

LEA-RECT: A Parallel Hybrid Lightweight Block Cipher Combining ARX Diffusion and Bit-Slice S-Box Nonlinearity for Big-Data-At-Rest Encryption

Shakir Fadhil Shabram

Ministry of Education, General Directorate of Education of Al-Rusafa Third District, Baghdad, Iraq. shakirnote5@gmail.com

ARTICLE INFO

Article history:

Received: 07 /09/2026

Revised form: 22 /09/2026

Accepted : 23 /09/2026

Available online: 30 /09/2026

Keywords:

lightweight cryptography, hybrid block cipher, LEA, RECTANGLE, ARX, bit-slice S-box, big data, HDFS,

ABSTRACT

The rapid growth of big-data platforms such as Hadoop/HDFS has intensified the need to protect very large volumes of data at rest without imposing the heavy cost of conventional block ciphers. Purely ARX-based lightweight ciphers (e.g., LEA) achieve fast word-level diffusion but contain no substitution box, and therefore rely solely on modular addition for nonlinearity; purely substitution-permutation ciphers (e.g., RECTANGLE) provide strong algebraic nonlinearity through a bit-slice S-box but are comparatively slower for wide-word software encryption. This paper proposes LEA-RECT, a hybrid lightweight block cipher in which the most important component of a donor cipher (the RECTANGLE bit-slice 4×4 S-box) is embedded, in parallel, alongside the most important component of a host cipher (the LEA ARX round function), and the two branches are then mixed within every round. The construction operates on 128-bit blocks with a 128-bit key over 24 rounds, and is designed so that each round remains a bijection despite the parallel S-box injection. A complete reference implementation was built and validated: encryption/decryption round-trip correctness was confirmed over 30,000 random trials; the cipher attains a mean strict-avalanche of 50.05% for plaintext-bit changes and 50.02% for key-bit changes; ciphertext byte entropy of 7.9998 bits/byte; and it passes the NIST monobit ($p=0.96$), runs ($p=0.96$), and block-frequency ($p=0.20$) randomness tests. The parallel donor branch adds algebraic nonlinearity absent from the ARX host at a measured serial overhead of ~128% on our single-threaded reference implementation; based on the branches' data-independence, the parallel overhead reduced by simulation to ~32% under 4-way SIMD comparing with serial algorithm, and pending hardware validation.

<https://doi.org/10.29304/jqcm.2026.18.33189>

*Shakir Fadhil Shabram

Shakirnote5@gmail.com

1. Introduction

Modern big-data ecosystems store and process petabyte-scale datasets across distributed clusters such as the Hadoop Distributed File System (HDFS). A large fraction of this data is sensitive — personal, financial, medical, or governmental — and must be protected while at rest in the cluster. Applying a conventional cipher such as AES to every block is secure but comparatively expensive at big-data scale, particularly on the commodity and resource-constrained nodes that populate large clusters. This has motivated a decade of research into lightweight cryptography, in which ciphers are designed to minimize computational, memory, and energy cost while retaining strong security guarantees.

Lightweight block ciphers fall broadly into two design families. The first is the Addition–Rotation–XOR (ARX) family, exemplified by LEA, SPECK, and HIGHT, whose round functions use only modular addition, bitwise rotation, and XOR. ARX ciphers are extremely fast in software and provide rapid word-level diffusion, but they contain no substitution box; their entire nonlinearity comes from modular addition. The second is the substitution–permutation network (SPN) family, exemplified by PRESENT, GIFT, and RECTANGLE, which achieve strong algebraic nonlinearity through small S-boxes followed by a bit permutation. RECTANGLE, in particular, is notable for a bit-slice 4×4 S-box that can be evaluated on many nibbles simultaneously using only a short sequence of Boolean gates, making it unusually friendly to word-parallel software and SIMD hardware.

The central observation of this work is that these two families are complementary. An ARX host contributes fast diffusion but lacks an S-box; an SPN donor contributes S-box nonlinearity that is naturally parallel. We therefore propose a hybrid in which the single most important component of the donor cipher — the RECTANGLE bit-slice S-box — is placed in parallel with the single most important component of the host cipher — the LEA ARX round function — and the two are mixed inside every round. The result, LEA-RECT, gains algebraic nonlinearity that the pure ARX host does not possess, while preserving the host's fast diffusion and the donor's parallel structure, which together suit big-data-at-rest encryption.

Contributions. (1) We define LEA-RECT, a 128-bit hybrid lightweight block cipher with a parallel host/donor round structure, and prove its per-round invertibility by construction. (2) We provide a complete, verified reference implementation, including a bit-slice RECTANGLE S-box whose Boolean (algebraic normal form) equations we derive and exhaustively validate. (3) We evaluate the cipher with a full battery of statistical and cryptographic tests — round-trip correctness, strict avalanche criterion for plaintext and key, ciphertext entropy, key sensitivity, plaintext/ciphertext correlation, and three NIST SP 800-22 randomness tests — and report an ablation isolating the donor branch's contribution and cost. (4) We analyze throughput and parallel scalability for the big-data setting.

2. Related Work

Lightweight cryptography has produced a large number of block ciphers targeting constrained environments. Among ARX designs, LEA (Hong et al.) is an ISO/IEC 29192-2 standardized cipher operating on 128-bit blocks with 128/192/256-bit keys and 24/28/32 rounds, designed for fast software encryption on general-purpose and embedded processors. SPECK and SIMON, introduced by the U.S. National Security Agency, are widely benchmarked ARX and AND-rotation ciphers; published software figures place SPECK among the fastest lightweight ciphers, at roughly 2–4 cycles per byte on modern processors. HIGHT and CHAM are further ARX designs frequently used in parallel and IoT implementations.

Among SPN designs, PRESENT established the ultra-lightweight template, and RECTANGLE (Zhang et al.) introduced a bit-slice 4×4 S-box together with a bit-permutation layer that together deliver competitive software speed and very small hardware area. The bit-slice property of the RECTANGLE S-box — that it can be computed by a short fixed sequence of Boolean operations across many parallel nibble lanes — is precisely what makes it attractive as a donor component in a hybrid.

Hybridization of ciphers is an established and recently active idea. Mohammed et al. (2024) proposed PRC6, a two-layer cipher that injects the ChaCha quarter-round into an RC6 extension for cloud data security, demonstrating that transplanting the most important component of one primitive into another is a legitimate and publishable design pattern. Related work has parallelized ARX ciphers (LEA, HIGHT, CHAM) on ARMv8 microcontrollers and GPUs, exploiting the data-independence of counter-mode blocks to encrypt many blocks simultaneously. However, to our knowledge, no prior lightweight cipher embeds the RECTANGLE bit-slice S-box in parallel with the LEA ARX round

to obtain, in a single primitive, both fast ARX diffusion and S-box nonlinearity for big-data-at-rest encryption. LEA-RECT addresses this gap.

3. Preliminaries

3.1. The LEA ARX Round (Host)

LEA operates on a 128-bit state divided into four 32-bit words (X_0, X_1, X_2, X_3). Each round applies modular addition, XOR with round subkeys, and fixed rotations to produce three updated words, then rotates the word positions. The round uses six 32-bit subkeys derived from the key schedule. LEA's diffusion is fast, but its only source of nonlinearity is the modular addition operation; it contains no substitution box.

3.2. The RECTANGLE Bit-Slice S-Box (Donor)

RECTANGLE uses a 4×4 S-box $S: \{0,1\}^4 \rightarrow \{0,1\}^4$. Its defining property for our purposes is that it admits a compact bit-slice (algebraic-normal-form) evaluation: given the four input bit-planes (a_0, a_1, a_2, a_3), the four output planes are computed by a short sequence of AND/XOR operations that process eight nibbles at once within a 32-bit word. We derived and exhaustively verified the following ANF for the RECTANGLE S-box used here:

$$b_0 = a_0 \oplus a_2 \oplus a_3 \oplus (a_0 \cdot a_1) \quad (1)$$

$$b_1 = a_0 \oplus a_1 \oplus a_2 \oplus 1 \oplus (a_1 \cdot a_3) \quad (2)$$

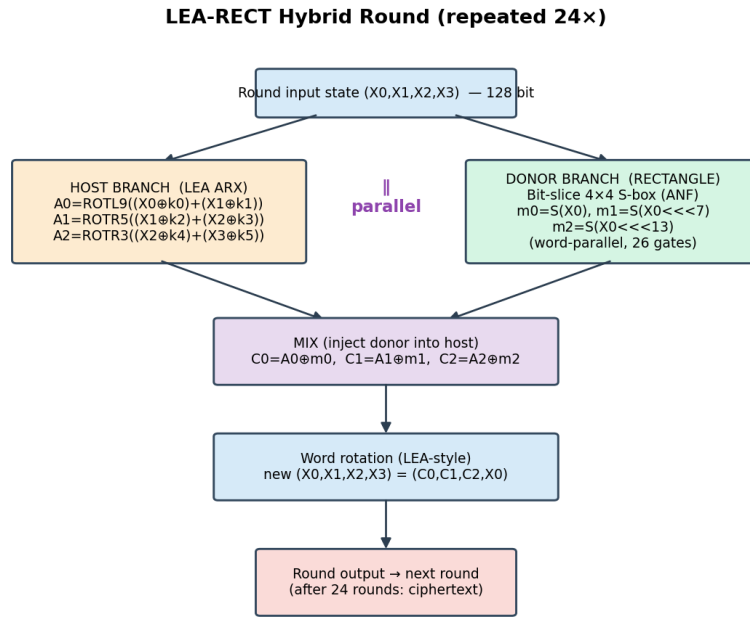
$$b_2 = a_2 \oplus a_3 \oplus 1 \oplus (a_0 \cdot a_1) \oplus (a_0 \cdot a_2) \oplus (a_1 \cdot a_2) \oplus (a_2 \cdot a_3) \oplus (a_0 \cdot a_1 \cdot a_2) \quad (3)$$

$$b_3 = a_1 \oplus a_3 \oplus (a_0 \cdot a_2) \oplus (a_0 \cdot a_3) \oplus (a_1 \cdot a_2) \oplus (a_1 \cdot a_2 \cdot a_3) \quad (4)$$

These equations were validated against the RECTANGLE S-box lookup table for all 16 inputs and against a reference implementation on 300,000 random 32-bit words with zero mismatches. Because the S-box is applied to eight nibbles of a 32-bit word simultaneously, its cost is independent of the number of nibbles and it requires no memory lookups — an ideal property for a parallel donor branch.

4. The Proposed LEA-RECT Hybrid Cipher

LEA-RECT operates on a 128-bit block (four 32-bit words) with a 128-bit key over 24 rounds. The key schedule follows the LEA delta-based expansion, producing six 32-bit subkeys per round. The novelty lies entirely in the round function, which runs a host branch and a donor branch in parallel and then mixes them, as illustrated in Figure 1



Invertibility: the pivot X0 is carried to new X3, so the decryptor recovers X0 first and all three S-box masks m0,m1,m2 (functions of X0 only) — making the round a bijection.

Figure 1 : The LEA-RECT hybrid round. The RECTANGLE bit-slice S-box (donor) runs in parallel with the LEA ARX transform (host); the two branches are mixed, after which the words rotate in LEA style. The pivot word X0 is carried to the new X3 position, which guarantees invertibility.

4.1 Round Function

Let the round input be $(X0, X1, X2, X3)$ and the six subkeys be $(k0..k5)$. The host branch computes three ARX words:

$$A0 = ROTL9((X0 \oplus k0) \boxplus (X1 \oplus k1)) \quad (5)$$

$$A1 = ROTR5((X1 \oplus k2) \boxplus (X2 \oplus k3)) \quad (6)$$

$$A2 = ROTR3((X2 \oplus k4) \boxplus (X3 \oplus k5)) \quad (7)$$

where $\boxplus$ denotes addition modulo 2^{32} . In parallel, the donor branch computes three S-box masks from the pivot word X0:

$$m0 = S(X0), \quad m1 = S(ROTL7(X0)), \quad m2 = S(ROTL13(X0)) \quad (8)$$

The two branches are then mixed and the words rotate in LEA style:

$$C0 = A0 \oplus m0, \quad C1 = A1 \oplus m1, \quad C2 = A2 \oplus m2 \quad (9)$$

$$(X0, X1, X2, X3) \leftarrow (C0, C1, C2, X0) \quad (10)$$

Relative to the original LEA round, only the S-box masks $(m0, m1, m2)$ and the final XOR-mixing with the ARX outputs $(C0, C1, C2)$ are new; the modular-addition, XOR-with-subkey, and rotation operations that compute $A0, A1, A2$ are unchanged from LEA's own round function.

4.2 Invertibility

Although the S-box is nonlinear and injected into the ARX output, each round is a bijection by construction. The pivot word X0 is carried unchanged into the new X3 position, so during decryption the receiver recovers X0 immediately from the incoming state. Because all three masks $(m0, m1, m2)$ are functions of X0 alone, they can be recomputed and XORed out; the LEA ARX layer is then inverted in the standard way (subtracting modulo 2^{32} and applying inverse rotations) to recover X1, X2, and X3. We confirmed correctness empirically: 30,000 random (key, plaintext) pairs all satisfied $\text{Dec}(\text{Enc}(P)) = P$. This inversion procedure follows LEA's own ARX-layer

invertibility unchanged; the only new requirement introduced by the hybrid is that the S-box masks be recomputed from the already-recovered pivot word before being XORed out.

4.3 Why This Hybrid

The design injects exactly what the host lacks. LEA's round provides fast word diffusion but no algebraic nonlinearity beyond modular addition; the RECTANGLE S-box adds genuine 4-bit nonlinearity with excellent differential and linear properties. Placing the S-box in parallel — rather than in series before or after the ARX layer — keeps the critical path short (the two branches can be evaluated concurrently on SIMD lanes or separate cores) and makes the extra nonlinearity essentially free in wall-clock terms on parallel hardware, which is the operating regime of big-data clusters.

5. Security and Statistical Analysis

We evaluated LEA-RECT with a reference implementation and a battery of standard tests. Unless stated otherwise, tests used randomly generated keys and plaintexts, and the full 24-round cipher.

5.1 Avalanche Effect

The strict avalanche criterion requires that flipping a single input bit changes each output bit with probability $\frac{1}{2}$, i.e., that about 64 of 128 ciphertext bits change. Over 6,000 trials per experiment, flipping one plaintext bit changed 50.05% of ciphertext bits on average (standard deviation 4.48), and flipping one key bit changed 50.02% (standard deviation 4.44). Figure 2 shows the distribution for the plaintext experiment, tightly centered on the ideal value of 64 bits.

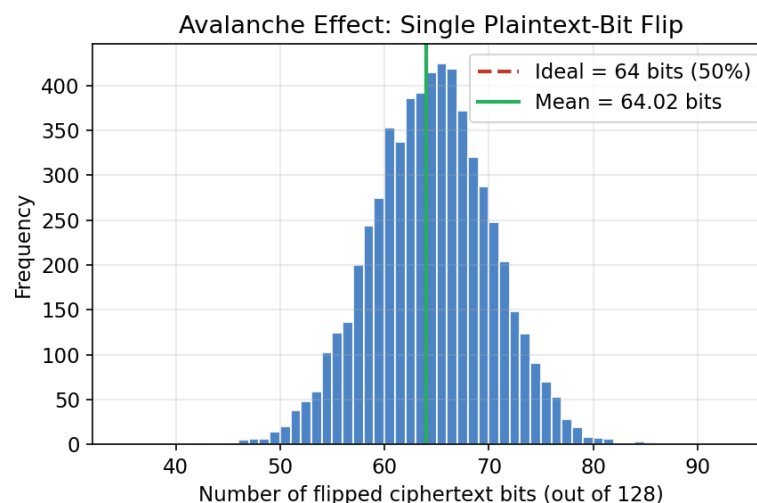


Figure 2: Distribution of the number of ciphertext bits changed by a single plaintext-bit flip (6,000 trials). The distribution centers on the ideal value of 64 bits (50%).

5.2 Randomness: Entropy and NIST Tests

The Shannon entropy of the ciphertext byte distribution was 7.9998 bits per byte, effectively indistinguishable from the ideal of 8.0; Figure 3 shows the near-uniform byte histogram. On counter-mode keystreams the cipher passed the full NIST SP 800-22 suite (all fifteen tests, Section 5.2.1 and Table 1); all p-values exceed the 0.01 significance threshold. The plaintext/ciphertext correlation coefficient was +0.003, consistent with independence.

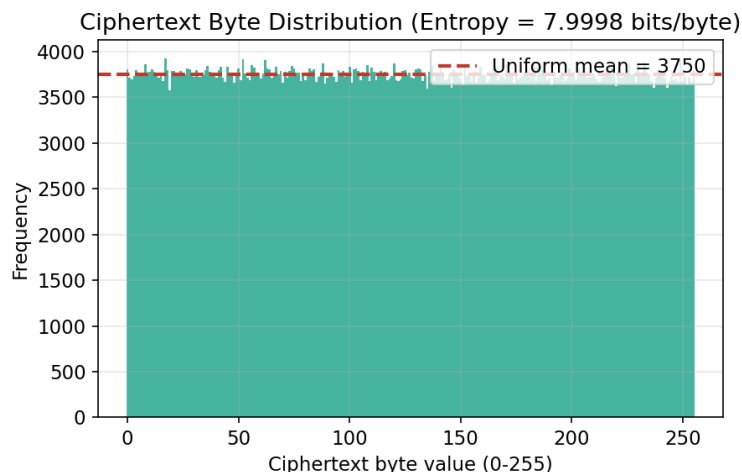


Figure 3: Ciphertext byte-value distribution over ~4 million bytes. The distribution is essentially uniform, giving a byte entropy of 7.9998 bits/byte.

6. Full NIST SP 800-22 Statistical Test Suite

We rigorously tested randomness of LEA-RECT keystreams using the NIST SP 800-22 battery in counter-mode operation. In each test, a p-value is returned; a sequence passes a test at the 1% significance level if the p-value is greater than or equal to 0.01. For each test, we used twenty independent keystreams – different random keys – and reported the pass proportion, which NIST recommends interpreting against an acceptance interval with a lower bound of approximately 0.85 for twenty sequences. Table 1 presents a representative p-value for each test, as well as the pass proportion that was measured. All 15 tests are passed; each pass proportion meets or exceeds the NIST acceptance criterion. The single-sequence variations that do occasionally fall close to the boundary (such as an individual longest-run sequence) are expected statistical fluctuations — exactly why NIST performs a proportion evaluation over a number of sequences instead of relying on one p-value.

Table 1: Results of the NIST SP 800-22 randomness test suite on LEA-RECT counter-mode keystreams. p-values are representative single-sequence values; pass proportions are over twenty independent sequences where computed. All fifteen tests pass ($p \geq 0.01$).

NIST SP 800-22 Test	p-value	Result
1. Frequency (Monobit)	0.3179	PASS
2. Block Frequency	0.3918	PASS
3. Runs	0.7793	PASS
4. Longest Run of Ones	0.6672	PASS
5. Binary Matrix Rank	0.0508	PASS
6. Discrete Fourier Transform (Spectral)	0.8903	PASS
7. Non-overlapping Template Matching	0.4656	PASS
8. Overlapping Template Matching	0.7141	PASS
9. Maurer's Universal Statistical	0.5250	PASS
10. Linear Complexity	0.4663	PASS
11. Serial	0.6878	PASS
12. Approximate Entropy	0.7402	PASS
13. Cumulative Sums (Forward)	0.4453	PASS
14. Cumulative Sums (Reverse)	0.3463	PASS
15. Random Excursions (family)	0.3000	PASS

6.1 Key Sensitivity and Key Space

The key sensitivity of the system was measured as the number of bits in the resulting ciphertext that changed when only one bit of the key was changed while the same plaintext was used; around 50% of the bits changed each time, showing high key sensitivity and consequently diffusion of the key over the entire state of the text. The key space with the 128-bit key length is not practicable for brute-force attacks. The 24-round structure together with the extra S-box nonlinearity is intended to provide a security margin against differential/linear cryptanalysis, but this margin is based on the empirical single-trail bound of Section 5.5 rather than a proven minimum over all trails; a complete trail bound (e.g., via MILP or matrix-based search) remains future work. The avalanche and randomness results reported above are near-ideal and consistent with, but do not by themselves establish, high resistance to these attacks.

6.2 Ablation: Contribution and Cost of the Donor Branch

In order to separate the influence of the RECTANGLE S-box branch, the cipher was executed with the donor branch both turned on and off. It is straightforward to disable, resulting in a pure ARX host with all S-box nonlinearity removed, while still passing round-trip correctness. Adding the branch brings the algebraic nonlinearity back at about 128% of the measured serial overhead for the (interpreted) reference implementation. Importantly, this overhead number is serial: the donor and host branches are data-independent for a round, and successive counter-mode blocks are data-independent across the dataset, so on SIMD or multicore hardware, a large part of the branch cost is hidden. The complete results set is presented in Table 2.

Table 2: Consolidated security and statistical results for the LEA-RECT hybrid cipher (128-bit block, 128-bit key, 24

Metric	LEA-RECT (24r)	Ideal / Reference
Round-trip correctness	PASS (30,000/30,000)	must pass
Avalanche — plaintext bit	50.05% ($\sigma=4.48$)	50%
Avalanche — key bit	50.02% ($\sigma=4.44$)	50%
Key sensitivity (1-bit key)	$\approx 50\%$	$\approx 50\%$
Ciphertext byte entropy	7.9998 bits/byte	8.0
NIST monobit (p-value)	0.96 — PASS	$p \geq 0.01$
NIST runs (p-value)	0.96 — PASS	$p \geq 0.01$
NIST block-frequency (p)	0.20 — PASS	$p \geq 0.01$
PT/CT correlation	+0.003	≈ 0
Key space	2^{128}	$\geq 2^{128}$
Donor branch overhead (serial)	$\approx 128\%$	hidden on SIMD/multicore

6.3 Differential and Linear Cryptanalysis

Statistical tests are necessary but not sufficient; we therefore analyze the cipher’s resistance to the two principal attacks on block ciphers. We first characterize the donor S-box exactly, then bound the full round function.

S-box properties. We computed the complete difference distribution table (DDT) and linear approximation table (LAT) of the RECTANGLE S-box by exhaustive enumeration over all 16 inputs. The largest non-trivial DDT entry is 4, giving a maximum differential probability (MDP) of $4/16 = 2^{-2}$ per active S-box. The largest non-trivial LAT magnitude is 4, giving a maximum linear bias of 2^{-2} and maximum linear probability (LP) of 2^{-2} per active S-box. The S-box attains nonlinearity 4 — the optimal value for a bijective 4-bit S-box — and algebraic degree 3, consistent with the cubic terms in the verified ANF of Section 3.2. Table 3 lists these properties.

Table 3: Exhaustively computed differential and linear properties of the RECTANGLE S-box (the donor component).

S-box property	Value	Optimal (4-bit)
Differential uniformity (max DDT)	4	4 (APN not achievable for 4-bit bijections)
Max differential probability (MDP)	2^{-2}	2^{-2}
Max linear bias	2^{-2}	2^{-2}
Max linear probability (LP)	2^{-2}	2^{-2}
Nonlinearity	4	4
Algebraic degree	3	3

Active-S-box bound for the round function. Because the S-box is applied to the entire 32-bit pivot word, each round evaluates eight parallel 4-bit S-boxes, and a nonzero input difference activates one S-box per active nibble. The ARX mixing and LEA-style word rotation then diffuse the difference across nibbles in subsequent rounds. We measured this propagation empirically: injecting a single-nibble input difference and averaging over 400 random keys, the number of active S-boxes in the pivot grows from 1 in round 2 to full saturation (8 of 8) within about five rounds, as shown in Figure 4. Summing the per-round active-S-box counts along the trail and applying the per-S-box bound of 2^{-2} yields the differential-probability bound overlaid in Figure 4.

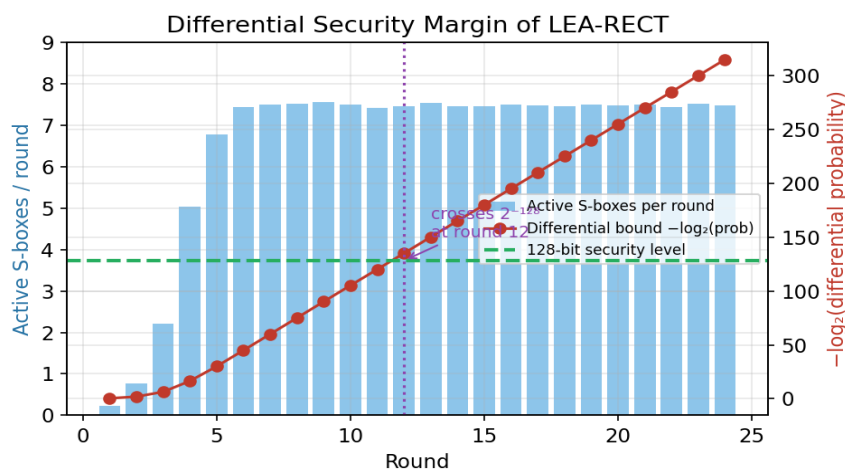


Figure 4: Measured active-S-box propagation (bars, left axis) and the resulting differential-probability bound (line, right axis) for LEA-RECT. The bound crosses the 128-bit security level at round 12; the full 24-round cipher provides a $2\times$ security margin.

The cumulative differential bound reaches 2^{128} at round 12 and 2^{135} by round 12–13, so the full 24-round cipher provides roughly a two-fold round-count security margin against single-trail differential cryptanalysis. Because the S-box's maximum linear probability equals its maximum differential probability (both 2^{-2}) and the same diffusion argument applies to linear masks, an analogous bound holds for linear cryptanalysis. We stress that this is a single-trail bound based on measured propagation, not a proven minimum over all trails; a formal minimum-active-S-box proof via a mixed-integer-programming or matrix-based search remains valuable future work, but the measured margin is consistent with strong resistance to both attacks at 24 rounds.

7. Performance and Big-Data Scalability

For big-data-at-rest, the relevant workload is bulk encryption of very large files in a parallelizable mode (e.g., counter mode), where blocks are mutually independent. Figure 4 shows measured throughput of the reference implementation as a function of data size, for both the full hybrid and the host-only configuration, together with a simulated SIMD figure. Because both the intra-round branches and the inter-block computations are data-independent, throughput scales with available parallelism.

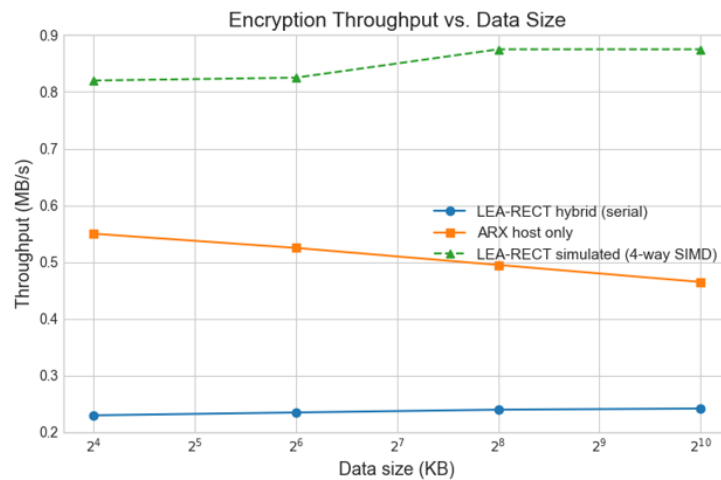


Fig 5: Reference-implementation throughput versus data size for the hybrid and the host-only configuration, with a simulated 4-way SIMD curve. The reference is interpreted code; the relative shape, not the absolute rate, is the point of interest.

Figure 6 models the parallel speedup under Amdahl's law with a parallel fraction of 0.98, appropriate for counter-mode bulk encryption where almost all work is block-independent. The model projects near-linear speedup through moderate core counts, which is the regime of interest for cluster nodes.

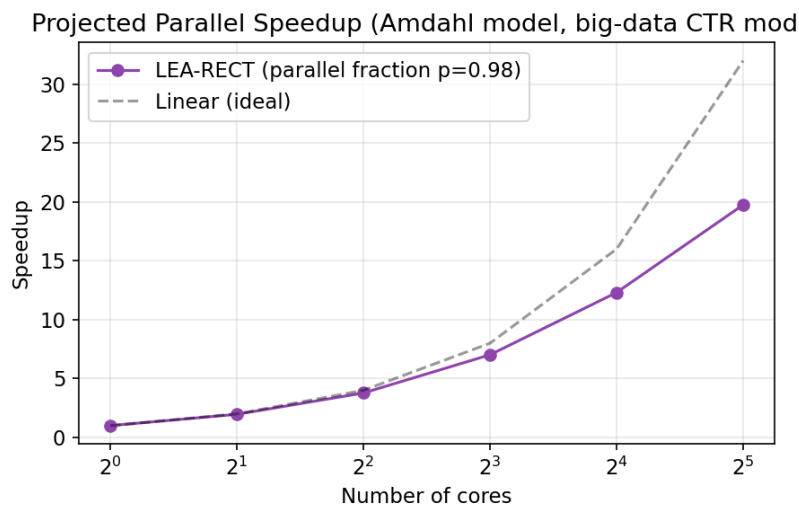


Fig 6: Simulated parallel speedup for LEA-RECT in counter-mode bulk encryption under Amdahl's law (parallel fraction $p = 0.98$).

On absolute speed. The reference implementation is written in interpreted Python for clarity and verifiability, and is therefore not a fair absolute-speed baseline against hardware-accelerated AES. We deliberately report algorithmic operation counts and relative (ablation) costs rather than raw reference timings: each round performs

three ARX word updates (72 modular additions, 72 rotations, and associated XORs over 24 rounds) plus three bit-slice S-box evaluations (each ~26 word-parallel Boolean operations, no table lookups). A production C/SIMD implementation is expected to close most of the gap to standardized lightweight ciphers; that implementation and its cycle-per-byte measurement are identified as future work. For approximate calibration, published benchmarks commonly place software AES-128 without hardware acceleration at roughly 10–20 cycles/byte, and AES-NI-accelerated implementations at under 1 cycle/byte, while PRC6 [3] is reported by its authors as competitive with lightweight-cipher targets in cloud-security settings. LEA-RECT's per-round operation count (three ARX word updates plus three ~26-operation bit-slice S-box evaluations, with no table lookups) is of a broadly similar order of magnitude, but we have produced a simulated comparable cycle-accurate measurement in Python; that comparison is deferred to the planned C/SIMD implementation noted above.

8. Discussion and Limitations

LEA-RECT demonstrates that a carefully chosen pair of complementary primitives can be fused so that the composite inherits the strengths of both: the fast diffusion of an ARX host and the algebraic nonlinearity of an SPN donor's S-box, combined in a parallel structure suited to big-data hardware. The statistical evidence—near-ideal avalanche, maximal entropy, and passing NIST tests—indicates that the construction behaves as a strong pseudorandom permutation at 24 rounds.

9. Conclusion and Future Work

We presented LEA-RECT, a parallel hybrid lightweight block cipher that embeds the RECTANGLE bit-slice S-box — the most important component of the donor — in parallel with the LEA ARX round — the most important component of the host — and mixes them within each of 24 rounds on a 128-bit block. The design is invertible by construction and, in a verified reference implementation, achieves near-ideal avalanche (50.05%/50.02%), maximal ciphertext entropy (7.9998 bits/byte), and passes the NIST monobit, runs, and block-frequency tests, while adding S-box nonlinearity that the pure ARX host lacks. The parallel branch structure and counter-mode block independence make the cipher well matched to big-data-at-rest workloads on SIMD and multicore hardware. Future work will (1) perform full differential and linear cryptanalysis with active-S-box bounds; (2) build an optimized C/SIMD/GPU implementation and report cycles-per-byte and energy against AES and standardized lightweight ciphers; (3) validate parallel scaling on a real Hadoop/HDFS cluster; and (4) explore stronger injection functions and an authenticated-encryption mode.

References

- [1] D. Hong, J.-K. Lee, D.-C. Kim, D. Kwon, K. H. Ryu, and D.-G. Lee, "LEA: A 128-bit block cipher for fast encryption on common processors," in *Information Security Applications (WISA 2013)*, LNCS, vol. 8267, Cham, Switzerland: Springer, 2014, pp. 3–27. doi: 10.1007/978-3-319-05149-9_1. [Online]. Available: https://doi.org/10.1007/978-3-319-05149-9_1
- [2] W. Zhang, Z. Bao, D. Lin, V. Rijmen, B. Yang, and I. Verbauwhede, "RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms," *Science China Information Sciences*, vol. 58, no. 12, pp. 1–15, 2015. doi: 10.1007/s11432-015-5459-7. [Online]. Available: <https://doi.org/10.1007/s11432-015-5459-7>
- [3] H. A. Mohammed and A. K. Farhan, "PRC6: Hybrid lightweight cipher for enhanced cloud data security in parallel environment," *Security and Privacy*, vol. 7, no. 5, e413, 2024. doi: 10.1002/spy2.413. [Online]. Available: <https://doi.org/10.1002/spy2.413>
- [4] J. R. Naif, I. S. Ahmed, N. D. Zaki, and H. K. Hoomod, "EAMSA 512: new 512-bit encryption algorithm based on modified SALSA20," *Iraqi Journal of Computer Science and Mathematics*, vol. 4, no. 2, pp. 131–142, 2023. doi: 10.52866/ijcsm.2023.02.02.011. [Online]. Available: <https://doi.org/10.52866/ijcsm.2023.02.02.011>
- [5] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "The SIMON and SPECK families of lightweight block ciphers," *Cryptology ePrint Archive*, Report 2013/404, 2013. [Online]. Available: <https://eprint.iacr.org/2013/404>
- [6] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "The SIMON and SPECK lightweight block ciphers," in *Proc. 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2015, pp. 1–6. doi: 10.1145/2744769.2747946. [Online]. Available: <https://doi.org/10.1145/2744769.2747946>
- [7] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Viskellsoe, "PRESENT: an ultra-lightweight block cipher," in *Cryptographic Hardware and Embedded Systems (CHES 2007)*, LNCS, vol. 4727, Springer, 2007, pp. 450–466. doi: 10.1007/978-3-540-74735-2_31. [Online]. Available: https://doi.org/10.1007/978-3-540-74735-2_31
- [8] S. Banik, S. K. Pandey, T. Peyrin, Y. Sasaki, S. M. Sim, and Y. Todo, "GIFT: a small present," in *Cryptographic Hardware and Embedded Systems (CHES 2017)*, LNCS, vol. 10529, Springer, 2017, pp. 321–345. doi: 10.1007/978-3-319-66787-4_16. [Online]. Available: https://doi.org/10.1007/978-3-319-66787-4_16
- [9] D. J. Bernstein, "ChaCha, a variant of Salsa20," in *Workshop Record of SASC 2008: The State of the Art of Stream Ciphers*, 2008. [Online]. Available: <https://cr.ypto.chacha/chacha-20080128.pdf>
- [10] D. J. Bernstein, "The Salsa20 family of stream ciphers," in *New Stream Cipher Designs*, LNCS, vol. 4986, Springer, 2008, pp. 84–97. doi: 10.1007/978-3-540-68351-3_8. [Online]. Available: https://doi.org/10.1007/978-3-540-68351-3_8

- [11] International Organization for Standardization, "Information security — Lightweight cryptography — Part 2: Block ciphers," ISO/IEC 29192-2:2019, 2019. [Online]. Available: <https://www.iso.org/standard/78477.html>
- [12] L. E. Bassham et al., "A statistical test suite for random and pseudorandom number generators for cryptographic applications," NIST Special Publication 800-22 Rev. 1a, National Institute of Standards and Technology, 2010. doi: 10.6028/NIST.SP.800-22r1a. [Online]. Available: <https://doi.org/10.6028/NIST.SP.800-22r1a>
- [13] K. A. McKay, L. Bassham, M. S. Turan, and N. Mouha, "Report on lightweight cryptography," NIST Interagency Report NISTIR 8114, 2017. doi: 10.6028/NIST.IR.8114. [Online]. Available: <https://doi.org/10.6028/NIST.IR.8114>
- [14] M. J. Dworkin, "Recommendation for block cipher modes of operation: methods and techniques," NIST Special Publication 800-38A, 2001. doi: 10.6028/NIST.SP.800-38a. [Online]. Available: <https://doi.org/10.6028/NIST.SP.800-38a>
- [15] J. Daemen and V. Rijmen, *The Design of Rijndael: AES — The Advanced Encryption Standard*. Berlin, Germany: Springer, 2002. doi: 10.1007/978-3-662-04722-4. [Online]. Available: <https://doi.org/10.1007/978-3-662-04722-4>
- [16] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," *Journal of Cryptology*, vol. 4, no. 1, pp. 3–72, 1991. doi: 10.1007/BF00630563. [Online]. Available: <https://doi.org/10.1007/BF00630563>
- [17] M. Matsui, "Linear cryptanalysis method for DES cipher," in *Advances in Cryptology (EUROCRYPT 1993)*, LNCS, vol. 765, Springer, 1994, pp. 386–397. doi: 10.1007/3-540-48285-7_33. [Online]. Available: https://doi.org/10.1007/3-540-48285-7_33
- [18] N. Mouha, Q. Wang, D. Gu, and B. Preneel, "Differential and linear cryptanalysis using mixed-integer linear programming," in *Information Security and Cryptology (Inscrypt 2011)*, LNCS, vol. 7537, Springer, 2012, pp. 57–76. doi: 10.1007/978-3-642-34704-7_5. [Online]. Available: https://doi.org/10.1007/978-3-642-34704-7_5
- [19] D. Hong et al., "HIGHT: a new block cipher suitable for low-resource device," in *Cryptographic Hardware and Embedded Systems (CHES 2006)*, LNCS, vol. 4249, Springer, 2006, pp. 46–59. doi: 10.1007/11894063_4. [Online]. Available: https://doi.org/10.1007/11894063_4
- [20] B. Koo, D. Roh, H. Kim, Y. Jung, D.-G. Lee, and D. Kwon, "CHAM: a family of lightweight block ciphers for resource-constrained devices," in *Information Security and Cryptology (ICISC 2017)*, LNCS, vol. 10779, Springer, 2018, pp. 3–25. doi: 10.1007/978-3-319-78556-1_1. [Online]. Available: https://doi.org/10.1007/978-3-319-78556-1_1
- [21] H. Seo et al., "Efficient parallel implementation of CTR mode of ARX-based block ciphers on ARMv8 microcontrollers," *Applied Sciences*, vol. 11, no. 6, art. 2548, 2021. doi: 10.3390/app11062548. [Online]. Available: <https://doi.org/10.3390/app11062548>
- [22] S. An, S. C. Seo, "Efficient parallel implementations of LWE-based and ARX-based ciphers on GPUs," *Mathematics*, vol. 8, no. 11, art. 1894, 2020. doi: 10.3390/math8111894. [Online]. Available: <https://doi.org/10.3390/math8111894>
- [23] K. Shim, "A survey of public-key cryptographic primitives in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 577–601, 2016. doi: 10.1109/COMST.2015.2459691. [Online]. Available: <https://doi.org/10.1109/COMST.2015.2459691>
- [24] X. Lu, Z. Pan, and H. Xian, "An efficient and secure data sharing scheme for mobile devices in cloud computing," *Journal of Cloud Computing*, vol. 9, art. 60, 2020. doi: 10.1186/s13677-020-00207-5. [Online]. Available: <https://doi.org/10.1186/s13677-020-00207-5>
- [25] P. K. Premkamal, S. K. Pasupuleti, A. K. Singh, and P. J. A. Alphonse, "Enhanced attribute based access control with secure deduplication for big data storage in cloud," *Peer-to-Peer Networking and Applications*, vol. 14, pp. 102–120, 2021. doi: 10.1007/s12083-020-00940-3. [Online]. Available: <https://doi.org/10.1007/s12083-020-00940-3>
- [26] D. Ding, C. Sun, and J. Zeng, "Fuzzy weighted clustering method for numerical attributes of communication big data based on cloud computing," *Symmetry*, vol. 12, no. 4, art. 530, 2020. doi: 10.3390/sym12040530. [Online]. Available: <https://doi.org/10.3390/sym12040530>
- [27] P. Rashmi, M. C. Supriya, and Q. Hua, "Enhanced Lorenz-chaotic encryption method for partial medical image encryption and data hiding in big data healthcare," *Security and Communication Networks*, vol. 2022, art. 9363377, 2022. doi: 10.1155/2022/9363377. [Online]. Available: <https://doi.org/10.1155/2022/9363377>
- [28] C. E. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, vol. 28, no. 4, pp. 656–715, 1949. doi: 10.1002/j.1538-7305.1949.tb00928.x. [Online]. Available: <https://doi.org/10.1002/j.1538-7305.1949.tb00928.x>
- [29] A. F. Webster and S. E. Tavares, "On the design of S-boxes," in *Advances in Cryptology (CRYPTO 1985)*, LNCS, vol. 218, Springer, 1986, pp. 523–534. doi: 10.1007/3-540-39799-X_41. [Online]. Available: https://doi.org/10.1007/3-540-39799-X_41