^{n-w}$ adaptively where $GF(2)$ is the binary Galois field. Let $y_i = TPP(z_i || y_{i-1})$ where y_0 is chosen by the challenger at random and let $f : GF(2)^{n-w} \to GF(2)^w$ be a random function. One bit is chosen randomly. The distinguisher is given the output $\mathcal{O}_f = \{f(z_i : i = 1, \dots, q)\}$ if the random bit is 0, and the output $\mathcal{O}_{TPP} = \{y_i : i = 1, \dots, q\}$ otherwise. The objective of the distinguisher is to correctly guess the source as either TPP or f with the highest possible probability. For any algorithm $\mathcal{A}_\mathcal{D}$ returning 1 when his/her guess is for TPP, define the advantage $\mathbf{Adv}^f_{TPP}(\mathcal{A}_\mathcal{D})$ as

$$\mathbf{Adv}^f_{TPP}(\mathcal{A}_\mathcal{D}) = |\Pr({\mathcal{A}_\mathcal{D}}^{TPP} = 1) - \Pr({\mathcal{A}_\mathcal{D}}^f = 1)|$$

where the former and latter probabilities are over the drawings through TPP and f respectively. Lemma 1 gives an upper bound for the advantage of any distinguisher.

Lemma 1: Let $q < 2^{n-1.5w}$. The advantage of distinguishing TPP from a random function f in pairwise distinct q queries is bounded by

$$\mathbf{Adv}^f_{TPP}(\mathcal{A}_\mathcal{D}) \leq \frac{1}{2}\sqrt{\frac{(2^w - 1)q(q-1)}{(2^{n-w} - 1)(2^{n-w} - q + 1)}}.$$

Proof: Let $f_1(z) = TPP'(z)$ by truncating the most significant $n - 2w$ bits of the output of another random permutation P' of $(n - w)$ bits. The expected numbers of occurrences of any output $\alpha \in GF(2)^w$ among $j - 1$ outputs are equal in f_1 and TPP, which we denote ℓ. Then, $\left|\frac{2^{n-w}-\ell}{2^n-j+1} - \frac{1}{2^w}\right| < \left|\frac{2^{n-2w}-\ell}{2^{n-w}-j+1} - \frac{1}{2^w}\right|$. That is,

$$|\Pr(TPP(z_j||y_{j-1}) = \alpha|\{z_i\}) - \Pr(TPP(z_j||y_{j-1}) = \alpha)|$$

is less than the distance

$$|\Pr(f_1(z_j) = \alpha|\{z_i\}) - \Pr(f_1(z_j) = \alpha)|$$

for α and $j > 2$ with $i = 1, \dots, j-1$. Therefore, $\mathbf{Adv}^f_{TPP}(\mathcal{A}_\mathcal{D}) \leq \mathbf{Adv}^f_{f_1}(\mathcal{A}_\mathcal{D})$. Let Y be the distribution coming from $y_i = TPP'(z_i)$ and X be the distribution coming from the outputs of f. Let $KL(Y; X)$ be the Kullback-Leibner divergence which is

$$\sum_\alpha \Pr(Y = \alpha) \log\left(\frac{\Pr(Y = \alpha)}{\Pr(X = \alpha)}\right).$$

It has an upper bound $\frac{(2^w-1)q(q-1)}{2(2^{n-w}-1)(2^{n-w}-q+1)}$ [33], [40]. Then, $\frac{1}{2} + \mathbf{Adv}^f_{f_1}(\mathcal{A}_\mathcal{D})$ is bounded above by $\frac{1}{2} + \left(\frac{1}{2}KL(Y; X)\right)^{1/2}$ by Pinker's inequality [33], which gives the result. ■

b) Game B: The distinguisher has the full advantage in case the queries collide in Game A. We introduce the following game. Let the challenger produce the keystream sequence $\{y_i\}$ of length D through $y_i = TPP(z_i)$, $P(z_i) = z_{i+1}$ where z_0 is randomly chosen and is kept secret. The challenger produces another sequence $\{y'_i\}$ through a True Random Number Generator (TRNG) and chooses one of $\{y_i\}$ or $\{y'_i\}$ with probability ½ to publish. A distinguisher is supposed to guess correctly with the highest possible probability via an algorithm $\mathcal{A}_\mathcal{D}$ whether the sequence he/she is given is either $\{y_i\}$ or $\{y'_i\}$.

Theorem 1: Let $D < 2^{n-w}$. The advantage of distinguishing $\{y_i\}$ from $\{y'_i\}$ by $\mathcal{A}_\mathcal{D}$ is bounded above by $\frac{D}{2^n}(2^{w/2} + \frac{1}{2})$.

Proof: Let $D < 2^{n-w}$. The probability that the period of y_i is less than D is $D/2^n$ which contributes $D/2^{n+1}$ to the advantage. Otherwise, all z_i's are pairwise distinct since P is a permutation. As in the proof of Lemma 1, let Y be the distribution coming from $\{y_i\}$ and X be the distribution coming from the outputs $\{y'_i\}$ of a $TRNG$. Then, $KL(Y; X) \leq (2^w - 1)D(D - 1)/2(2^n - 1)(2^n - D + 1)$ and hence the advantage is bounded by

$$\frac{1}{2}\sqrt{\frac{(2^w - 1)D(D-1)}{(2^n - 1)(2^n - D + 1)}} + \frac{D}{2^{n+1}}$$

by Pinker's inequality [33]. Note that $[(2^w - 1)D(D - 1)]/[4(2^n - 1)(2^n - D + 1)] \leq D^2 2^w/2^{2n}$ for $D < 2^{n-w}$. ■

IV. Design Rationale and Diffusion Layer

An SPN cipher consists of s-bit S-boxes and a linear transformation as its confusion and diffusion layers respectively. The block length is n and the number of S-boxes is $t = n/s$. We choose a 5-bit perfect nonlinear permutation whose nonlinearity is 12, as the S-box for our design, possessing a unique property where one-bit difference does not produce a one-bit output difference in the least significant two bits and also any input bit in the least significant two bits has no bias with any output bit. Its table is given in Section V.

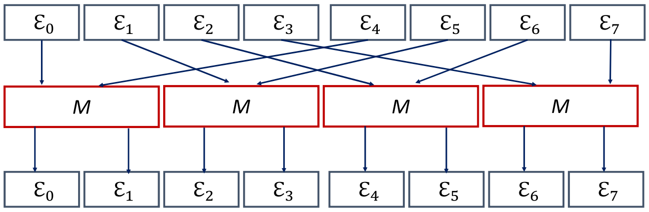

Fig. 2. High level structure of one permutation block.

A. Linear Transformation

The objective of the linear transformation is to maximize the number of active S-boxes with respect to both linear and differential cryptanalysis. An S-box is considered to be differentially active if it exhibits non-zero differential behavior, and linearly active if there exists a linear mask in its output.

We assume the number of S-boxes, t, is divisible by 8 and we define 8 equivalence (eq.) classes $\mathcal{E}_0, \ldots, \mathcal{E}_7$, which are pairwise disjoint and their union gives all the indices of S-boxes. $\mathcal{E}_i$ contains indices of $t/8$ S-boxes and it is not important which S-boxes are contained in which class for our statements. We define an $n/4 \times n/4$ invertible binary matrix M and our linear transformation is constructed through M as

$$M \cdot \begin{pmatrix} \mathcal{E}_i^r \\ \mathcal{E}_{i+4}^r \end{pmatrix} = \begin{pmatrix} \mathcal{E}_{2i}^{r+1} \\ \mathcal{E}_{2i+1}^{r+1} \end{pmatrix} \text{ for } i = 0, 1, 2, 3 \qquad (1)$$

where $\left(\begin{smallmatrix} \mathcal{E}_i^r \\ \mathcal{E}_{i+4}^r \end{smallmatrix}\right)$ is an $n/4 \times 1$ vector consisting of the bits of the S-boxes in $\mathcal{E}_i$ and $\mathcal{E}_{i+4}$ in the r-th round in a fixed order starting from the S-box indices in $\mathcal{E}_i^r$. We ignore the round number r in the notation for simplicity from now on. The vector $\left(\begin{smallmatrix} \mathcal{E}_{2i} \\ \mathcal{E}_{2i+1} \end{smallmatrix}\right)$ is defined similarly. Any differentially (linearly) active eq. class contains at least one differentially (linearly) active S-box.

Recall that the differential branch number of a matrix in terms of a given word size is the minimum number of the sum of the nonzero words of all the nonzero inputs and their outputs. The branch number of the transpose matrix gives the linear branch number. If the branch number is the maximum possible then the matrix is called a Maximum Distance Separable (MDS) matrix.

Definition 1: We say an invertible matrix M **provides Second-Degree Diffusion (SDD)** if it satisfies the following.

- M is MDS as a 2×2 matrix over a pair of eq. classes.
- The row reduced echelon form of any successive $s \times s$ submatrix as an s-bit word is nonzero and contains at most one 1 in a row and at most one 1 in a column.

We assume M provides SDD by default. Proposition 1 states that a nonzero s-bit vector produces a nonzero output by the multiplication of at least one of the $s \times s$ submatrices.

Proposition 1: Let an $s \times s$ matrix S be a submatrix of M whose i-th column and j-th row are nonzero. Then $SX \neq 0$ and $Y^\intercal S \neq 0$ for any vector X whose i-th coordinate is 1 and Y whose j-th coordinate is 1 where $Y^\intercal$ is Y's transpose.

Proof: Let PS be in row reduced echelon form and $\exists k$ with $(PS)_{k,i} = 1$ since its i-th column is nonzero. The k-th row of PS contains exactly one 1, which implies that the k-th coordinate of PSX is 1. Then, $PSX \neq 0$. So, $SX \neq 0$ since P is invertible. The second part is similar. ■

If M provides SDD then its transpose also provides SDD. So, any generic statement about differential analysis has an analogy for linear analysis. Therefore, we skip the proofs for the linear case. If one S-box is active then both of the eq. classes are differentially and linearly active in both the next and the previous rounds since M is an MDS matrix over the words of eq. classes.

If M provides SDD then M^{-1} also provides SDD. In fact, the branch number of M^{-1} as a 2×2 matrix is also 3. Moreover, for any $s \times s$ submatrix S of M, consider $S \cdot X = Y$. Then the nonzero bits of Y can be inverted into the bits of X in the support since the row-reduced echelon form of S contains at most one 1 in a row and at most one 1 in a column. Let the corresponding submatrix of M^{-1} be S' with $S' \cdot Y = X'$ where X' is the support of X whose bits appear in Y. Then, the row reduced echelon form of S' also contains at most one 1 in a row a since we can recover the support X' as a subset of X. A similar argument is valid on the transpose of M.

We show by Theorem 4 and Theorem 6 that matrices providing SDD have fast diffusion. The dependencies among active S-boxes should be considered when there is no randomization with round key addition. We minimize the dependencies by sending each bit of the output of an S-box to different S-boxes in our construction, DIZY, where each passive bit randomizes the S-box inputs.

Lemma 2: If only one S-box is active in the i-th round, then there are 2, 4, and 8 differentially (linearly) active eq. classes in the next (previous) three rounds respectively. Moreover, there are at least 4 differentially (linearly) active S-boxes in the $(i+4)$-th round ($(i-4)$-th round).

Proof: Considering the indices in modulo 8, $\mathcal{E}_{2j}$ and $\mathcal{E}_{2j+1}$ are differentially active in the next round by Equation 1. Similarly, $\mathcal{E}_{4j}$, $\mathcal{E}_{4j+1}$, $\mathcal{E}_{4j+2}$ and $\mathcal{E}_{4j+3}$ are active in the $(i+2)$-th round and they are all distinct. Eventually, all the eq. classes will be differentially active in the $(i+3)$-th round since their indices take all the values from 0 to 7 modulo 8. In the $(i+4)$-th round, at least half of the eq. classes are active since M is MDS, yielding at least 4 differentially active S-boxes. ■

Theorem 2: If there is one active S-Box in the i-th round then there are 2, 4, and 8 differentially/linearly active eq. classes in the $(i \pm 1)$-th, $(i \pm 2)$-th and $(i \pm 3)$-th rounds respectively.

Proof: We have 2, 4, and 8 differentially active eq. classes in the next three rounds respectively by Lemma 2. Moreover, there are 2, 4, and 8 differentially active eq. classes in the previous three rounds also since M^{-1} provides SDD. ■

Corollary 1: If there is only one active S-Box in the i-th round then there are at least 37 differentially/linearly active S-boxes in 9 rounds starting from $i-4$ up to $i+4$.

Proof: There are at least 2, 4, 8, and 4 differentially/linearly active S-boxes starting from $i-4$ up to $i+4$ rounds by Theorem 2 and Lemma 2. So, we have at least 37 active S-boxes. ■

Matrices with high branch numbers constitute a good diffusion with Equation 1.

Proposition 2: Let the branch numbers of M and its transpose with respect to s-bit words be d and d_T respectively.

If only one S-box is active in a round, there are at least d (d_T), $2d+1$ ($2d_T+1$) and $5d$ ($5d_T$) differentially (linearly) active S-boxes in the following (previous) two, three and four rounds respectively, starting from the current round.

Proof: If there are k_1 and k_2 differentially active S-boxes in two active eq. classes ($k_1, k_2 \neq 0$) in the second round, then $k_1 + k_2 \geq d-1$. Similarly, if l_1, l_2, l_3, l_4 (all are nonzero) are the number of active S-boxes in four active eq. classes in the third round, we have $l_1+l_2 \geq d-k_1$ and $l_3+l_4 \geq d-k_2$. If t_i is the number of active S-boxes in the $(i-1)$-th active eq. class for $i = 0, \ldots, 7$ in the fourth round, then $t_{2i-1}+t_{2i} \geq d-l_i$ for $i = 0, 1, 2, 3$. Then, we have $k_1+k_2+1 \geq d$, $\sum_{i=0}^{3} l_i + d \geq 2d+1$ and $\sum_{i=0}^{7} t_i + \sum_{i=0}^{3} l_i + k_1 + k_2 + 1 \geq 5d$ active S-boxes in two, three and four rounds respectively. ■

We can exploit Proposition 2 to maximize the number of active S-boxes by using MDS matrices over s-bit words.

Corollary 2: Let M be an MDS matrix over s-bit words. If one S-box is active in the i-th round, then there are at least $5t/2+9$ differentially/linearly active S-boxes in seven rounds starting from the $(i-3)$-th round.

Proof: The inverse and transpose matrices are also MDS. If only one S-box is active in the i-th round, then there are at least $5t/4+5$ differentially/linearly active S-boxes in the forthcoming and previous four rounds by Proposition 2. So, we have at least $2 \cdot (5t/4+5) - 1 = 5t/2+9$ active S-boxes in seven rounds starting from the $(i-3)$-th round. ■

We focus on lightweight constructions. So, we prefer lightweight matrices, which also can have nice diffusion properties.

Lemma 3: If there are at least 3 differentially/linearly active eq. classes, then there are at least 6 differentially/linearly active eq. classes in two consecutive rounds.

Proof: M is MDS over a pair of eq. classes by Definition 1. So, its branch number is 3. At least two M is active which gives at least 6 active eq. classes in two rounds. ■

We prove that there are at least 6 eq. classes in two consecutive rounds on average when we consider 8 or more rounds in Theorem 4. For this generalization, we need to introduce Lemma 4 and then Proposition 3 first.

Lemma 4: Assume there are 3 active eq. classes in the 2nd and 4th rounds. Then, there are 6 active eq. classes in the 1st and 5th rounds and 3 or 4 active eq. classes in the 3rd round.

Proof: The number of active eq. classes in the third round cannot be 6 since these 6 eq. classes cannot be three pairs of the form $\mathcal{E}_i, \mathcal{E}_{4i}$ to produce 3 active classes in the fourth round. 3 active classes can never produce 5 active classes. So, the only possibilities are 3 and 4. In both cases, we have two adjacent active classes in the fourth round. These two adjacent active classes produce 4 active classes in the fifth round by Proposition 3 and the other active class produces two more active classes. So, we have 6 active classes. The same argument in the decryption direction implies that there are also 6 active classes in the first round. ■

Proposition 3: Assume one odd indexed eq. class and one even indexed eq. class are differentially active. Then, four eq. classes are differentially active in the next round and all the eq. classes will be differentially active after two-round encryption. Similarly, if the indices of two linearly active eq. classes are $2i$ or $2i+1$ and $2i+4$ or $2i+5$ for $i = 0, 1$ then four eq. classes are linearly active in the previous round and all the eq. classes are linearly active after two-round decryption.

Proof: Let $\mathcal{E}_i$ and $\mathcal{E}_j$ be active. Assume i is even and j is odd. If i and j are adjacent then four eq. classes are differentially active in the next round and all the eq. classes will be differentially active after two-round encryption since $\mathcal{E}_{2j \mod 8}$, $\mathcal{E}_{2j+1 \mod 8}$, $\mathcal{E}_{2j+2 \mod 8}$ are all distinct modulo 4. If they are not adjacent, then either (i and $j+4 \mod 8$) or ($i+4$ and $j \mod 8$) or ($i+4$ and $j+4 \mod 8$) are adjacent. But, all these pairs will make the same eq. classes differentially active in the next round according to Equation 1. For the case of linear cryptanalysis, the pairs $(2i, 2i+4) \mod 8$, $(2i, 2i+5) \mod 8$, $(2i+1, 2i+4) \mod 8$ and $(2i+1, 2i+5) \mod 8$ make the same eq. classes linearly active in the previous round. Then, pick one pair whose indices differ by 4. This pair makes four eq. classes linearly active in the previous round and all the eq. classes linearly active after two-round decryption. ■

All the technical details in this section are preliminary to prove Theorem 3 and Theorem 4.

Theorem 3: There are at least 13 differentially/linearly active S-boxes in 5 consecutive rounds.

Proof: Assume two active eq. classes are $\mathcal{E}_j$ and $\mathcal{E}_{j+2}$ or $\mathcal{E}_j$ and $\mathcal{E}_{j+6}$. Then, we have 4 active eq. classes in the next round. They are either $\{\mathcal{E}_0, \mathcal{E}_1, \mathcal{E}_4, \mathcal{E}_5\}$ or $\{\mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_6, \mathcal{E}_7\}$. Without loss of generality, assume $\{\mathcal{E}_0, \mathcal{E}_1, \mathcal{E}_4, \mathcal{E}_5\}$ are active. Then, at least one in $\{\mathcal{E}_0, \mathcal{E}_1\}$ and at least one in $\{\mathcal{E}_2, \mathcal{E}_3\}$ are active in the forthcoming round. If four of them are active then, all the eq. classes will be active in the next round. If three of them are active then 6 of the eq. classes will be active in the next round. So, assume two of them active. If they are odd and even ($\{\mathcal{E}_0, \mathcal{E}_3\}$ or $\{\mathcal{E}_1, \mathcal{E}_2\}$) then again we have 4 and then 8 active eq. classes in next two rounds by Proposition 3. So, in the worst case $\{\mathcal{E}_0, \mathcal{E}_2\}$ or $\{\mathcal{E}_1, \mathcal{E}_3\}$ are active. These two active eq. classes will again produce at least 4 active eq. classes. So, it will go on as 2-4-2-4... in the worst case. If there are one or two active classes in the fourth round, this time, use M^{-1} and repeat the argument above for the decryption to show there are at least 13 differentially active classes. Other cases are already treated in the previous statements. ■

Theorem 3 gives practical security limits for the small number of rounds. In addition, Theorem 4 shows that any matrix providing SDD can constitute a very fast diffusion layer.

Theorem 4: There are at least 24 differentially/linearly active S-boxes in 8 consecutive rounds.

Proof: If all the pairs of two consecutive rounds $(2i-1, 2i)$ for $i = 1, 2, 3, 4$ contain at least one round having more than two active eq. classes, then $(2i-1, 2i)$ contains at least 6 active eq. classes by using Lemma 3 both in encryption and decryption. So, assume one of the pairs $(2i-1, 2i)$ contains 2-2, 2-1, or 1-2 active classes. Then 2-2 will extend to 2-2-4-8-4 in both encryption and decryption directions. Similarly, 2-1 will extend 2-1-2-4-8-4 in the encryption direction and 1-2-4-8-4 in the decryption direction by Theorem 2. In the worst case, one active class is in the beginning or at the end. So, the worst case is 1-2-4-8-4-2-1-2 or 2-1-2-4-8-4-2-1. In all cases, we have at least 24 active S-boxes. ■

We can further improve the diffusion property of the matrices.

Definition 2: Assume M provides SDD. Then, M provides SDD completely if it satisfies the following conditions.

- If one of $\mathcal{E}_{2i}$ or $\mathcal{E}_{2i+1}$ as outputs of M is differentially passive then the other contains at least two differentially active S-boxes for $i = 0, 1, 2, 3$.
- If one of $\mathcal{E}_i$ or $\mathcal{E}_{i+4}$ as outputs of M^{-1} is linearly passive then the other contains at least two linearly active S-boxes for $i = 0, 1, 2, 3$.

The matrices providing SDD completely propagate the number of active S-boxes faster.

Theorem 5: If M provides SDD completely, then there are at least 16 differentially/linearly active S-boxes in 5 rounds.

Proof: Consider the 2nd and 4th rounds. If there are *-3-*-3-* or *-3-*-4-* active classes then there are at least 6-3-3-3-6 or *-3-3-4-8 active classes respectively by Lemma 4. If there are more than 3 active classes in the 2nd and 4th rounds, then there are at least 4 active S-boxes both in the third and in the last rounds. If there is one active S-box in the 2nd or 4th round then there are at least 2-2-2-4-8 or 8-8-4-2-2 active S-boxes. If there are 2 active S-boxes in the second round then we have either 4-2-4-2-4 active classes with at least 4-4-4-4-4 active S-boxes or 4-2-1-2-4 active classes with at least 4-4-2-2-4 active S-boxes in the worst case. ■

Theorem 6: If M provides SDD completely, then there are at least 30 differentially/linearly active S-boxes in 8 rounds.

Proof: If there is one active eq. class in a round, then there are 1-2-4-8-4-2-1-2 active eq. classes and 1-2-4-8-8-4-2-2 active S-boxes in the worst case. For the case of two active eq. classes in a round, we have 2-4-2-4-2-4-2-4 active eq. classes in the worst case. Then there are at least 2-4-4-4-4-4-4-4 active S-boxes. Recall that case 2-2 will extend to 2-2-4-8-4 both in the encryption and in the decryption direction. If there are 3 differentially active eq. classes in a round then the forthcoming five rounds will have at least 3-3-3-6-3, 3-3-4-8-4, 3-4-8-4-4, or 3-6-6-3-3 active eq. classes both in the encryption and the decryption direction by Lemma 4. Then there at least 3-4-4-6-6, 3-4-4-8-8, 3-4-8-8-4, and 3-6-6-6-4 differentially active S-boxes in 5 rounds. We have at least 17 active S-boxes in 4 rounds for these cases. ■

B. Properties of Our Matrices

We minimize the number of XOR operations in our matrices. They do not provide SDD by themselves. However, we still take advantage by complying with some of the criteria in Definition 1 (the first condition) and we list the similar diffusion properties as follows.

- The first three bits of each s-bit word have two copies and the last two bits are single. Moreover, a single bit is not XORed with another single bit.
- Both matrices provide SDD completely for the characteristics whose single bits in each s-bit word are either both active or both passive.
- Single bits of an s-bit word go to different classes and come from different classes.

TABLE I

THE BUILDING BLOCKS OF DIZY

Cipher	DIZY-80	DIZY-128
Internal state size	120 bit	160 bit
S-box (word) size & numbers	5 bit & 24	5 bit & 32
Matrix size & numbers	30 × 30 & 4	40 × 40 & 4
Subblock size & numbers	15 bit & 8	20 bit & 8
Number of rounds	15	15

- Four input bits of any two XORs come from at least three different s-bit words.
- (40-bit) The output bits of each s-bit word are sent to 8 different s-bit words and the input bits in an s-bit word comes from 8 different s-bit words.
- (40-bit) If two bits go to same s-bit word, their copies go to different s-bit words.

V. DESCRIPTIONS OF NEW DESIGNS

a) High Level Description: We call our algorithms as DIZY-80 for 80-bit key and DIZY-128 for 128-bit key. The state update function is an SPN structure having 15 rounds. Its one round consists of constant additions, an S-box layer, 4 binary matrix multiplications with M and the permutation of eight subblocks respectively. Each subblock defines an eq. class. Table I depicts the sizes and the numbers of the parameters.

b) Round function: We call each round of the state update function the round function. The round function starts with the addition of a constant. The i-th round constant is XORed to the least significant four bits of each $s = 5$-bit word for the round number $i = 1, \ldots, 15$, which is produced by clocking the LFSR whose characteristic polynomial is $x^4 + x + 1$, i times starting from the initial value $(1, 0, 0, 0)$. Then, 24 and 32 S-box operations are performed in parallel for DIZY-80 and DIZY-128 respectively.The next step consists of the multiplications of 4 binary matrices with each 30-bit or 40-bit parts of the internal state for DIZY-80 or DIZY-128 respectively. We call each matrix M. Last, we permute eight subblocks as $(0, 4, 1, 5, 2, 6, 3, 7)$. Let us remark that all the statements in Section IV are invariant with respect to the order of the matrix multiplication and the permutation.

c) Output function: After 15 iterations of the round function, the 32 bits of the internal state are given as the output. These output bits are the least significant $3i$-th bitsof the internal state for $i = 0, \ldots, 31$. The maximum of the total lengths of keystreams is 2^{32} blocks which are 2^{37} bits per one key.

d) Initialization Phase: Initially, the state is null. First of all, the most significant 48\64 bits of the key are XORed with the 48-bit\64-bit part of the internal state which is formed by concatenating the most significant two bits of each 5-bit word in order, for DIZY-80\128 respectively. Each two-bit value then forms the most significant two bits of the corresponding 5-bit word of the internal state. As the next step, the round function is run once and the remaining key bits are incorporated similarly: The least significant 32\64 key bits are XORed two by two with the most significant two

TABLE II

M FOR DIZY-80

6,22	16,18	0,18	15	1
7,27	20,13	2,23	21	11
12,17	26,3	10,28	25	5
2,27	21,9	11,24	16	0
7,17	15,14	1,29	20	10
12,25	22,4	5,19	26	6

TABLE III

M FOR DIZY-128

22,2	35,8	16,28	10	31
20,7	27,13	1,33	15	36
25,12	32,18	6,38	0	21
37,17	26,3	11,23	5	30
27,17	30,9	0,24	11	35
32,2	36,14	5,29	16	20
37,7	21,19	10,34	1	25
22,12	31,4	15,39	6	26

bits of the 5-bit words of the most significant 120\160-bit state in the second round for DIZY-80\128 respectively. The round function is run 14 times without incorporating any input afterward. The key initialization step is completed at the end of the 15th round. The IV is incorporated with the same procedure of key initialization and this step is completed in 15 more rounds. That is, after 30 rounds in total in the initialization phase, the first internal state is obtained.

***S*-box:** (00, 04, 0e, 09, 0d, 0b, 1e, 1b, 1c, 14, 13, 18, 17, 1d, 05, 0c, 0f, 11, 08, 15, 03, 1f, 19, 06, 10, 02, 16, 07, 1a, 0a, 01, 12)

M Matrices: The matrices are given in Table II and Table III. Each row in Table II and Table III contains the outputs of a 5-bit word. The bit numbers stored in the boxes are the indices and two indices in a box are XORed. For instance, the most significant bit of the 2nd word is $x_7 \oplus x_{27}$ for DIZY-80.

VI. Security Analysis

The bounds on the advantages of distinguishing the update functions are $2^{-41}\backslash 2^{-72}$ and $2^{-81}\backslash 2^{-112}$ for DIZY-80\128 respectively in Game A\B in Section III. We adopt a 112-bit security level for DIZY-128 in compliance with the NIST security requirements for the competition of lightweight algorithms: "An AEAD algorithm shall not specify key lengths that are smaller than 128 bits. Cryptanalytic attacks on the AEAD algorithm shall require at least 2^{112} computations" [55].

The complexity of the offline phases of the tradeoff attacks recovering internal states is N/D in [11], [12], [13] where $N = 2^{120}\backslash 2^{160}$ for DIZY-80\128 respectively. Data is limited to $D = 2^{32}$. Therefore, the complexities in the offline phases are $2^{88}\backslash 2^{128}$ for DIZY-80\128 respectively. The time and memory complexities in the online phase are $(N/D)^{2/3}$ [11], [12], [13]. Hence, the time and memory complexities are $2^{59}\backslash 2^{85}$ for DIZY-80\128 respectively.

A. Key and IV Collisions

The seed (the first internal state) after the key and IV initialization is a one-to-one function of the key for a fixed IV.

Proposition 4: Two different keys produce different seeds.

Proof: Any difference in key bits produces a pair in the structure (00000, 01000, 10000, 11000) in the first round. The S-box send these vectors to (**00**000, **11**100, **01**111, **10**000). The first two bits take all the values 0,1,2 and 3 where there is at least one bit difference. But these bits have copies having no XOR in the M matrices. So, any other difference has no effect on these single bits. Therefore a nonzero difference still stays in the second round. But, since there is no key bit incorporation in the least significant two bits, a nonzero difference occurs at the end of the second round. ■

It is much more difficult to give a result like Proposition 4 for different IVs. It is also computationally infeasible to find IV_1 and IV_2 for two different keys K_1 and K_2 such that (K_1, IV_1) and (K_2, IV_2) produce the same keystream sequence.

B. Period

It is not easy to give tight lower bounds for the periods. However, the expected periods of the update functions are $2^{119}\backslash 2^{159}$ for DIZY-80\128 respectively. The probability that the period of a keystream is less than 2^{32} is less than $2^{-88}\backslash 2^{-128}$ for DIZY-80\128 respectively.

C. Correlation and Algebraic Attacks

We expect the next state functions of DIZY to be pseudorandom permutations and the output functions (TPP) to be pseudorandom functions. So, there must be no correlation or algebraic relation between the internal state and output bits.

D. Guess and Determine Attacks

An SISS cipher may easily be prone to guess-and-determine attacks due to its small state. This is why we adopt the design approach of utilizing pseudorandom permutations as state update functions. The trivial guess-and-determine attack is to guess $n - w$ most significant bits and determine the initial state from the w bit output. So, $k + w \leq n$. We conjecture that there is no guess-and-determine attack faster than the trivial one since the state update function is a pseudorandom permutation. It is computationally infeasible to recover the input of a pseudorandom permutation from its partial output.

E. Active S-Boxes

If there are differences in the first three output bits of the differentially active S-boxes or if there is a linear mask in the first three input bits of the linearly active S-boxes then our both M matrices behave as if they provide SDD and all the statements in Section IV will be valid for such characteristics. With our S-box, we can assume that M provides SDD if there is a one-bit input difference of an S-box. Because we have either the last two bits or at least one bit in the first three bits active. Then, M diffuses these differences into two different classes. So, we can achieve a high level of diffusion which is as fast as those of the linear layers providing SDD.

Theorem 7: There are at least 20\22 active S-boxes in any eight consecutive rounds of DIZY-80\128 respectively.
The proof makes use of the properties in Section IV-B and examines similar cases as in the statements in Section IV. We skip the proof for the sake of space.

F. Truncated Differential Attack

Truncated differential attacks on SPN ciphers are generally based on the probabilities of the activeness of S-boxes [56]. If an S-box is active then 6 S-boxes will probably be active (8 S-boxes for the 128-bit case). The probability that any output bit of an active S-box is active is 16/31 for randomly chosen input difference since there are 31 nonzero vectors and each coordinate takes 16 1's and 15 0's. This probability is one-half if at least one output bit is active. Similarly, if i output bits are passive then one of the remaining output bits is active with probability $2^{4-i}/(2^{5-i}-1)$.

We examine the 128-bit case for the sake of simplicity. We have $p_{i+1} = \sum_{\ell=0}^{3}\binom{3}{\ell}\left(\frac{15+16p_i}{31}\right)^{8-2\ell}\left(\frac{16-16p_i}{31}\right)^{2\ell}$ where p_i is the probability that an S-box is passive in the i-th round.Then, $p_{i+1} = 31^{-8}(15+16p_i)^2((15+16p_i)^2 + (16-16p_i)^2)^3$. Note that $\lim_{i\to\infty} p_i = 2^{-5}$. There are dependencies between the activeness of the S-boxes in the eq. classes $\mathcal{E}_i$ and $\mathcal{E}_{i+4}$ which complicate the computations.

G. Integral Attack

Any activeness corresponds to a permutation in an integral attack [57]. If one eq. class is active, for a matrix satisfying SDD then all the eq. classes are active after three rounds. So, any eq. class will be balanced after four rounds and each vector will appear an even number of times. A bijective S-box layer produces a balanced set from this set, which introduces a five-round distinguisher for the integral characteristic. We can add two more rounds in the decryption direction. So, integral sets of order 2, that is, 4 active eq. classes (the classes 0-2-4-6 or 1-3-5-7) taking all the $2^{n/2}$ values produce integral sets of order 1 in the next round and active eq. classes after two rounds. Therefore, 4 active eq. classes produce balanced sets in each eq. classes in 7 rounds. This 7-round characteristic is not valid for our matrices since they do not provide SDD. Eventually, we do not expect any integral characteristic for 8 or more rounds.

H. Impossible Differentials

Any input/output difference that never occurs forms an impossible differential [58]. The straightforward impossible differential characteristic is given for $3+2=5$ rounds as$1-2-4-8$ active eq. classes in 3 rounds in the encryption direction and $1-2-4$ eq. classes in 2 rounds in the decryption direction for the matrices providing SDD. In our case, we have at least $1-2-3-5$ active eq. classes for a one-bit difference. So, there are at least 5 active eq. classes. On the other hand, there are at most 4 active eq. classes in the decryption direction if we have one active eq. class. So, we obtain a contradiction in the middle, which simply implies that 1 active bit produces at least two active eq. classes in the fifth round. Since the algorithms provide the SAC property in 5 rounds and we use bitwise linear transformations, we do not expect any impossible differential characteristic in 10 rounds.

I. Boomerang Attack

A boomerang attack is a meet-in-the-middle attack [59] The straightforward characteristic is 1-1-2-3-5 active S-boxes in the encryption direction and 1-1-2-3 active S-boxes in the decryption direction. Using the ladder switch method, it is possible to bypass 5 active S-boxes in the fourth round. Hence $p = q = 2^{-28}$ and $p^2q^2 = 2^{-112}$ for 7 rounds. We do not expect any boomerang characteristic for 8 or more rounds.

TABLE IV

Frequencies Are 10 MHz Except Fruit [22]. We Use the Extension Ht for Our Parallel Implementations

SISS C.	CMOS nm	Area GE	Power µW	Thrpt. Mbit/sec	Energy pJ/bit
DIZY-80	65	**1010**	23.8	0.49	48.57
DIZY-128	65	**1297**	32.5	0.38	85.53
DIZY-80-ht	65	2037	58.1	14.22	**4.09**
DIZY-128-ht	65	2686	75.7	14.22	**5.32**
Lizard [1]	90	1481	51.8	1.99	25.99
LILLE-40 [1]	90	911	35.0	0.56	62.5
Fruit-F [22]	180	965	NA	NA	NA
Plantlet [1]	90	886	35.4	1.66	21.32
Stream C.					
Trivium [1]	90	1870	78.4	0.53	148.87
Grain-v1 [1]	90	1164	40.6	2.85	14.26
Grain-128 [1]	90	1700	71.5	2.00	35.73
Block C.					
Present-80 [1]	90	1440	52.2	19.41	2.69
Midori64 [1]	90	1542	60.6	37.6	1.61

VII. Implementation Results

We have implemented DIZY both in FPGA and ASIC[1] . The performance results are depicted in Table IV. The numbers for the other ciphers are from [1]. We do not compare with the NIST lightweight algorithm submissions in [55] since they are authenticated encryptions and their hardware area costs and energy consumption are much higher.

We have implemented the algorithms in VHDL using the Vivado 2017.3 Webpack tool. The serial versions cost (151,131)\(187,172) (LUTS, FFs) and the parallel versions cost (234,124)\(305,164) (LUTS, FFs) for DIZY-80\128 respectively in Spartan-7. The logic-level representations of the algorithms have been generated utilizing Cadence Genus. The ASIC implementations have been executed using the standard cell library of the TSMC 65 nm CMOS process, with a driving voltage of 1V and a clock frequency of 10 MHz. The computation of Gate Equivalence (GE) numbers has been performed by dividing the total area by the smallest 2-input NAND gate in the cell library.

Our results are compared to the algorithms presented in [1], which employ a round-based implementation and utilize regular flip flops. To maintain compatibility with the performances in [1], 64-bit frames are generated at a frequency of 10 MHz. The energy consumption values for 64-bit blocks, as reported in [1], are presented in Table IV. We should note that there are finer results as 1161 GE for Lizard and 1075 GE for Present-80. However, we consider [1] to compare the energy consumption fairly.

In order to optimize energy efficiency during frame processing, the IV initialization is selected as the starting point. That is, the key initialization is performed offline to conserve

[1] https://github.com/cagdasgAbuHafs/DIZY-cipher.git

cycles during each frame. The DIZY-ht implementations utilize parallel implementation of the S-boxes and M matrices, resulting in an efficient generation of a 64-bit block in only 45 cycles.

In addition to the data presented in Table IV, estimated power consumption results at 100 KHz intervals are also available for consideration in the design of extremely power-constrained devices. The estimated values for the DIZY-80\80-ht\128\128-ht algorithms are 5.60\10.26\7.27\13.75 ÂB5W respectively.

VIII. Conclusion

We address the issue of state size limitations for keystream generators imposed by tradeoff attacks. We introduce a new construction method of diffusion layers for SPN structures by utilizing matrices providing SDD and a new stream cipher mode which we call a TPP mode. We analyze both the diffusion layer and the mode intensively by proving several security statements. We construct an SISS cipher which we call DIZY, to exemplify our constructions and give its security analyses. Additionally, we present a hardware implementation of our proposed cipher and exhibit its effectiveness. Our results indicate that it is one of the most efficient lightweight ciphers available, with a hardware area cost of approximately 1.3K GE for the DIZY-128 version, making it one of the smallest symmetric ciphers with a 128-bit key. We posit that future advancements in the construction of SISS ciphers have the potential to further improve performance results. Moreover, we keep the number of eq. classes in our theoretical infrastructure as 8. Further study can focus on generalizing statements for 2^m eq. classes for any positive integer $m \in \mathbf{Z}^+$.

Acknowledgment

The authors would like to thank their colleagues from TUTEL Laboratory for guiding them in Cadence, and also would like to thank the anonymous reviewers, F. Birinci, N. F. Er, F. Karakoç, F. Kazancı, M. S. Kiraz, E. Kükkubaş A. A. Selçuk, E. Ulusoy, and B. Ustaoğlu for their invaluable comments.

References

[1] S. Banik et al., “Towards low energy stream ciphers,” *IACR Trans. Symmetric Cryptol.*, vol. 2018, no. 2, pp. 1–19, 2018, doi: 10.13154/tosc.v2018.i2.1-19.

[2] B. Zhang, “Cryptanalysis of GSM encryption in 2G/3G networks without rainbow tables,” in *Advances in Cryptology—ASIACRYPT 2019* (Lecture Notes in Computer Science), vol. 11923. Springer, 2019, pp. 428–456, doi: 10.1007/978-3-030-34618-8_15.

[3] Y. Shaked and A. Wool, “Cryptanalysis of the Bluetooth E_0 cipher using OBDD’s,” in *Information Security* (Lecture Notes in Computer Science), vol. 4176. Springer, 2006, pp. 187–202, doi: 10.1007/11836810_14.

[4] R. Verdult, F. D. Garcia, and J. Balasch, “Gone in 360 seconds: Hijacking with Hitag2,” in *Proc. USENIX*, 2012, pp. 237–252.

[5] W. Aerts et al., “A practical attack on KeeLoq,” *J. Cryptol.*, vol. 25, no. 1, pp. 136–157, Jan. 2012, doi: 10.1007/s00145-010-9091-9.

[6] C. Meijer and R. Verdult, “Ciphertext-only cryptanalysis on hardened Mifare classic cards,” in *Proc. SIGSAC*, 2015, pp. 18–30, doi: 10.1145/2810103.2813641.

[7] C. Tezcan, “Brute force cryptanalysis of MIFARE classic cards on GPU,” in *Proc. 3rd Int. Conf. Inf. Syst. Secur. Privacy*, 2017, pp. 524–528, doi: 10.5220/0006262705240528.

[8] I. Mantin, “A practical attack on the fixed RC4 in the WEP mode,” in *Advances in Cryptology—ASIACRYPT 2005* (Lecture Notes in Computer Science), vol. 3788. Springer, 2005, pp. 395–411, doi: 10.1007/11593447_21.

[9] M. Morii and Y. Todo, “Cryptanalysis for RC4 and breaking WEP/WPA-TKIP,” *IEICE Trans. Inf. Syst.*, vol. 94, no. 11, pp. 2087–2094, 2011, doi: 10.1587/transinf.E94.D.2087.

[10] F. Armknecht and V. Mikhalev, “On lightweight stream ciphers with shorter internal states,” in *Fast Software Encryption* (Lecture Notes in Computer Science), vol. 9054. Springer, 2015, pp. 451–470, doi: 10.1007/978-3-662-48116-5_22.

[11] A. Biryukov and A. Shamir, “Cryptanalytic time/memory/data tradeoffs for stream ciphers,” in *Advances in Cryptology—ASIACRYPT 2000* (Lecture Notes in Computer Science), vol. 1976. Berlin, Germany: Springer, 2000, pp. 1–13, doi: 10.1007/3-540-44448-3_1.

[12] S. Babbage, “Improved ‘exhaustive search’ attacks on stream ciphers,” in *Proc. Eur. Conv. Secur. Detection*, May 1995, pp. 161–166.

[13] J. D. Golić, “Cryptanalysis of alleged A5 stream cipher,” in *Proc. EUROCRYPT*, in Lecture Notes in Computer Science, vol. 1233. Cham, Switzerland: Springer, 1997, pp. 239–255.

[14] V. Lallemand and M. Naya-Plasencia, “Cryptanalysis of full Sprout,” in *Proc. CRYPTO*, in Lecture Notes in Computer Science, vol. 9215. Cham, Switzerland: Springer, 2015, pp. 663–682, doi: 10.1007/978-3-662-47989-6_32.

[15] S. Maitra, S. Sarkar, A. Baksi, and P. Dey, “Key recovery from state information of Sprout: Application to cryptanalysis and fault attack,” *IACR Cryptol. ePrint Arch.*, vol. 10, p. 236, Mar. 2015. [Online]. Available: http://eprint.iacr.org/2015/236

[16] B. Zhang and X. Gong, “Another tradeoff attack on Sprout-like stream ciphers,” in *Advances in Cryptology—ASIACRYPT 2015* (Lecture Notes in Computer Science), vol. 9453. Springer, 2015, pp. 561–585, doi: 10.1007/978-3-662-48800-3_23.

[17] M. F. Esgin and O. Kara, “Cryptanalysis of full Sprout with TMD tradeoff attacks,” in *Selected Areas in Cryptography—SAC 2015* (Lecture Notes in Computer Science), vol. 9566. Springer, 2015, pp. 67–85, doi: 10.1007/978-3-319-31301-6_4.

[18] V. Mikhalev, F. Armknecht, and C. Müller, “On ciphers that continuously access the non-volatile key,” *IACR Trans. Symmetric Cryptol.*, vol. 2016, no. 2, pp. 52–79, 2016, doi: 10.13154/tosc.v2016.i2.52-79.

[19] Y. Todo, W. Meier, and K. Aoki, “On the data limitation of small-state stream ciphers: Correlation attacks on fruit-80 and plantlet,” in *Selected Areas in Cryptography—SAC 2019* (Lecture Notes in Computer Science), vol. 11959. Springer, 2019, pp. 365–392, doi: 10.1007/978-3-030-38471-5_15.

[20] S. Wang, M. Liu, D. Lin, and L. Ma, “Fast correlation attacks on Grain-like small state stream ciphers and cryptanalysis of plantlet, fruit-v2 and fruit-80,” *IACR Cryptol. ePrint Arch.*, vol. 2019, p. 763, Jan. 2019. [Online]. Available: https://eprint.iacr.org/2019/763

[21] S. Banik, K. Barooti, and T. Isobe, “Cryptanalysis of plantlet,” *IACR Trans. Symmetric Cryptol.*, vol. 2019, no. 3, pp. 103–120, 2019, doi: 10.13154/tosc.v2019.i3.103-120.

[22] V. Amin-Ghafari, F. Lin, and Z. Zhou, “A new idea in response to fast correlation attacks on small-state stream ciphers,” *Microprocessors Microsyst.*, vol. 96, Feb. 2023, Art. no. 104720, doi: 10.1016/j.micpro.2022.104720.

[23] S. Banik, T. Isobe, and M. Morii, “On design of robust lightweight stream cipher with short internal state,” *IEICE Trans. Fundamentals Electron., Commun. Comput. Sci.*, vol. 101, no. 1, pp. 99–109, 2018, doi: 10.1587/transfun.E101.A.99.

[24] H. Hu, “Fruit-80: A secure ultra-lightweight stream cipher for constrained environments,” *Entropy*, vol. 20, no. 3, p. 180, Mar. 2018, doi: 10.3390/e20030180.

[25] V. Amin-Ghafari, M. A. Orumiehchiha, and S. Rostami, “An attack on the LILLE stream cipher,” *IACR Cryptol. ePrint Arch.*, vol. 15, p. 111, Mar. 2023. [Online]. Available: https://eprint.iacr.org/2023/111

[26] A. Bogdanov et al., “PRESENT: An ultra-lightweight block cipher,” in *Proc. CHES*, in Lecture Notes in Computer Science, vol. 4727. Cham, Switzerland: Springer, 2007, pp. 450–466, doi: 10.1007/978-3-540-74735-2_31.

[27] G. Yang, B. Zhu, V. Suder, M. D. Aagaard, and G. Gong, “The Simeck family of lightweight block ciphers,” in *Proc. CHES*, in Lecture Notes in Computer Science, vol. 9293. Cham, Switzerland: Springer, 2015, pp. 307–329, doi: 10.1007/978-3-662-48324-4_16.

[28] J. Guo, T. Peyrin, A. Poschmann, and M. J. B. Robshaw, "The LED block cipher," in *Proc. CHES*, in Lecture Notes in Computer Science, vol. 6917. Cham, Switzerland: Springer, 2011, pp. 326–341, doi: 10.1007/978-3-642-23951-9_22.

[29] A. Poschmann, S. Ling, and H. Wang, "256 bit standardized crypto for 650 GE—GOST revisited," in *Cryptographic Hardware and Embedded Systems, CHES 2010* (Lecture Notes in Computer Science), vol. 6225. Springer, 2010, pp. 219–233, doi: 10.1007/978-3-642-15031-9_15.

[30] S. Banik et al., "*Midori*: A block cipher for low energy," in *Advances in Cryptology—ASIACRYPT 2015* (Lecture Notes in Computer Science), vol. 9453. Springer, 2015, pp. 411–436, doi: 10.1007/978-3-662-48800-3_17.

[31] M. Hamann, M. Krause, and W. Meier, "LIZARD—A lightweight stream cipher for power-constrained devices," *IACR Trans. Symmetric Cryptol.*, vol. 2017, no. 1, pp. 45–79, 2017, doi: 10.13154/tosc.v2017.i1.45-79.

[32] D. K. Dalai, S. Pal, and S. Sarkar, "A state bit recovery algorithm with TMDTO attack on lizard and grain-128A," *Des., Codes Cryptogr.*, vol. 90, no. 3, pp. 489–521, Mar. 2022, doi: 10.1007/s10623-021-00984-3.

[33] A. J. Stam, "Distance between sampling with and without replacement," *Statistica Neerlandica*, vol. 32, no. 2, pp. 81–91, Jun. 1978.

[34] C. Hall, D. A. Wagner, J. Kelsey, and B. Schneier, "Building PRFs from PRPs," in *Proc. CRYPTO*, in Lecture Notes in Computer Science, vol. 1462. Cham, Switzerland: Springer, 1998, pp. 370–389, doi: 10.1007/BFb0055742.

[35] M. Bellare, T. Krovetz, and P. Rogaway, "Luby-Rackoff backwards: Increasing security by making block ciphers non-invertible," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science), vol. 1403. Springer, 1998, pp. 266–280, doi: 10.1007/BFb0054132.

[36] L. Grassi and B. Mennink, "Security of truncated permutation without initial value," *IACR Cryptol. ePrint Arch.*, vol. 2022, p. 508, May 2022. [Online]. Available: https://eprint.iacr.org/2022/508

[37] W. Choi, B. Lee, and J. Lee, "Indifferentiability of truncated random permutations," in *Advances in Cryptology—ASIACRYPT 2019* (Lecture Notes in Computer Science), vol. 11921. Springer, 2019, pp. 175–195, doi: 10.1007/978-3-030-34578-5_7.

[38] Y. Dodis, L. Reyzin, R. L. Rivest, and E. Shen, "Indifferentiability of permutation-based compression functions and tree-based modes of operation, with applications to MD6," in *Fast Software Encryption* (Lecture Notes in Computer Science), vol. 5665. Springer, 2009, pp. 104–121, doi: 10.1007/978-3-642-03317-9_7.

[39] S. Gilboa, S. Gueron, and B. Morris, "How many queries are needed to distinguish a truncated random permutation from a random function?" *J. Cryptol.*, vol. 31, no. 1, pp. 162–171, Jan. 2018, doi: 10.1007/s00145-017-9253-0.

[40] B. Mennink, "Linking Stam's bounds with generalized truncation," in *Topics in Cryptology—CT-RSA 2019* (Lecture Notes in Computer Science), vol. 11405. Springer, 2019, pp. 313–329, doi: 10.1007/978-3-030-12612-4_16.

[41] D. A. McGrew and J. Viega, "The security and performance of the Galois/counter mode (GCM) of operation," in *Progress in Cryptology—INDOCRYPT 2004* (Lecture Notes in Computer Science), vol. 3348. Springer, 2004, pp. 343–355.

[42] S. Gueron and Y. Lindell, "GCM-SIV: Full nonce misuse-resistant authenticated encryption at under one cycle per byte," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2015, pp. 109–119, doi: 10.1145/2810103.2813613.

[43] B. Cogliati and Y. Seurin, "EWCDM: An efficient, beyond-birthday secure, nonce-misuse resistant MAC," in *Advances in Cryptology—CRYPTO 2016* (Lecture Notes in Computer Science), vol. 9814. Springer, 2016, pp. 121–149, doi: 10.1007/978-3-662-53018-4_5.

[44] B. Mennink and N. Samuel, "Encrypted and its dual: Towards optimal security using mirror theory," in *Advances in Cryptology—CRYPTO 2017* (Lecture Notes in Computer Science), vol. 10403. Springer, 2017, pp. 556–583, doi: 10.1007/978-3-319-63697-9_19.

[45] P. Rogaway, "Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC," in *Advances in Cryptology—ASIACRYPT* (Lecture Notes in Computer Science), vol. 3329. Germany: Springer, 2004, pp. 16–31.

[46] T. Iwata, "New blockcipher modes of operation with beyond the birthday bound security," in *Fast Software Encryption* (Lecture Notes in Computer Science), vol. 4047. Springer, 2006, pp. 310–327.

[47] B. Mennink and S. Neves, "Optimal PRFs from blockcipher designs," *IACR Trans. Symmetric Cryptol.*, vol. 2017, no. 3, pp. 228–252, 2017, doi: 10.13154/tosc.v2017.i3.228-252.

[48] E. Biham and A. Shamir, "Differential cryptanalysis of the full 16-round DES," in *Advances in Cryptology—CRYPTO* (Lecture Notes in Computer Science), vol. 740. Springer, 1992, pp. 487–496, doi: 10.1007/3-540-48071-4_34.

[49] M. Matsui, "Linear cryptanalysis method for DES cipher," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science), vol. 765. Springer, 1993, pp. 386–397, doi: 10.1007/3-540-48285-7_33.

[50] M. Hellman, "A cryptanalytic time-memory trade-off," *IEEE Trans. Inf. Theory*, vol. IT-26, no. 4, pp. 401–406, Jul. 1980.

[51] P. Oechslin, "Making a faster cryptanalytic time-memory trade-off," in *Advances in Cryptology—CRYPTO 2003* (Lecture Notes in Computer Science), vol. 2729. Springer, 2003, pp. 617–630, doi: 10.1007/978-3-540-45146-4_36.

[52] A. Mileva, V. Dimitrova, O. Kara, and M. J. Mihaljević, "Catalog and illustrative examples of lightweight cryptographic primitives," in *Security of Ubiquitous Computing Systems*. Cham, Switzerland: Springer, 2021, pp. 21–47, doi: 10.1007/978-3-030-10591-4_2.

[53] O. Kara, "Tradeoff attacks on symmetric ciphers," in *Cryptography—Recent Advances and Future Developments*. London, U.K.: IntechOpen, 2021. [Online]. Available: https://www.intechopen.com/online-first/tradeoff-attacks-on-symmetric-ciphers

[54] C. D. Cannière and B. Preneel, "Trivium," in *New Stream Cipher Designs* (Lecture Notes in Computer Science), vol. 4986. Springer, 2008, pp. 244–266, doi: 10.1007/978-3-540-68351-3_18.

[55] NIST Ligthweight Cryptography Standardization Project. *NIST Call of Algorithms*. [Online]. Available: https://csrc.nist.gov/Projects/lightweight-cryptography

[56] L. R. Knudsen, "Truncated and higher order differentials," in *Fast Software Encryption* (Lecture Notes in Computer Science), vol. 1008. Springer, 1994, pp. 196–211, doi: 10.1007/3-540-60590-8_16.

[57] L. R. Knudsen and D. A. Wagner, "Integral cryptanalysis," in *Fast Software Encryption* (Lecture Notes in Computer Science), vol. 2365. Springer, 2002, pp. 112–127, doi: 10.1007/3-540-45661-9_9.

[58] E. Biham, A. Biryukov, and A. Shamir, "Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science), vol. 1592. Springer, 1999, pp. 12–23, doi: 10.1007/3-540-48910-X_2.

[59] D. A. Wagner, "The boomerang attack," in *Fast Software Encryption* (Lecture Notes in Computer Science), vol. 1636. Springer, 1999, pp. 156–170, doi: 10.1007/3-540-48519-8_12.

Çağdaş Gül received the master's degree in information security engineering from Marmara University. He is currently a Researcher with TÜBLGEM.

Orhun Kara received the Ph.D. degree from Bilkent University. He is currently an Associate Professor with the Department of Mathematics, İzmir Institute of Technology, and also a Researcher with LGEM.