LEANY: A Hardware-Friendly Wide-Block Block Cipher

Gökçe Düzyol, Nida Fidan, Kamil Otal, and Selman Pınar

TÜBİTAK BİLGEM National Research Institute of Electronics and Cryptology
Gebze, Kocaeli, Türkiye
{gokce.duzyol, nida.fidan, kamil.otal, selman.pinar}@tubitak.gov.tr

Abstract. Wide-block block ciphers have attracted significant interest due to emerging high-throughput encryption requirements and NIST's Accordion Mode Process. The primary choice for such ciphers is Rijndael-256, also known as WAES. At the same time, independent efforts continue to explore more efficient designs and implementations.
In this paper, we present LEANY, a 256-bit block cipher with a 256-bit key. LEANY is a nibble-oriented SPN whose state is an 8×8 matrix of 4-bit nibbles. Each round consists of four layers: key addition, a 4-bit S-box layer, row-wise nibble shifts, and multiplication by an 8×8 MDS matrix over GF(2^4). This structure follows the same shift-then-mix principle as AES, but with a wider state and a smaller cell size. LEANY consists of only 12 rounds.
Since the linear diffusion layer has branch number 9, the wide-trail strategy guarantees that every four consecutive rounds activate at least $9^2 = 81$ S-boxes in any differential or linear characteristic. Consequently, upper bounds on the best differential and linear trail probabilities fall below 2^{-256} after seven rounds. We further analyze LEANY against impossible differential, invariant subspace, integral, meet-in-the-middle, boomerang, and several algebraic attacks.
Finally, FPGA implementations demonstrate both compact and high-performance hardware realizations: the iterative architecture occupies only about 1.4K LUTs, whereas the fully pipelined architecture achieves a steady-state throughput exceeding 50 Gbps. Compared with existing 256-bit wide-block block ciphers, including WAES, LEANY provides substantially better area efficiency and competitive throughput (TP) and throughput-to-area ratio (TAR) results.

Keywords: Wide-block block cipher · Substitution–permutation network · Wide-trail strategy · MDS matrix · FPGA implementation.

1 Introduction

1.1 Motivation

Two largely independent trends are pushing symmetric cryptography toward primitives with blocks wider than the classical 128 bits. The first is the sheer

growth of data volumes and line rates. Network backbones, storage systems, and memory-encryption engines nowadays operate at tens to hundreds of gigabits per second, and a single key may protect amounts of data that were unthinkable when the AES was standardized. For a 128-bit block, the birthday bound of 2^{64} blocks is an intrinsic ceiling: widely deployed modes such as CTR, GCM, and XTS lose their security guarantees distinguishing advantages, leakage of plaintext equality, or degraded authenticity bounds well before the codebook is exhausted, which in practice forces conservative rekeying and data limits. Doubling the block size to 256 bits pushes this generic ceiling to 2^{128} blocks and effectively removes it from consideration for any realistic application. This observation is one of the driving forces behind NIST's ongoing Accordion Mode process [21], which explicitly considers Rijndael-256 as the underlying wide-block primitive, and behind the recent hardware-oriented study of that cipher under the name WAES [18].

The second trend is the migration to post-quantum security levels. Grover's algorithm [13] reduces the cost of exhaustive key search from 2^k to roughly $2^{k/2}$ quantum operations, and 256-bit keys have therefore become the default recommendation for long-term confidentiality. A 256-bit block is the natural companion of a 256-bit key: it keeps generic attacks that exploit the block size (multi-target trade-offs and collision-based attacks on modes) comfortably out of reach, and it yields a primitive whose effective security level is not silently degraded by the mode of operation built on top of it.

Rijndael-256/256 satisfies both requirements, but it inherits the byte-oriented structure of the AES, and this structure becomes expensive in hardware once the state doubles. Every round of the 256-bit variant evaluates thirty-two 8-bit S-boxes, each of which costs dozens of LUTs on an FPGA, so that recently reported fully pipelined implementations occupy more than 45,000 LUTs [18]. For area-constrained platforms that must nevertheless sustain high throughput—line-rate encryptors, storage controllers, or FPGA accelerators sharing the fabric with other logic—this cost is difficult to justify.

LEANY starts from the observation that the security argument of the AES, the wide-trail strategy [9], does not depend on the cell size: it only requires an S-box layer, a cell permutation that spreads the cells of each column over distinct columns, and a mixing matrix with a high branch number. LEANY therefore keeps the shift-then-mix structure of the AES but instantiates it with the smallest sensible cell, the nibble. In particular, the round function of LEANY resembles to the structure of the internal permutation P_{256} of the PHOTON hash function [14]. A cryptographically optimal 4-bit S-box maps directly onto the LUT fabric of an FPGA, an 8×8 MDS matrix over $\mathrm{GF}(2^4)$ reduces to a shallow XOR network, and a branch number of 9 yields at least 81 active S-boxes over every four rounds, strong enough that twelve rounds provide a comfortable security margin, provable by a simple counting argument and without any automated trail search. The key schedule is reduced to a single XOR with a rotating round constant, so the entire cipher fits in about 1.4K LUTs in an iterative implemen-

tation, while a pipelined version exceeds 50 Gbps. In short, LEANY is designed to offer Rijndael-256-like security guarantees at a fraction of the hardware cost.

1.2 Related Work

Wide-block block ciphers with a native 256-bit block size are comparatively uncommon in the literature; most well-known symmetric primitives fix their block at 128 bits or smaller, even when supporting a 256-bit key. Below, we summarize the design and structure of the 256-bit-block ciphers most relevant to LEANY, which we later use as points of comparison in Section 5 (hardware) and throughout our security analysis.

Rijndael-256/256. Rijndael, designed by Daemen and Rijmen [9], was submitted to NIST's AES competition supporting block and key sizes of 128, 192, and 256 bits in any combination; NIST's eventual FIPS-197 standard, however, fixed the block size at 128 bits, discarding the wider-block variants. The 256-bit-block, 256-bit-key configuration Rijndael-256/256 uses 14 rounds and an extended 4×8-byte state (as opposed to AES's 4×4), but otherwise follows the same SubBytes/ShiftRows/MixColumns/AddRoundKey structure. This variant has recently attracted renewed attention: NIST proposed standardizing it as a dedicated wide-block AES variant in 2024, and a hardware implementation under the name WAES-256 was recently reported by Malal and Tezcan [18].

Threefish-256. Threefish, designed by Ferguson et al. as the underlying tweakable block cipher of the Skein hash function [12], supports block/key sizes of 256, 512, and 1024 bits. Unlike LEANY and Rijndael-256/256, Threefish is an ARX design: rounds consist only of modular addition, XOR, and word rotation (no S-boxes), with a subkey injected every fourth round. Threefish-256 uses 72 rounds. Its lack of a substitution layer makes it structurally distinct from SPN designs like LEANY, trading table-based confusion for a purely arithmetic MIX operation.

Kalyna-256/256. Kalyna, designed by Oliynykov et al. [22], was adopted as the national encryption standard of Ukraine (DSTU 7624:2014) in 2015. It supports block and key sizes of 128, 256, and 512 bits (with the key size equal to or double the block size), and follows an explicitly Rijndael-like SPN structure, but with an enlarged MDS matrix, four distinct S-boxes (rather than AES's single S-box), and pre-/post-whitening performed with modulo-2^{64} addition rather than XOR. The Kalyna-256/256 configuration uses 14 rounds.

RC6 with a 256-bit block. RC6, designed by Rivest, Robshaw, Sidney, and Yin [23] as an AES-competition finalist, is parameterized as RC6-$w/r/b$, where w is the word size (the block consists of four w-bit words), r is the number of rounds, and b is the key length in bytes. The AES-submission configuration fixes $w = 32$ for a 128-bit block, but the general RC6 construction permits any word size; setting $w = 64$ yields a 256-bit block. RC6 is derived from RC5 and relies on data-dependent rotations and integer multiplication (rather than a fixed S-box)

for confusion, making it, like Threefish, structurally distinct from table-based SPN designs.

Saturnin. Saturnin, proposed by Canteaut et al. [5], is a 256-bit block cipher with a 256-bit key designed for post-quantum security and submitted to the NIST Lightweight Cryptography competition. Its 256-bit state is arranged as a $4 \times 4 \times 4$ array of 4-bit nibbles, making it a three-dimensional analogue of AES. In its Super-S-box representation, Saturnin behaves similarly to AES operating on 16-bit words, allowing its security analysis to leverage the wide-trail strategy. The cipher employs lightweight 4-bit S-boxes and an efficient MDS matrix optimized for bitsliced software implementations on microcontrollers. Saturnin uses 10 super-rounds (20 rounds) for encryption, while a 16-super-round variant is used for hashing.

1.3 Our Contributions

Our contributions can be summarized as follows.

- **A new hardware-friendly wide-block block cipher.** We specify LEANY (Section 3), a 256-bit block cipher with a 256-bit key: a nibble-oriented SPN with an 8×8 state of 4-bit cells, an optimal 4-bit S-box, a branch-9 MDS diffusion layer over $\mathbb{F}_{2^4}$, an extremely lightweight key schedule, and 12 rounds. Reference test vectors are provided.
- **Provable trail bounds and a broad security analysis.** Using the wide-trail strategy, we show that every four consecutive rounds activate at least $9^2 = 81$ S-boxes, so that the probability (resp. correlation potential) of any differential (resp. linear) trail over eight or more rounds is at most $2^{-324} < 2^{-256}$ (Section 4). We further analyze impossible differential, integral, meet-in-the-middle, boomerang, algebraic and cube attacks, and we prove that invariant subspace and nonlinear invariant attacks are excluded for *every* key by showing that the round constants attain the maximal invariant dimension of 256.
- **A cheap and provably effective round-constant strategy.** We show that rotating a single nothing-up-my-sleeve constant by one *bit* per round which is implementable as a plain shift register, suffices to reach the maximal invariant dimension, a design guideline of independent interest for other ciphers with lightweight key schedules.
- **FPGA implementations and benchmarking.** We report iterative and fully pipelined FPGA implementations (Section 5), requiring about 1.4K LUTs in the iterative case and exceeding 50 Gbps in the pipelined case, and we compare them with the available implementations of 256-bit block ciphers, including WAES-256 and Threefish-256, showing substantially better area efficiency at competitive throughput.

1.4 Paper Organization

The remainder of this paper is organized as follows. Section 2 recalls the necessary background on differential and linear cryptanalysis, MDS matrices, and

the wide-trail strategy. Section 3 specifies LEANY and motivates its components. Section 4 presents the security analysis, concluding with a summary of the attainable rounds for each attack family. Section 5 reports the FPGA implementations and compares them with related work, and Section 6 concludes the paper.

2 Preliminaries

This section fixes the notation and recalls the standard notions used throughout the paper, following the terminology of Daemen and Rijmen [9].

2.1 Notation

$\mathbb{F}_2$ denotes the binary field and $\mathbb{F}_{2^4} = \mathbb{F}_2[x]/(x^4+x+1)$ the field with 16 elements, whose elements (*nibbles*) we write as hexadecimal digits. For $u \in (\mathbb{F}_{2^4})^m$, the *nibble weight* $\mathrm{wt}(u)$ is the number of nonzero coordinates of u; $\oplus$ denotes bitwise XOR and $\lll$ a cyclic left rotation.

2.2 Differential and Linear Trails and Active S-boxes

Differential cryptanalysis [3] studies the propagation of XOR differences. For an n-bit S-box S, the *differential probability* of a differential $(a \to b)$ is $\mathrm{DP}_S(a,b) = 2^{-n}\,\#\{x : S(x) \oplus S(x \oplus a) = b\}$, and $\mathrm{DP}_{\max} = \max_{a\neq 0,b} \mathrm{DP}_S(a,b)$. Linear cryptanalysis [19] studies linear approximations: the *correlation* of a mask pair (a,b) is $c_S(a,b) = 2\Pr_x[a\cdot x \oplus b\cdot S(x) = 0] - 1$, $c_{\max}$ is the maximal absolute correlation over $a, b \neq 0$, and $c_S(a,b)^2$ is the *correlation potential* (or linear probability) of the approximation.

In an SPN, a *differential (resp. linear) trail* over r rounds specifies the intermediate difference (resp. mask) at the boundary of every layer, and an S-box is *active* if its input difference (resp. mask) is nonzero. Under the standard independence assumptions, the probability (resp. correlation potential) of a trail is the product of the differential probabilities (resp. correlation potentials) of its active S-boxes. Hence, if every nontrivial r-round trail has at least N active S-boxes, every r-round differential trail has probability at most $\mathrm{DP}_{\max}^{N}$ and every r-round linear trail has correlation potential at most $c_{\max}^{2N}$; lower-bounding the number of active S-boxes is therefore the designer's central task.

2.3 MDS Matrices

Definition 1 (Branch number). *Let M be an $m \times m$ matrix over $\mathbb{F}_{2^c}$. The* differential branch number *of M is $\mathcal{B}_d(M) = \min_{u\neq 0}\big(\mathrm{wt}(u) + \mathrm{wt}(Mu)\big)$, and the* linear branch number *is $\mathcal{B}_\ell(M) = \mathcal{B}_d(M^{\mathsf{T}})$.*

Since a single nonzero input coordinate activates at most all m output coordinates, $\mathcal{B}_d(M) \leq m+1$, the diffusion-layer analogue of the Singleton bound. M is called *maximum distance separable* (MDS) if the linear code $\{(u, Mu) : u \in \mathbb{F}_{2^c}^m\}$ of length $2m$ and dimension m attains this bound, i.e., has minimum distance $m+1$.

Proposition 1 ([9]). *For an $m \times m$ matrix M over $\mathbb{F}_{2^c}$, the following are equivalent: (i) M is MDS; (ii) every square submatrix of M is nonsingular; (iii) $\mathcal{B}_d(M) = m+1$. Moreover, if M is MDS, then so are M^{T} and M^{-1}; in particular $\mathcal{B}_d(M) = \mathcal{B}_\ell(M) = m + 1$, both maximal.*

MDS matrices are the standard tool for maximal diffusion per layer; e.g., the MixColumns matrix of the AES is a 4×4 MDS matrix over $\mathbb{F}_{2^8}$ with branch number 5 [9]. Enlarging the matrix while shrinking the field—LEANY uses an 8×8 MDS matrix over $\mathbb{F}_{2^4}$ with branch number 9—increases the guaranteed number of active S-boxes per column while keeping every coefficient multiplication a small XOR network, an attractive trade-off in hardware.

2.4 The Wide-Trail Strategy

The wide-trail strategy [9] designs the linear layer so that *every* trail is forced to contain many active S-boxes, allowing trail probabilities to be bounded by counting rather than by search. We consider *AES-like* ciphers: the state is an $m \times m$ matrix of c-bit cells, and each round applies (i) a cell-wise S-box layer, (ii) a cell permutation π, (iii) a column-wise multiplication by a matrix M over $\mathbb{F}_{2^c}$, and a round-key addition.

Definition 2 (Diffusion-optimal permutation). *A cell permutation π is* diffusion optimal *if, for every column, its m cells are mapped by π to m pairwise distinct columns.*

Theorem 1 (Two-round propagation [9]). *In an AES-like cipher whose mixing matrix has branch number $\mathcal{B}$, every nontrivial differential or linear trail over two consecutive rounds has at least $\mathcal{B}$ active S-boxes.*

Theorem 2 (Four-round propagation [9]). *In an AES-like cipher whose mixing matrix has branch number $\mathcal{B}$ and whose cell permutation is diffusion optimal, every nontrivial differential or linear trail over four consecutive rounds has at least $\mathcal{B}^2$ active S-boxes.*

Intuitively, Theorem 1 applied column-wise bounds the number of active cells in rounds 1–2 and 3–4 by $\mathcal{B}$ times the number of active columns at the respective boundaries, and diffusion optimality of π spreads these cells over at least $\mathcal{B}$ distinct columns in the middle of the trail, multiplying the two bounds; see [9] for the full proof. Both theorems hold for linear trails because $\mathcal{B}_\ell(M) = \mathcal{B}_d(M)$ for MDS matrices (Proposition 1). For LEANY, $\mathcal{B} = 9$, so Theorem 2 guarantees at least 81 active S-boxes over every four consecutive rounds (Section 4).

3 Specification of the Cipher

LEANY is a 256-bit block cipher with a 256-bit key and $r = 12$ rounds. The state is an 8×8 matrix of 4-bit nibbles. Each round follows the same four-layer

SPN structure: round key addition, substitution, nibble-wise row rotation, and column-wise matrix diffusion; after the last round, a final whitening key is added. The main parameters of the cipher and their roles are summarized in Table 1, and the pseudocode is given in Algorithm 1. The individual components are specified in the remainder of this section.

Table 1. Overview of the parameters and notation of LEANY.

Parameter	Meaning	Value / definition
n	Block size	256 bits
k	Key size	256 bits
r	Number of rounds	12
c	Cell (nibble) size	4 bits
k_i	i-th bit of the master key K	k_0 the most significant bit
s_i	i-th bit of the state	s_0 the most significant bit
a_i	i-th 4-bit state element	$s_{4i}\|s_{4i+1}\|\cdots\|s_{4i+3}$, $0 \le i \le 63$
$a_{i,j}$	State element in row i, col. j	a_{8i+j}, $0 \le i, j \le 7$
S	Substitution layer	64 parallel 4-bit S-boxes (Table 2)
$\mathrm{DP}_{\max}$	Max. diff. probability of S	2^{-2}
$c_{\max}$	Max. abs. correlation of S	2^{-1}
M	Diffusion matrix	8×8 MDS over $\mathbb{F}_{2^4}$, $f(x) = x^4 + x + 1$
$\mathcal{B}(M)$	Branch number of M	9
RK_i	Round keys	$K \oplus RC_{i-1}$ for $1 \le i \le r+1$
k_j^i	j-th bit of round key RK_i	$RK_i = k_0^i\, k_1^i \cdots k_{255}^i$
RC_i	Round constants	$RC_{\text{seed}} \lll i$ bits; seed from digits of π

Algorithm 1 LEANY

```
Derive round keys RK_1, ..., RK_{r+1} from K
for i = 1 to r do
    addRoundKey(STATE, RK_i)
    sBoxLayer(STATE)
    shiftRow(STATE)
    MixColumn(STATE)
end for
addRoundKey(STATE, RK_{r+1})
```

3.1 The Round Components and Key Schedule

The State. The algorithm operates on a 256-bit state denoted by $s_0 s_1 \cdots s_{255}$, where s_0 represents the most significant bit of the state and s_{255} represents

the least significant bit. The state bits $s_0 s_1 \cdots s_{255}$ are grouped into sixty-four 4-bit state elements $a_0, a_1, \cdots, a_{63}$, such that $a_i = s_{4i} \| s_{4i+1} \| s_{4i+2} \| s_{4i+3}$ for $0 \leq i \leq 63$. The state elements a_i are arranged in an 8×8 matrix in row-wise order as follows:

$$\begin{bmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 \\ a_8 & a_9 & a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{16} & a_{17} & a_{18} & a_{19} & a_{20} & a_{21} & a_{22} & a_{23} \\ a_{24} & a_{25} & a_{26} & a_{27} & a_{28} & a_{29} & a_{30} & a_{31} \\ a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} & a_{39} \\ a_{40} & a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} \\ a_{48} & a_{49} & a_{50} & a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \\ a_{56} & a_{57} & a_{58} & a_{59} & a_{60} & a_{61} & a_{62} & a_{63} \end{bmatrix}$$

Round Key Addition. A round key is added to the state by a simple bitwise XOR operation. Given round key $RK_i = \{k_0^i, \cdots, k_{255}^i\}$ for $1 \leq i \leq r+1$ and current state $\{s_0, \cdots, s_{255}\}$, round key addition consists of the operation for $0 \leq j \leq 255$,

$$s_j \leftarrow s_j \oplus k_j^i.$$

Substitution Layer. A 4-bit to 4-bit S-box S is used. The hexadecimal input–output mapping of the S-box S is given in Table 2.

The S-box S was selected for its cryptographic strength. Its main properties, computed from its difference distribution table and linear approximation table, are as follows:

- *Differential probability:* the differential uniformity of S is 4, i.e., the maximal probability of a nonzero differential is $\mathrm{DP}_{\max} = 4/16 = 2^{-2}$.
- *Linear bias:* the linearity of S is 8, i.e., the maximal absolute bias of a nontrivial linear approximation is $\varepsilon_{\max} = 8/32 = 2^{-2}$; equivalently, the maximal absolute correlation is $c_{\max} = 2\varepsilon_{\max} = 2^{-1}$ and the nonlinearity of S is 4.
- *Algebraic degree:* the algebraic degree of both S and S^{-1} is 3, the maximum possible for a 4-bit permutation; moreover, since the linearity is strictly below 2^4, no nonzero component function of S is affine.
- *Fixed points:* S has no fixed points, i.e., $S[x] \neq x$ for all $x \in \mathbb{F}_2^4$.

Differential uniformity 4, linearity 8, and algebraic degree 3 are the best values achievable by bijective 4-bit S-boxes, so S is *optimal* in the sense of the classification of Leander and Poschmann [16]. These criteria coincide with those used in established lightweight designs such as PRESENT [4].

Three considerations guided the choice of the S-box: First, on the 6-input LUT fabric of modern FPGAs any 4-bit to 4-bit S-box has the same cost (one LUT per output bit, or two dual-output LUTs), so the S-box could be selected purely on cryptographic grounds, without any hardware penalty; we therefore restricted the search to optimal S-boxes in the sense above. Second, it is verified by exhaustive search over the 322 560 that affine bijections of $\mathbb{F}_2^4$ that S is *affine equivalent* to the S-box of PRESENT. Hence S belongs to the same optimal

equivalence class, and it inherits all affine-invariant properties of that thoroughly studied S-box: the differential and linear spectra, the algebraic degree, and the boomerang uniformity discussed in Section 4. Third, among the representatives of this class we chose one that has no fixed points; the invariant-subspace analysis of Section 4.7 is, in any case, independent of the particular S-box within the class.

Table 2. LEANY S-box

Input	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Output	3	0	6	5	A	7	C	B	4	D	9	E	1	8	F	2

In the substitution layer, each 4-bit state element a_i, for $0 \le i \le 63$, is replaced by $S[a_i]$ to produce the updated state.

ShiftRow Layer Unlike a bit-level permutation, the diffusion of LEANY begins with a nibble-level operation. The state matrix is treated as eight rows of eight nibbles each, and row i (for $0 \le i \le 7$) is cyclically rotated to the left by i nibble positions; row 0 is left unchanged. Formally, writing row i of the state as $a_{i,0}\, a_{i,1} \cdots a_{i,7}$, this layer produces

$$\overline{a}_{i,j} = a_{i,\,(j+i) \bmod 8}, \qquad 0 \le i, j \le 7.$$

The inverse of this layer rotates row i to the right by i positions instead.

MixColumn Layer. An 8×8 matrix M defined over the finite field $\mathbb{F}_{2^4}$ with the irreducible polynomial $f(x) = x^4 + x + 1 \in \mathbb{F}_2[x]$ is employed:

$$M = \begin{bmatrix} 1 & 9 & d & 6 & 2 & c & a & 8 \\ 9 & 1 & 6 & d & c & 2 & 8 & a \\ d & 6 & 1 & 9 & a & 8 & 2 & c \\ 6 & d & 9 & 1 & 8 & a & c & 2 \\ 2 & c & a & 8 & 1 & 9 & d & 6 \\ c & 2 & 8 & a & 9 & 1 & 6 & d \\ a & 8 & 2 & c & d & 6 & 1 & 9 \\ 8 & a & c & 2 & 6 & d & 9 & 1 \end{bmatrix}.$$

The matrix M is maximum distance separable (MDS) over $\mathbb{F}_{2^4}$, i.e., every square submatrix of M is invertible. Consequently, M achieves the maximum possible branch number for an 8×8 matrix, $\mathcal{B}(M) = 9$: for any nonzero input column vector u, the total number of nonzero nibbles in u and Mu combined is at least 9. This guarantees that a single active nibble entering the layer activates all eight output nibbles of its column, providing full diffusion within each column after a single application of the layer.

The matrix M is a *Hadamard* matrix over $\mathbb{F}_{2^4}$: writing $h = (h_0, \ldots, h_7) = (\mathtt{1}, \mathtt{9}, \mathtt{d}, \mathtt{6}, \mathtt{2}, \mathtt{c}, \mathtt{a}, \mathtt{8})$ for its first row, its entries satisfy $M_{i,j} = h_{i \oplus j}$. We selected M among the MDS Hadamard matrices of this form for the following reasons.

- *Cheap and structured evaluation.* Every coefficient is multiplied by a fixed constant of $\mathbb{F}_{2^4}$, which is a tiny XOR network: multiplication by $\mathtt{1}, \mathtt{9}, \mathtt{2}, \mathtt{d}, \mathtt{8}, \mathtt{6}, \mathtt{c}, \mathtt{a}$ costs $0, 1, 1, 3, 3, 5, 5, 8$ XOR gates, respectively. A direct evaluation of M on one column costs 432 XOR gates before any sharing of common subexpressions, and the Hadamard structure exposes many such subexpressions (the same eight constants appear in every row, in a butterfly-like pattern), which the synthesizer exploits.
- *Symmetric and nearly involutory.* A Hadamard matrix is symmetric, $M^{\mathsf{T}} = M$, so the linear branch number equals the differential branch number without any further argument, and M satisfies $M^2 = \mathtt{a} \cdot I$, whence $M^{-1} = \mathtt{c} \cdot M$. Decryption can therefore reuse the encryption matrix circuit, followed by one multiplication by the constant $\mathtt{c}$ per nibble (5 XORs), instead of a second, unrelated 8×8 matrix; a decryption circuit thus costs almost the same as the encryption circuit.

Each column of the state matrix is multiplied by M, where each column is interpreted as an eight-tuple of elements in $\mathbb{F}_{2^4}$. For each column index k, $0 \le k \le 7$, the column vector $(a_{0,k}, a_{1,k}, \ldots, a_{7,k})^{\mathsf{T}}$ is multiplied by the matrix M to produce the updated state:

$$M \cdot \begin{bmatrix} a_{0,k} \\ a_{1,k} \\ \vdots \\ a_{7,k} \end{bmatrix} = \begin{bmatrix} \overline{a}_{0,k} \\ \overline{a}_{1,k} \\ \vdots \\ \overline{a}_{7,k} \end{bmatrix}.$$

Key Schedule. The key schedule of LEANY uses a single 256-bit master key K and a single 256-bit round-constant seed RC_{seed}, taken from the fractional hexadecimal digits of π as a nothing-up-my-sleeve constant:

$RC_{\text{seed}} = \mathtt{243F6A8885A308D313198A2E03707344A4093822299F31D0082EFA98EC4E6C89}.$

For each round $i = 0, 1, \ldots, r$, the round constant RC_i is obtained by cyclically rotating RC_{seed} to the left by i bits:

$$RC_0 = RC_{\text{seed}}, \qquad RC_i = RC_{i-1} \lll 1 \text{ bit}, \quad 1 \le i \le r.$$

The round key for round i is then

$$RK_i = K \oplus RC_{i-1}, \qquad 1 \le i \le r,$$

and the final whitening key is

$$RK_{r+1} = K \oplus RC_r.$$

Rotating RC_{seed} by a single bit rather than by a full nibble is a deliberate design choice motivated by resistance against invariant subspace attacks, as discussed in Section 4.7. Since the round-constant register can be implemented in hardware as a simple 256-bit cyclic shift register updated by one bit per round, the key schedule remains lightweight, requiring neither an LFSR nor a round-constant lookup table.

Table 3. Reference test vectors for LEANY-256/256.

Vector	Field	Value
1	Plaintext	FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
	Key	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
	Ciphertext	F979CE46 2472CF4A D2811E69 CDD5F2AF B0158C07 57B457FB 1B0084FA 902BB7B3
2	Plaintext	01234567 89ABCDEF 01234567 89ABCDEF 01234567 89ABCDEF 01234567 89ABCDEF
	Key	00112233 44556677 8899AABB CCDDEEFF 01234567 89ABCDEF FEDCBA98 76543210
	Ciphertext	71CC3005 6969F69E 9C576AC7 1DAA2D23 9EE2DB31 972EEA61 BB9231D3 F3F95F79
3	Plaintext	13579BDF 02468ACE FEDCBA98 76543210 2468ACE0 13579BDF 88888888 31415926
	Key	FEDCBA98 76543210 13579BDF 02468ACE 22223333 44445555 66667777 88889999
	Ciphertext	C7FA2BB3 1C28BB14 147E478F CDD54F4A 0E78D084 A89C9BB2 3F8676DC D63F24BA

3.2 Test Vectors

A set of reference test vectors for LEANY is presented in Table 3. These vectors can be used to verify the correctness of software and hardware implementations.

4 Security Analysis of LEANY

This section analyzes the resistance of LEANY against the main families of cryptanalytic attacks, using the notation and results of Section 2. We emphasize that the security of LEANY aims resistance against single-key attacks, hence we do not include any related-key attacks. Recall from Section 3 that the S-box satisfies $\mathrm{DP}_{\max} = 2^{-2}$ and $c_{\max} = 2^{-1}$, and that M is MDS with $\mathcal{B}(M) = 9$. The findings of this section are summarized in Table 4 at its end.

4.1 Resistance to Differential Cryptanalysis

Differential cryptanalysis [3] exploits differentials of high probability over a large number of rounds. For LEANY, the ShiftRow is diffusion optimal in the sense of Definition 2. Together with $\mathcal{B}(M) = 9$, Theorems 1 and 2 yield the following.

Corollary 1. *Every nontrivial differential or linear trail over four consecutive rounds of* LEANY *has at least* $\mathcal{B}(M)^2 = 81$ *active S-boxes. More generally, a nontrivial trail over* $4a$ *rounds has at least* $81a$ *active S-boxes.*

Since $\mathrm{DP}_{\max} = 2^{-2}$, every differential trail over eight rounds has probability at most $\left(2^{-2}\right)^{162} = 2^{-324} < 2^{-256}$, so differential distinguishers based on single trails are impossible from eight rounds on. We do not expect any 7-round distinguishers since $(2^{-2})^{81 \cdot 7/4} = 2^{-283,5} < 2^{-256}$ on average. Our most generous expectation is for a 6-round distinguisher and a 7-round attack.

4.2 Resistance to Linear Cryptanalysis

Linear cryptanalysis [19] exploits linear approximations with high absolute correlation. Since M is MDS, M^{T} is MDS as well (Proposition 1), so the linear branch number of the diffusion layer also equals 9, and Corollary 1 applies verbatim to linear trails. As $c_{\max} = 2^{-1}$, the correlation potential contributed by each active S-box is at most 2^{-2}; hence every linear trail over eight rounds has correlation potential at most $2^{-324} < 2^{-256}$, and the data complexity of a corresponding attack—inversely proportional to the correlation potential—would exceed the full codebook. Similar to the differential cryptanalysis, we do not expect any 7-round distinguishers considering the minimum number of activated S-boxes per round on average. Therefore, our most generous expectation is for a 6-round distinguisher and a 7-round attack.

4.3 Resistance to Impossible Differential Cryptanalysis

Impossible differential cryptanalysis [2] uses differentials of probability zero, typically constructed by the miss-in-the-middle technique from two probability-one truncated differentials that contradict each other. The reach of this technique is governed by the diffusion speed of the cipher. LEANY achieves full diffusion after two rounds in both directions: a single active nibble activates its entire column through MixColumn, ShiftRow then distributes these eight nibbles into eight pairwise distinct columns, and the following MixColumn activates all 64 nibbles; the decryption direction behaves identically because M^{-1} is also MDS (Proposition 1) and the inverse ShiftRow offsets are likewise pairwise distinct. Probability-one truncated differentials therefore extend over at most two rounds from each side, and miss-in-the-middle contradictions yield impossible differentials that are structurally analogous to the classical four-round impossible differentials of AES-like ciphers. Key-recovery extensions around such a four-round distinguisher must filter pairs through the branch-9 diffusion layer on a 256-bit state, which quickly inflates the data and time complexity. We conservatively estimate that impossible differential attacks cannot reach beyond seven rounds.

4.4 Resistance to Integral and MitM Cryptanalysis

Integral attacks. The nibble-oriented structure of LEANY admits the classical Square property [8, 15]. Consider a Λ-set of 2^4 plaintexts that are identical

except in one nibble, which ranges over all 16 values. After one round, the set has one *active* (all-values) column; after two rounds, all 64 nibbles are active; and after three rounds, every nibble is *balanced*, i.e., its XOR over the set equals zero. This gives a three-round distinguisher with 2^4 chosen plaintexts, which extends to four rounds by prepending a structure with one fully active column (2^{32} chosen plaintexts); division-property-based refinements [24] may gain in the order of one additional round. Appending key-recovery rounds requires partially decrypting through the MixColumn layer, guessing 32-bit key chunks per active column, so the complexity grows rapidly with each appended round. We do not expect integral attacks to reach beyond six rounds.

Meet-in-the-middle attacks. Classical MitM attacks require decomposing the cipher into consecutive subciphers that depend on (partially) independent key bits. In LEANY, every round key equals the full 256-bit master key XORed with a public constant, so every single round depends on all key bits and no such decomposition exists. Demirci–Selçuk-type MitM distinguishers [10] instead tabulate the parameters of truncated trails through the middle rounds; for LEANY, the 64-cell state and the guarantee of at least 81 active S-boxes over any four rounds (Corollary 1) inflate the number of such parameters far beyond exhaustive key search already for short distinguishers. We therefore do not expect MitM techniques to reach beyond seven rounds.

4.5 Resistance to Boomerang Attacks

The boomerang attack of Wagner [26] combines differential characteristics over two parts of a cipher. In the sandwich formulation $E = E_1 \circ E_m \circ E_0$, the upper and lower characteristics hold with probabilities p and q, and the interaction in the middle part succeeds with probability r, giving the quartet probability $P_{\text{boom}} = p^2 q^2 r$; for S-box based ciphers, the Boomerang Connectivity Table (BCT) of Cid et al. [6] evaluates the middle transition.

Exhaustive evaluation of the 4-bit S-box of LEANY gives boomerang uniformity 16, with exactly two nontrivial deterministic entries, $\mathrm{BCT}(1,5) = 16$ and $\mathrm{BCT}(\mathtt{F},\mathtt{F}) = 16$, i.e., switching probability one. Placing one of these transitions in the middle S-box layer and propagating the remaining differences exactly through ShiftRows and the branch-9 MDS layer, we obtain a five-round characteristic with $(\Delta_i, \nabla_o) = (1,5)$ in the switching layer. In the upper part E_0, eight active S-boxes produce a single active nibble with value $\Delta_i = 1$ at the input of E_m; seven transitions have probability 2^{-2} and one has probability 2^{-3}, giving $p = 2^{-17}$. The middle transition satisfies $\mathrm{BCT}(1,5) = 16$, while all remaining positions are probability-one ladder switches, hence $r = 1$. The lower part E_1 follows an $8 \to 1 \to 8$ active-S-box pattern with all 17 active S-boxes attaining probability 2^{-2}, so $q = 2^{-34}$. Consequently, the complete five-round boomerang characteristic has probability $P_{\text{boom}} = (2^{-17})^2 (2^{-34})^2 = 2^{-102}$.

Direct one-round extensions of this five-round distinguisher fail: prepending one round forces 64 additional active S-boxes and, after optimizing the compatible difference values, reduces the quartet probability to 2^{-364}; appending one

round produces the four-round lower pattern $8 \to 1 \to 8 \to 64$ with 81 active S-boxes and yields probability 2^{-370}. Both values are below the 2^{-256} random-quartet probability of a 256-bit permutation. Thus, although the deterministic BCT entries of the LEANY S-box are exploitable, the useful boomerang characteristic is confined to five rounds: it may serve as a core for an outer-round key-recovery extension, but the distinguisher does not extend to six rounds and does not threaten the full 12-round cipher.

4.6 Resistance to Algebraic Attacks and its Variants

Algebraic attacks model the cipher as a system of low-degree polynomial equations and attempt to solve it, e.g., by linearization or Gröbner-basis techniques [7]. Every 4-bit S-box satisfies at least 21 linearly independent quadratic equations over $\mathbb{F}_2$ in its eight input and output bits [4]. An encryption with LEANY involves $12 \cdot 64 = 768$ S-box evaluations, giving at least $768 \cdot 21 = 16{,}128$ quadratic equations in $768 \cdot 8 = 6{,}144$ intermediate state variables plus 256 key variables. Systems orders of magnitude smaller remain unsolved by known techniques, and the dense MDS layer destroys the sparsity that successful algebraic attacks exploit, so algebraic attacks pose no threat to full-round LEANY.

Resistance to Cube Attacks. Cube attacks [11], like higher-order differential attacks, exploit a low algebraic degree in the public variables. The S-box of LEANY has algebraic degree 3, so the degree after t rounds is bounded by $\min(3^t, 255)$; it grows as $3, 9, 27, 81, 243, \ldots$ and essentially reaches the maximal degree of a 256-bit permutation after five rounds. A direct cube attack with linear superpolies requires a cube of dimension at least $3^t - 1$: this is still theoretically possible for five rounds ($3^5 - 1 = 242$), but impossible for six rounds since $3^6 - 1 = 728$ exceeds the whole plaintext space.

The internal structure of LEANY nevertheless allows cube testers to go further. If all four bits of a nibble are activated, round-key addition composed with the 4-bit S-box is a key-dependent permutation of the nibble space and does not reduce the cube. Choosing active nibbles on shifted diagonals aligns them into full columns after ShiftRows, and the column-wise MDS layer maps full active columns to full active columns, so the first two S-box layers are "free" for such a cube. Three shifted diagonals give a 96-dimensional cube, while after the two free S-box layers the remaining degree over six rounds is bounded by $3^4 = 81$. Since $96 > 81$, the cube sum of any six-round output coordinate is zero, yielding a cube-tester distinguisher with data and time complexity about 2^{96}.

This six-round core extends by one round using a partial last-round key guess: since the final inverse linear layer is column-wise, guessing the 32 key bits of one output column peels one round for that column and reduces a seven-round computation to the six-round core. The correct guess recovers the zero-sum property, whereas a wrong guess passes each tested output bit with probability about $1/2$, so a few independent output-bit or cube tests eliminate false positives. This gives a theoretical seven-round key recovery attack with data complexity

about 2^{96} and time complexity about 2^{96+32} per column, or about 2^{131} for all columns.

The approach does not extend to eight rounds. After peeling one round, a seven-round core remains whose degree bound, even with the two initial S-box layers bypassed, is $3^5 = 243$; a zero-sum cube tester would need a cube of dimension larger than 243, i.e., all eight shifted diagonals, which degenerates into a full-codebook property. Peeling two rounds instead destroys the column-local structure: after one inverse round, InvShiftRows and the MDS layer spread the dependencies, so the second inverse round would require guessing essentially the full key. Hence the attack stops at seven rounds, and the best cube and cube-tester techniques identified here leave LEANY with a substantial security margin against cube and higher-order differential cryptanalysis.

4.7 Resistance to Invariant Subspace Attacks

When a key schedule is lightweight e.g., when round keys are derived by simply XORing the master key with public round constants, the cipher becomes susceptible to invariant subspace [17] and nonlinear invariant [25] attacks, which can produce a large class of weak keys or, in the worst case, a distinguisher valid for every key. Let L denote the linear layer of one round of LEANY (MixColumn $\circ$ ShiftRow, viewed as an $\mathbb{F}_2$-linear map on $\mathbb{F}_2^{256}$), and let D denote the $\mathbb{F}_2$-span of the round constants used by the key schedule. Following the criterion of [1], we compute $W_L(D)$, the smallest L-invariant subspace of $\mathbb{F}_2^{256}$ containing D. If $\dim W_L(D) = 256$, no invariant subspace, linear or, via the associated nonlinear invariant framework, nonlinear can be simultaneously compatible with the S-box layer and the linear-plus-constant layer, for any master key. This rules out the entire invariant-attack family independently of the S-box's specific behavior.

For LEANY, the round constants are derived from a single 256-bit *nothing-up-my-sleeve* seed RC_0, taken from the hexadecimal digits of π, with RC_r obtained by cyclically rotating RC_0 left by r *bits*. The rotation is deliberately bit-level rather than nibble-level: because L is $\mathrm{GF}(2^4)$-linear, any nibble-aligned construction confines the constants to a proper submodule of $\mathbb{F}_2^{256}$, capping $\dim W_L(D)$ at 252 regardless of the shift amount or direction, as we verified by exhaustive search. A rotation by an amount not a multiple of 4 bits breaks this nibble alignment. Using the algorithm of [1], we computed $\dim W_L(D)$ for LEANY's linear layer combined with this round-constant sequence and obtained $\dim W_L(D) = 256$, the theoretical maximum, proving that LEANY admits no invariant-subspace or nonlinear-invariant attack of this class for any key.

4.8 Summary of the Security Analysis

Table 4 summarizes the results of this section. For each attack family, we list the number of rounds covered by the longest distinguisher supported by our analysis, together with a conservative estimate of the maximal number of rounds that could be attacked when generous key-recovery extensions are taken into account. No attack family is expected to reach beyond seven of the twelve rounds. So

LEANY retains a security margin of at least five rounds even under these highly conservative assumptions.

Table 4. Longest distinguishers and conservative estimates of the maximal number of attackable rounds of LEANY (12 rounds in total) per attack family.

Attack family	**Distinguisher (rounds)**	**Attacked rounds (est.)**
Differential	≤ 7	7
Linear	≤ 7	7
Impossible differential	4	7
Integral / division property	4–5	6
Meet-in-the-middle	≤ 5	7
Algebraic / cube	6	7
Boomerang	5	6
Invariant (subspace/nonlinear)	excluded for all keys (Sect. 4.7)	

5 Hardware Implementation and Benchmarking

The proposed LEANY block cipher was synthesized using Xilinx Vivado 2022.2 targeting Kintex-7 and Artix-7 FPGA devices. Two hardware organizations were implemented: an iterative architecture consisting of a single round circuit and a fully pipelined architecture with one pipeline stage per encryption round. The reported resource utilization corresponds only to the encryption core; the external serializer interface is excluded from the area measurements.

The hardware implementations are evaluated using the commonly reported FPGA metrics of area, latency, maximum operating frequency, throughput, and throughput per LUT (TP/LUT). Area is measured in terms of Lookup Tables (LUTs) and Flip-Flops (FFs). Since LUT counts are consistently reported across different FPGA families and are less dependent on device-specific slice organization, TP/LUT is adopted as the primary area-efficiency metric.

For the iterative implementation, one encryption requires 14 clock cycles, consisting of one cycle to load the plaintext, twelve cycles for the encryption rounds, and one cycle for the final whitening and ciphertext output. Therefore, the throughput is computed as

$$\text{Throughput} = \frac{\text{Block Size} \times f_{\max}}{\text{Clock Cycles}},$$

where $f_{\max}$ denotes the maximum operating frequency after place-and-route.

The fully pipelined implementation employs a 12-stage pipeline, where each pipeline stage implements one encryption round. In addition to the pipeline stages, one clock cycle is required to load the plaintext into the pipeline and one clock cycle is required for the final whitening and ciphertext output. Consequently, the end-to-end latency of the pipeline is 14 clock cycles, even though the

pipeline itself consists of only 12 stages. However, once the pipeline is completely filled, one 256-bit ciphertext block is produced every clock cycle. Consequently, the steady-state throughput is given by

$$\text{Throughput}_{\text{steady}} = \text{Block Size} \times f_{\max}.$$

As an area-efficiency metric, we report throughput-per-area (TAR), defined as the throughput normalized by LUT count,

$$\text{TAR} = \frac{\text{Throughput}}{\text{LUT}},$$

reported in Mbps/LUT.

It should be noted that the reported latency corresponds to the time elapsed between the arrival of the first plaintext block and the appearance of the corresponding ciphertext at the output. In the pipelined architecture, although a new ciphertext is generated every clock cycle after the pipeline is filled, the latency of an individual block remains 14 clock cycles.

Table 5 and Table 6 compare the proposed implementation with previously published FPGA implementations of 256-bit block ciphers, for the iterative and pipelined architectures, respectively. To the best of our knowledge, only a limited number of 256-bit block ciphers have publicly available FPGA implementation results, making direct comparisons relatively scarce.

The iterative implementation of LEANY occupies only 1358 LUTs on the Kintex-7 device while achieving an encryption latency of 66.67 ns and a throughput of 3.84 Gbps. The corresponding Artix-7 implementation requires only 1372 LUTs, demonstrating that LEANY can be implemented with a compact hardware footprint while maintaining low encryption latency.

The fully pipelined implementation increases the hardware cost to 14,676 LUTs on Kintex-7 and 15,143 LUTs on Artix-7, but improves the steady-state throughput to 53.76 Gbps and 51.20 Gbps, respectively. This illustrates the expected trade-off between hardware area and throughput, where additional hardware resources enable significantly higher throughput without increasing the latency experienced by an individual data block.

Compared with the recently proposed WAES-256, which is the 256-bit block variant of Rijndael, the proposed LEANY architecture requires substantially fewer hardware resources. The WAES-256 implementation occupies more than 45,000 LUTs even on modern FPGA platforms, whereas the proposed iterative LEANY implementation requires approximately 1,400 LUTs. However, WAES-256 targets a different design objective by maximizing throughput through aggressive pipelining and has also been evaluated on newer FPGA families such as Kintex UltraScale+, achieving operating frequencies exceeding 800 MHz. Consequently, direct throughput comparisons should be interpreted with care since both FPGA technology and architectural optimization goals differ substantially.

Overall, the available FPGA implementations demonstrate that relatively few 256-bit block ciphers have been thoroughly evaluated in hardware. Existing implementations generally emphasize either compact hardware realization (e.g.,

Table 5. FPGA implementation results of LEANY (Iterative architecture) and comparison with related 256-bit block ciphers. Here, "R." refers to the number of rounds and "Lat." refers to the latency.

Cipher	Device	R.	LUTs	FFs	Lat. (ns)	Freq. (MHz)	TP (Gbps)	TAR (Mbps/LUT)
LEANY [ours]	Kintex-7	12	1358	780	66.67	210	3.840	2.83
	Artix-7	12	1372	780	66.67	210	3.840	2.80
Threefish-256 [20]	Spartan-6	72	5325	4000	—	164	0.584	0.11

Table 6. FPGA implementation results of LEANY (Pipelined architecture) and comparison with related 256-bit block ciphers. Here, "R." refers to the number of rounds and "Lat." refers to the latency.

Cipher	Device	R.	LUTs	FFs	Lat. (ns)	Freq. (MHz)	TP (Gbps)	TAR (Mbps/LUT)
LEANY [ours]	Kintex-7	12	14676	3341	66.67	210	53.760	3.66
	Artix-7	12	15143	3341	70.00	200	51.200	3.38
WAES-256 [18]	Artix-7	14	46617	52286	446.0	282.48	72.316	1.55
	Kintex US+	14	45591	52371	156.5	805.1	206.11	4.52

Threefish-256) or maximum throughput through aggressive pipelining on modern FPGA devices (e.g., WAES-256). In contrast, LEANY offers a balanced design that achieves a compact hardware footprint, low encryption latency, and competitive throughput. Furthermore, the fully pipelined implementation demonstrates that the proposed round function is well suited for high-throughput hardware architectures, achieving more than 53 Gbps on a Kintex-7 FPGA and 51 Gbps on an Artix-7 FPGA while maintaining a pipeline latency of only 14 clock cycles.

6 Conclusion

We introduced LEANY, a 256-bit block cipher with a 256-bit key that transfers the wide-trail design philosophy of the AES to a nibble-oriented state. The combination of an optimal 4-bit S-box, a diffusion-optimal ShiftRow, and an 8×8 MDS matrix over $\mathbb{F}_{2^4}$ with branch number 9 guarantees at least 81 active S-boxes over every four rounds, so that the probability and correlation potential of every eight-round differential and linear trail fall below $2^{-324} < 2^{-256}$, a bound obtained by hand without any automated search. Our analysis of impossible differential, integral, meet-in-the-middle, boomerang, algebraic, and cube attacks indicates that no attack family reaches beyond seven of the twelve rounds, and we proved that the bit-rotated round constants attain the maximal invariant dimension, excluding invariant subspace and nonlinear invariant attacks for every

key. On FPGAs, LEANY occupies only about 1.4K LUTs in an iterative architecture and exceeds 50 Gbps in a fully pipelined one, offering, to the best of our knowledge, the best area efficiency reported for a 256-bit block cipher to date.

We invite third-party cryptanalysis of LEANY, including automated trail and distinguisher searches that may sharpen the conservative estimates of Section 4. Future work includes ASIC and optimized software implementations, and the integration of LEANY into accordion-style modes of operation within NIST's ongoing standardization effort.

References

[1] Christof Beierle et al. "Proving resistance against invariant attacks: How to choose the round constants". In: *Annual International Cryptology Conference*. Springer. 2017, pp. 647–678.

[2] Eli Biham, Alex Biryukov, and Adi Shamir. "Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials". In: *Advances in Cryptology – EUROCRYPT '99*. Springer. 1999, pp. 12–23.

[3] Eli Biham and Adi Shamir. "Differential cryptanalysis of DES-like cryptosystems". In: *Journal of Cryptology* 4.1 (1991), pp. 3–72.

[4] Andrey Bogdanov et al. "PRESENT: An ultra-lightweight block cipher". In: *Cryptographic Hardware and Embedded Systems – CHES 2007*. Springer. 2007, pp. 450–466.

[5] Anne Canteaut et al. "Saturnin: a suite of lightweight symmetric algorithms for post-quantum security". In: *IACR Transactions on Symmetric Cryptology* (June 2020), pp. 160–207. ISSN: 2519-173X. DOI: 10.46586/tosc.v2020.iS1.160-207. URL: https://tosc.iacr.org/index.php/ToSC/article/view/8621.

[6] Carlos Cid et al. "Boomerang connectivity table: a new cryptanalysis tool". In: *Annual international conference on the theory and applications of cryptographic techniques*. Springer. 2018, pp. 683–714.

[7] Nicolas T. Courtois and Josef Pieprzyk. "Cryptanalysis of block ciphers with overdefined systems of equations". In: *Advances in Cryptology – ASIACRYPT 2002*. Springer. 2002, pp. 267–287.

[8] Joan Daemen, Lars Knudsen, and Vincent Rijmen. "The block cipher Square". In: *Fast Software Encryption – FSE '97*. Springer. 1997, pp. 149–165.

[9] Joan Daemen and Vincent Rijmen. *The design of Rijndael*. Vol. 2. Springer, 2002.

[10] Hüseyin Demirci and Ali Aydin Selçuk. "A meet-in-the-middle attack on 8-round AES". In: *Fast Software Encryption – FSE 2008*. Springer. 2008, pp. 116–126.

[11] Itai Dinur and Adi Shamir. "Cube attacks on tweakable black box polynomials". In: *Advances in Cryptology – EUROCRYPT 2009*. Springer. 2009, pp. 278–299.

[12] Niels Ferguson et al. "The skein hash function family". In: *Submission to NIST (round 3)* 7.7.5 (2010), p. 3.

[13] Lov K. Grover. "A fast quantum mechanical algorithm for database search". In: *Proceedings of the 28th Annual ACM Symposium on Theory of Computing – STOC '96*. 1996, pp. 212–219.

[14] Jian Guo, Thomas Peyrin, and Axel Poschmann. "The PHOTON family of lightweight hash functions". In: *Advances in Cryptology – CRYPTO 2011*. Springer. 2011, pp. 222–239.

[15] Lars Knudsen and David Wagner. "Integral cryptanalysis". In: *Fast Software Encryption – FSE 2002*. Springer. 2002, pp. 112–127.

[16] Gregor Leander and Axel Poschmann. "On the classification of 4 bit S-boxes". In: *International Workshop on the Arithmetic of Finite Fields – WAIFI 2007*. Springer. 2007, pp. 159–176.

[17] Gregor Leander et al. "A cryptanalysis of PRINTcipher: The invariant subspace attack". In: *Advances in Cryptology – CRYPTO 2011*. Springer. 2011, pp. 206–221.

[18] Ahmet Malal and Cihangir Tezcan. "First Fully Pipelined High Throughput FPGA Implementation and GPU Optimization of Wider Variant of AES". In: *Journal of Cryptographic Engineering* 16.1 (2026), p. 1.

[19] Mitsuru Matsui. "Linear cryptanalysis method for DES cipher". In: *Advances in Cryptology – EUROCRYPT '93*. Springer. 1993, pp. 386–397.

[20] K Mhaidat, M Altahat, and O Al-Khaleel. "High-Throughput Hardware Implementation of Threefish Block Cipher on FPGA". In: *Proceedings of the 4th International Conference on Information & Communication Systems (iCICS), Iribd, Jordan*. 2013, pp. 23–25.

[21] National Institute of Standards and Technology. *Proposal of Requirements for an Accordion Mode: Discussion Draft for the NIST Accordion Mode Workshop 2024*. NIST Cryptographic Standards Discussion Draft. 2024.

[22] Roman Oliynykov et al. *A New Encryption Standard of Ukraine: The Kalyna Block Cipher*. Cryptology ePrint Archive, Paper 2015/650. 2015. URL: `https://eprint.iacr.org/2015/650`.

[23] Ronald L Rivest et al. "The RC6TM block cipher". In: *First advanced encryption standard (AES) conference*. 1998, p. 16.

[24] Yosuke Todo. "Structural evaluation by generalized integral property". In: *Advances in Cryptology – EUROCRYPT 2015*. Springer. 2015, pp. 287–314.

[25] Yosuke Todo, Gregor Leander, and Yu Sasaki. "Nonlinear invariant attack: Practical attack on full SCREAM, iSCREAM, and Midori64". In: *Advances in Cryptology – ASIACRYPT 2016*. Springer. 2016, pp. 3–33.

[26] David Wagner. "The boomerang attack". In: *International Workshop on Fast Software Encryption*. Springer. 1999, pp. 156–170.