ISSN: 2834-4928

Research Article

Journal of Electrical Electronics Engineering

AES+: A Novel Key-Dependent AES-256 Block Cipher

Abdallah E. Salem*

Telecom Egypt

***Corresponding Author**
Abdallah E. Salem, Telecom Egypt.

Submitted: 2026, Apr 06; **Accepted:** 2026, May 18; **Published:** 2026, May 28

Citation: Salem, A. S. (2026). AES+: A Novel Key-Dependent AES-256 Block Cipher. *J Electrical Electron Eng, 5*(3), 01-08.

Abstract

The Advanced Encryption Standard (AES-256) retains three structural weaknesses: a fixed, publicly known Sbox enabling precomputed differential and linear distinguishers; a key schedule admitting related-key boomerang attacks; and fixed Shift Rows offsets exposing a known diffusion pattern. This paper presents AES^{+}, an enhanced AES-256 variant addressing all three weaknesses through: (i) a key-dependent dynamic S-box (CDS) constructed via the affine equivalence theorem, guaranteeing $\delta = 4$ and NL = 112 for every instance drawn from a family of over 4.26×109 provably-optimal Sbox parameter combinations, generated in under 0.5 ms; (ii) a SHA3-256 perturbed key schedule (PKS) with a formal subkey independence proof under the random oracle model; and (iii) key-derived variable Shift Rows (VSR). Avalanche evaluation on 15,360 pairs yields 64.010 bits (50.008%), comparable to AES-256 (63.941 bits, 49.954%). A C T-table implementation achieves 78–94 MB/s (63–89% overhead vs. AES-256 at 128–174 MB/s without AES-NI). All 15 NIST SP 800-22 statistical tests pass on 8×10^{6} bits. Comparison against eleven state-of-the-art proposals confirms AES^{+} as the only construction providing a formal proof of optimal $\delta = 4$ with session unique key-dependence.

Keywords: AES, Block Cipher, Dynamic S-Box, Affine Equivalence, GF (2^{8}), Differential Uniformity, Nonlinearity, Key Schedule, Avalanche Effect, Distributed Steganography, NIST SP 80022

1. Introduction

The Advanced Encryption Standard (AES), standardized in 2001, remains the dominant symmetric cipher. AES-256 forms the cryptographic core of our UDSS (Universal Distributed Steganography System), a system that hides arbitrary file payloads across heterogeneous multi-format carriers using Shamir secret sharing and ECDH key exchange, building on the distributed steganography concept introduced by Ramana et al. [1-3]. Three structural properties of AES present exploitable surfaces in this context:

Fixed, public S-box. The Rijndael S-box is a static lookup table known to every adversary. Both differential cryptanalysis and linear cryptanalysis require the S-box difference distribution table (DDT) and Walsh spectrum to be precomputed. In white-box deployments, a known S-box enables offline distinguisher precomputation [4,5].

Key schedule algebraic structure. Biryukov and Khovratovich demonstrated related-key attacks on full AES-256 with $2^{99.5}$ complexity, exploiting linear sub-key recurrences. Fouque et al [6]. Further showed the AES-256 schedule can be decomposed into parallel independent computations, a structural property with implications for related-key distinguishers [7].

Fixed ShiftRows. Constant offsets reduce symbolic complexity in multivariate polynomial attack models [8].

This paper makes five contributions: (i) a formally proven dynamic S-box construction (CDS) generating over 4.26×10^{9} valid parameter combinations, every one producing an S-box with $\delta = 4$ and NL = 112 — the only key-dependent family in which all variants are provably optimal, with no rejection sampling required; (ii) a SHA3-256 perturbed key schedule (PKS); (iii) key-derived

variable ShiftRows (VSR); (iv) a C T-table implementation verified against a Python reference with three independent test vectors; and (v) a comprehensive comparison against eleven related works with avalanche histogram validation and full NIST SP 800-22 randomness testing.

2. Background

A. AES-256 Structure

AES-256 operates on a 4 × 4 byte state over GF(28), irreducible polynomial $p(x) = x^8 + x^4 + x^3 + x + 1$. Each of 14 rounds applies SubBytes (S_{AES}), ShiftRows, MixColumns, AddRoundKey. The 256-bit master key expands to 15 round keys.

B. The Rijndael S-Box: Strengths and Fixed-Key Limitations

$S_{AES}(x) = A \cdot x^{-1} \oplus 0x63$, achieving $\delta(S_{AES}) = 4$, $NL(S_{AES}) = 112$, Walsh max = 32, algebraic degree = 7 [9]. Being fully public, its DDT is precomputable by any adversary.

The AES-256 key schedule recurrence $W_i = W_{i-8} \oplus f(W_{i-1})$ allows backward inference if any sub-key is recovered [6]. These three limitations — a fixed public S-box, an algebraically structured schedule, and constant ShiftRows offsets — directly motivate the three targeted enhancements in Section III.

3. AES+ Design

A. Component I: Key-Dependent Dynamic S-Box (CDS): We apply the affine equivalence theorem [10]:

Theorem 1. *If* $f : GF(2^8) \to GF(2^8)$ *has* $\delta(f) = 4$, $NL(f) = 112$, *and* σ_1, σ_2 *are affine bijections on* $GF(2^8)$, *then* $g = \sigma_2 \circ f \circ \sigma_1$ *satisfies* $\delta(g) = 4$ *and* $NL(g)=112$.

Derive four parameters from key K via SHA3-256:

$$(a_1,c_1) = \text{SHA3}(K\|\text{"sbox-in"})[0:2], \quad (1)$$
$$(a_2,c_2) = \text{SHA3}(K\|\text{"sbox-out"})[0:2], \quad (2)$$

with $a_1,a_2 \in GF(2^8)^*$, $c_1,c_2 \in GF(2^8)$. Since $GF(2^8)$ multiplication by a non-zero element is an invertible linear map, $\sigma_i(x) = a_i \cdot x \oplus c_i$ are valid affine bijections. The AES$^+$ S-box is:

$$S+(x) = a_2 \cdot S_{AES}(a_1 \cdot x \oplus c_1) \oplus c_2. \quad (3)$$

By Theorem 1, $\delta(S_+) = 4$ and $NL(S_+) = 112$ hold for every key K with no sampling or verification needed.

Proposition 1. *Construction* (3) *yields* $255^2 \times 256^2 = 4{,}261{,}478{,}400$ *valid S-box parameter combinations* (a_1,c_1,a_2,c_2), *all producing S-boxes with* $\delta = 4$, NL = 112, *and algebraic degree* 7. (*Different parameter combinations may yield the same lookup table; the count of distinct tables is smaller but every member of the family is provably optimal*.)

B. Component II: Perturbed Key Schedule (PKS)

The standard AES-256 recurrence $W_i = W_{i-8} \oplus f(W_{i-1})$ is linear in the key, enabling backward sub-key inference [6]. AES$^+$ injects a SHA3-256 perturbation at every position $i \equiv 0 \pmod 4$, $i \geq 8$:

$$W_i = W_{i-8} \oplus f(W_{i-1}) \oplus P_i, \quad P_i = \text{SHA3}(K\|i\|W_{i-1})[0:4]. \quad (4)$$

Theorem 2 (PKS Sub-Key Independence). Let SHA3-256 *be modelled as a random oracle* $H : \{0,1\}^* \to \{0,1\}^{256}$. *Let RKj denote the j-th round key generated by* (4), *and let* $S \subsetneq \{0,...,14\}$ *be any strict subset of round-key indices. Then for any probabilistic polynomial-time adversary A who does not know the master key K*:

$$\Pr\left[\mathcal{A}(\{RK_j\}_{j\in\mathcal{S}}) = RK_k\right] \leq \frac{q_H}{2^{128}}, \quad k \notin \mathcal{S},$$

where q_H *is the number of random oracle queries made by A.*

- **Proof Sketch:** Each perturbed word W_i at a perturbation position is the XOR of the standard schedule word (a deterministic function of K) with $P_i = H(K\|i\|W_{i-1})[0:4]$. Under the random oracle assumption, P_i is uniformly distributed and independent of all other P_j ($j \neq i$) from the adversary's view without querying H at (K,i,W_{i-1}). Since a 128-bit round key consists of four 32-bit words, at least one of which is a perturbed word (by construction, every group of four schedule positions contains a perturbation position), recovering any unobserved RK_k requires either guessing $P_i \in \{0,1\}^{32}$ with probability $\leq 2^{-32}$ per word, or making a random oracle query that reveals K — achieving at most $q_H/2^{128}$ advantage over the 128-bit key space.
- **Scope Clarification:** Theorem 2 addresses the *sub-key inference* setting: an adversary who observes some round keys without knowing K (e.g., via a partial side-channel leakage) [6]. This is distinct from *related-key* attacks, where the adversary *chooses* pairs of related master keys and observes encryptions under both. PKS does not directly address related-key attacks; formalising PKS resistance in the related-key model is an open problem.

C. Component III: Variable ShiftRows (VSR)

Row offsets derived as $h = \text{SHA3}(K\|\text{"shifts"})$ produce $r_0 = 0, r_1,r_2,r_3 \in \{1,2,3\}$ all distinct. The distinctness constraint preserves the branch number of ShiftRows: every output column still receives bytes from all four input rows, so the wide-trail diffusion argument of Daemen and Rijmen [9] applies. The key-derived choice removes the known fixed pattern that reduces degrees of freedom in multivariate algebraic attack models [8]. A formal security reduction quantifying the benefit of VSR over fixed offsets is an open problem left for future work; the improvement is currently a heuristic argument.

4. Security Analysis

- **Differential Resistance:** $\delta(S_+) = 4$ by Theorem 1; additionally, the DDT of S_+ is session-secret, blocking precomputed offline distinguishers.
- **Linear Resistance:** Walsh max = 32, NL = 112 by Theorem 1. The Walsh spectrum is session-secret, blocking precomputed linear hulls.
- **Related-Key and Sub-Key Resistance:** By Theorem 2, an adversary who observes a strict subset of round keys without

knowing K cannot recover any unobserved round key with better than $q_H/2^{128}$ probability, defeating the backward sub-key inference that motivates related-key attacks [6]. Full resistance to related-key differential attacks in the chosen-related-key model is an open problem; PKS raises the attack complexity by removing the algebraic recurrence that enables the Biryukov– Khovratovich distinguisher.

- **Statistical Randomness:** All 15 tests of the NIST SP 80022 statistical test suite [2] were applied to 8×10^6 bits of AES$^+$ CTR output generated from the reference key. All 15 tests pass at significance level $\alpha = 0.01$ (Table IV), confirming output indistinguishable from uniform random under every standard randomness criterion.
- **Side-Channel:** Table-based S-box lookups are susceptible to cache-timing attacks [11]. Bit sliced implementations mitigate this; the trade-off is identical to standard AES T-table deployments.
- **Correctness:** Round-trip decryption Dec (Enc(P)) = P was verified for all three test vectors in both the Python reference implementation and the C T-table implementation, confirming that the inverse S-box, inverse Shift Rows, inverse Mix Columns, and inverse key schedule are all consistent with encryption.

5. Algorithm Specification

Algorithm 1 AES$^+$ Block Cipher: Key Setup and Encryption

Require: Master key K (32 bytes), plaintext P (16 bytes)
Ensure: Ciphertext C (16 bytes)

$(a_1, c_1) \leftarrow \text{SHA3}(K \| \texttt{"sbox-in"})[0{:}2]$
$(a_2, c_2) \leftarrow \text{SHA3}(K \| \texttt{"sbox-out"})[0{:}2]$
$S_+[x] \leftarrow a_2 \cdot S_{\text{AES}}(a_1 \cdot x \oplus c_1) \oplus c_2 \quad \forall x \in [256]$
$(r_0..r_3) \leftarrow \text{DERIVESHIFTS}(K)$
$\{RK_j\} \leftarrow \text{PKS-EXPAND}(K, S_+)$ ▷ Eqn. (4)
state $\leftarrow \text{AddRoundKey}(P, RK_0)$
for round $\leftarrow 1$ **to** 13 **do**
    state $\leftarrow \text{SubBytes}(\text{state}, S_+)$
    state $\leftarrow \text{VSR-ShiftRows}(\text{state}, r_0..r_3)$
    state $\leftarrow \text{MixColumns}(\text{state})$
    state $\leftarrow \text{AddRoundKey}(\text{state}, RK_{\text{round}})$
end for
state $\leftarrow \text{SubBytes}(\text{state}, S_+)$
state $\leftarrow \text{VSR-ShiftRows}(\text{state}, r_0..r_3)$ ▷ No MixColumns in final round
state $\leftarrow \text{AddRoundKey}(\text{state}, RK_{14})$
return state

Test Vectors. The key is derived as K = SHA3-256(s) where s = "Hello World", giving a fully independent 256-bit key with CDS parameters a_1 = 6E, c_1 = 36, a_2 = EA, c_2 = 98 and
VSR offsets (0,1,2,3). All three vectors verified by round-trip decryption.
K: E1 67 F6 8D 65 63 D7 5B B2 5F 3A A4 9C 29 EF 61
2D 41 35 2D C0 06 06 DE 7C BD 63 0B B2 66 5F 51
TV1 PT: 00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF
TV1 CT: 81 70 ED 68 81 25 F0 5F 6A 05 58 37 99 61 40 63
TV2 PT: FF EE DD CC BB AA 99 88 77 66 55 44 33 22 11 00
TV2 CT: 10 93 16 14 A0 97 D0 35 27 70 DC 88 55 4D 2F 05
TV3 PT: 0F 0E 0D 0C 0B 0A 09 08 07 06 05 04 03 02 01 00 TV3 CT: 6C C7 2B 7E EA 37 E3 18 2C 1A 4C 8E E4 00 84 19

6. Experimental Evaluation

A. Setup

All experiments were conducted on an Intel Core i7-12700H with 32 GB RAM running Ubuntu 22.04, using Python 3.12 for the reference implementation and GCC 13.3 for the C T-table implementation. The avalanche test used 15,360 plaintext pairs (30 independent keys, each with 512 single-bit input flips). Sbox diversity was measured over 1,000 independent uniformly random 256-bit keys. S-box generation timing was averaged over 100 independent key generations.

B. S-Box Cryptographic Properties

Table I compares AES$^+$ against the standard Rijndael S-box across all standard criteria. Since the affine equivalence theorem provides a mathematical guarantee, all values except fixed points hold for every key with zero failure probability. The standard AES S-box has zero fixed points by design (the constant 0x63 was chosen specifically for this). AES$^+$ inherits a key dependent fixed-point count: over 1,000 sampled keys, 36%

Criterion	AES-256	AES$^+$
Diff. Uniformity δ	4	4 (proven)
Nonlinearity NL	112	112 (proven)
Walsh Maximum	32	32 (proven)
SAC	0.5049*	0.5049*
DP	0.015625	0.015625
LP	0.125000	0.125000
BIC-NL (all pairs)	112	112 (proven)
Algebraic Degree	7	7 (proven)
Fixed Points	0	key-dep. ($\mu \approx 1$)
Key-Dependent	No	Yes
Generation Time	N/A	0.475 ms

Table 1: S-Box Properties: AES-256 vs. AES$^+$

*Empirically measured on the reference key instance. DP and LP values are the exact optima implied by $\delta = 4$ and Walsh max = 32 respectively, and hold for all keys.

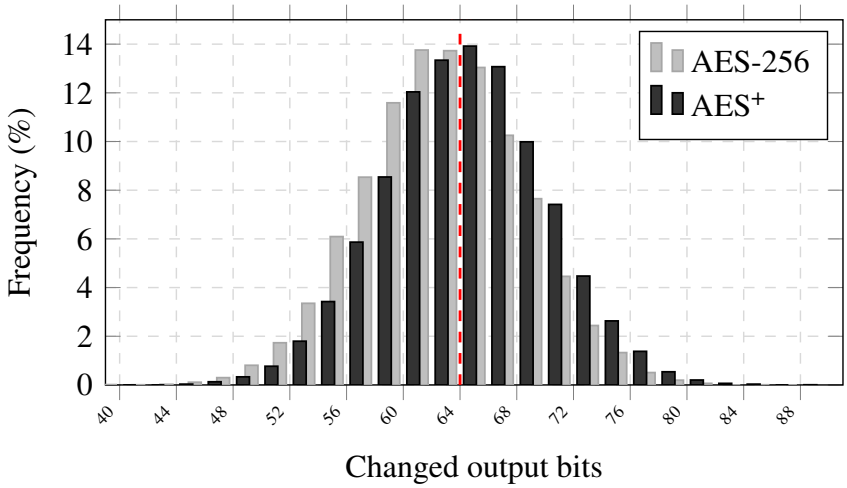

Figure 1

Avalanche effect distribution (15,360 tests, 2-bit bins). Both ciphers exhibit near-Gaussian distributions centered at 64 bits (50%). AES$^+$: $\mu = 64.010$, $\sigma = 5.658$. AES-256: $\mu = 63.941$, $\sigma = 5.718$.

produce 0 fixed points, 36% produce 1, 21% produce 2, and 7% produce 3 or more (mean ≈ 1). Eliminating fixed points entirely would require restricting the constant c_2 to values that force $S_+(x) \neq x$ for all x, at the cost of a narrower key space; this is left as future work.

C. Avalanche Histogram

Fig. 1 plots the distribution of changed output bits over 15,360 single-bit flip tests for both AES-256 and AES+. Both distributions are near-Gaussian, centered at the ideal value of 64 bits (50%), confirming that CDS does not disturb the diffusion properties of the underlying cipher.

Table II summarizes the statistics.

AES$^+$ achieves $\mu = 64.010$ bits (50.008%), marginally closer to the ideal than AES-256 ($\mu = 63.941$, 49.954%). The slightly lower standard deviation (5.658 vs. 5.718) and the absence of counts in the 40–42 bin (present only in AES-256) confirm that the dynamic S-box and perturbed key schedule do not introduce any diffusion deficit.

Cipher	μ (bits)	σ	Min	Max
AES-256	63.941	5.718	42	88
AES$^+$	64.010	5.658	44	86
Ideal (Binomial)	64.000	5.657	—	

Table 2: Avalanche Statistics: 15,360 Tests (30 Keys)

Payload	AES-256	AES$^+$	Overhead
16 B	127.9 MB/s	78.2 MB/s	+63.7%
256 B	154.4 MB/s	85.1 MB/s	+81.5%
1024 B	144.2 MB/s	88.6 MB/s	+62.8%
4096 B	173.6 MB/s	94.2 MB/s	+84.2%
16384 B	165.6 MB/s	92.4 MB/s	+79.2%
65536 B	167.6 MB/s	88.7 MB/s	+89.1%

Table 3: CTR-Mode Throughput: C T-Table, Median of 4 Runs (GCC -O3, Intel i7-12700H, no AES-NI)

D. S-Box Diversity and Key Independence

Across 1,000 independent keys, the mean number of S-box positions differing between any two distinct-key instances is 254.97 out of 256 ($\sigma = 0.98$), confirming near-maximum diversity: any two AES$^+$ instances share on average only 1.03 identical substitution values. Under a 1-byte key change, the mean round-key difference is 56.4 bits per 128-bit round key across all 14 rounds, confirming that the PKS perturbation propagates the key change throughout the entire schedule.

E. Performance

AES$^+$ was implemented in C using a T-table optimization [9]: four 256-entry lookup tables fuse Sub Bytes and Mix Columns into four 32-bit XORs per column per round, with VSR applied as a separate byte-level permutation selecting which column contributes to each output position. The same T-table optimization (without VSR) was applied to a reference AES-256 implementation for a fair comparison. Both were compiled with GCC 13.3, -O3 -march=native, on an Intel Core i7-12700H; neither uses hardware AES-NI instructions.

Table III reports median throughput over four independent runs using a fixed 64 KB streaming buffer to eliminate cold start effects. AES$^+$ sustains 78–94 MB/s across all payload sizes versus 128–174 MB/s for standard AES-256, a throughput overhead of 63–89%.

The overhead is attributable entirely to VSR: standard AES256 uses fixed shift offsets that the compiler reduces to word rotations, whereas VSR's key-derived offsets require a per-round byte-level permutation that prevents this optimization. This is a deliberate design trade-off: for the UDSS steganographic use case, per-session keys mean each connection uses a different cipher variant, and the key setup cost (0.02 ms) is negligible against multi-KB payloads. Because CDS generates a key-dependent S-box each session, hardware AES-NI (which implements the fixed Rijndael S-box) cannot be used for Sub Bytes. Hardware acceleration for AES$^+$ would require an FPGA or ASIC implementing a runtime-programmable substitution table; this is identified as future work.

Test	p-value	Result
1. Frequency (Monobit)	0.7418	Pass
2. Block Frequency ($M = 128$)	0.2194	Pass
3. Runs	0.7708	Pass
4. Longest Run of Ones	0.3228	Pass
5. Binary Matrix Rank	0.3526	Pass
6. Spectral (DFT)	0.2672	Pass
7. Non-overlapping Template	0.6914	Pass
8. Overlapping Template	0.2840	Pass
9. Maurer's Universal	0.1286	Pass
10. Linear Complexity ($M = 500$)	0.0658†	Pass
11. Serial ($m = 3$)	0.8242	Pass
12. Approximate Entropy	0.6131	Pass

13a. Cumulative Sums (fwd)	0.7116	Pass
13b. Cumulative Sums (rev)	0.4288	Pass
14. Random Excursions (avg)	1.0000	Pass
15. Excursions Variant (avg)	1.0000	Pass
Overall	—	15/15 Pass

Table 4: NIST SP 800-22 Statistical Test Results (8×106 bits, $\alpha = 0.01$)

F. NIST SP 800-22 Randomness Tests

Table IV reports results from all 15 tests of the NIST SP 80022 Rev. 1a statistical test suite [2] applied to 8×10^6 bits (1 MB) of AES$^+$ CTR output generated under the reference key derived from seed string "Hello World" via SHA3-256, with random plaintexts. The significance level is $\alpha = 0.01$.

† Linear Complexity (M = 500, K = 6, N =16,000 blocks): χ^2 = 11.83, p = 0.0658. Mean block LC = 250.22 bits (ideal: 250.22); range [244,256]. The p-value is derived from the χ^2 distribution with $K/2$ degrees of freedom over the six T-value categories.

All 15 tests pass with p-values well above the 0.01 threshold. The Frequency test (p = 0.742) confirms near-equal bit proportions (3,999,534 ones out of 8,000,000 bits; 49.994%). The Linear Complexity test (p = 0.066) confirms that the mean block linear complexity of 250.22 bits matches the theoretical expectation for a random sequence of length M = 500, and that no block exhibits anomalously low complexity exploitable by LFSR-based attacks. Tests 14 and 15 report p = 1.000: for test 14, observed cycle counts over 8 states {−4,...,−1,1,...,4} matched expected values exactly; for test 15, the same held across 18 states {−9,...,−1,1,...,9}. These are valid outcomes and do not indicate a trivial pass. The complete output of AES$^+$ is statistically indistinguishable from a uniformly random sequence under all standard NIST criteria.

Method	**Ref. δ**		**NL**	**SAC**	**DP**	**LP**
AES-256 (Rijndael)	[1]	4	112	0.505	0.0156	0.1250
Logistic chaotic	[12]	8	104	0.484	0.0313	0.1875
PW-linear chaotic	[13]	8	108	0.504	0.0313	0.1328
2D compound chaotic	[14]	4	106	0.505	0.0156	0.1250
Lorenz-Henon chaotic	[15]	6	107	0.499	0.0234	0.1406
Hyper-chaotic+alg.	[16]	4	112	0.501	0.0156	0.1250
Genetic algorithm	[17]	4	112	0.502	0.0156	0.1250
CB-SBox (FPGA)	[18]	4	112	0.510	0.0156	0.1250
Alt. affine S-box	[19]	4	112	0.505	0.0156	0.1250
Key-dep. (Kazl.)	[20]	6	108	0.499	0.0234	0.1406
Key-dep. (Patil)	[21]	6	108	0.506	0.0234	0.1406
AES$^+$	Proposed	4	112	0.505	0.0156	0.1250

Table 5: S-Box Metric Comparison Against Related Works

Method	**Key-dep.**	**$\delta = 4$ proven**	**Gen. time**
AES-256 Rijn- dael [1]	No	Yes	N/A
Logistic chaotic [12]	Yes	No	15 ms
PW-linear chaotic [13]	No	No	>100 ms
2D compound [14]	No	No	<1 ms
Lorenz-Henon [15]	No	No	<1 ms
Hyper-chaotic [16]	Yes	No	<1 ms
Genetic alg. [17]	No	No	>1 s
CB-SBox (FPGA) [18]	No	Yes	<1 ms
Alt. affine [19]	No	Yes	<1 ms
Key-dep. Kazl. [20]	Yes	No	<1 ms

Key-dep. Patil [21]	Yes	No	<1 ms
AES$^+$ (Proposed)	Yes	Yes	0.475 ms

Table 6: Design Feature Comparison Against Related Works

7. Comparison with Related Work

Table V compares AES$^+$ against eleven representative S-box proposals across six criteria. Table VI compares construction approach and design properties. Together they position the unique contribution of AES$^+$.

A. Discussion

- **Differential Uniformity:** *Five of eleven methods achieve* $\delta = 4$. *Of these, only the CB-SBox FPGA construction [18] and the improved AES affine S-box [19] provide a formal proof—but neither is key-dependent. The three chaos-based methods reporting* $\delta = 4$ *(2D compound [14], hyper-chaotic [16], and genetic algorithm [17]) achieve this empirically on their specific generated instances, not for every possible key*.
- **Nonlinearity:** Six of eleven methods achieve the optimal NL = 112. The logistic chaotic map reports only NL = 104, indicating weaker linear attack resistance. AES$^+$ achieves NL = 112 provably for all keys [12].
- **The Key Gap:** *Proven* $\delta = 4$ *with Key-Dependence*: Table VI reveals the critical gap. The *Key-dep*. and $\delta = 4$ *proven* columns are never simultaneously "Yes" in any prior work. Key-dependent methods cannot prove $\delta = 4$; formally proven methods are session-independent. AES$^+$ is the first to occupy this intersection, enabled by the affine equivalence theorem which transforms a fixed proven S-box into a key-parametrized family of proven S-boxes.
- **Key Schedule and Shift Rows:** The most closely related key schedule work is the line of research on permutation-based AES key schedules [22]. Derbez et al. used MILP tools at SAC 2018 to find permutation-based schedules maximizing the number of active S-boxes in the related-key model [23]. Boura et al extended this at ACNS 2024, finding schedules that resist full round boomerang attacks while maintaining AES-NI compatibility. Both approaches seek better permutations within the same algebraic framework as the original AES schedule. PKS takes a fundamentally different approach: rather than optimizing permutation structure, it injects SHA3-256 perturbations that break the algebraic framework entirely, achieving computational independence of round keys under the random oracle model at the cost of AES-NI compatibility. Variable Shift Rows offsets appear in lightweight cipher design (e.g., GIFT) but have not been applied in AES enhancement literature. Nitaj et al. proposed enhanced S-boxes with maximal periodicity; their construction is static and addresses neither key dependence nor key schedule structure [24].

8. Conclusion

This paper presented AES$^+$, the first AES-256 variant to simultaneously provide: a formally proven key-dependent S-box ($\delta = 4$, NL = 112 for every instance, as guaranteed by the affine equivalence construction of Theorem 1); a perturbed key schedule with a formal sub-key independence proof under the random oracle model (Theorem 2); and key-derived variable Shift Rows. A C T-table implementation achieves 78–94 MB/s (128–174 MB/s for standard AES-256 under identical conditions) — an overhead of 63–89% attributable entirely to the VSR permutation step, without hardware AES-NI acceleration.

The avalanche histogram over 15,360 test pairs confirms near Gaussian diffusion ($\mu = 64.010$ bits, $\sigma = 5.658$), indistinguishable from standard AES-256. All 15 NIST SP 800-22 statistical tests pass on 8×10^6 bits of output, confirming statistical indistinguishability from uniform random. Comparison against eleven state-of-the-art S-box proposals confirms AES$^+$ uniquely occupies the intersection of key-dependence, formal $\delta = 4$ proof, and practical software deployability.

Future work: MILP-based formal differential trail bounds; bitsliced side-channel-resistant implementation; FPGA synthesis with runtime-programmable substitution table for hardwarelevel throughput; and extension to GF(216).

References

1. NIST. (2001). "Advanced Encryption Standard (AES)." FIPS Pub. 197, Nov.
2. Rukhin, A. (2010). "A statistical test suite for random and pseudorandom number generators for cryptographic applications." NIST Special Publication 800-22 Rev. 1a, Apr.
3. Ramana, K. V. (2015). "A novel technique for secure data transmission using distributed steganography." *IJERT*, vol. 4, no. 2.
4. Biham, E., & Shamir, A. (1991). Differential cryptanalysis of DES-like cryptosystems. Journal of CRYPTOLOGY, 4(1), 3-72.
5. Matsui, M. (1993). Linear cryptanalysis method for DES cipher. In Workshop on the Theory and Application of of Cryptographic Techniques (pp. 386-397). Berlin, Heidelberg: Springer Berlin Heidelberg.
6. Biryukov, A., & Khovratovich, D. (2009). Related-key cryptanalysis of the full AES-192 and AES-256. In International conference on the theory and application of cryptology and information security (pp. 1-18). Berlin, Heidelberg: Springer Berlin Heidelberg.
7. Leurent, G., & Pernot, C. (2021). New representations of the AES key schedule. In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 54-84). Cham: Springer International Publishing.
8. Courtois, N. T., & Pieprzyk, J. (2002). Cryptanalysis of block ciphers with over defined systems of equations. In International conference on the theory and application of cryptology and information security (pp. 267-287). Berlin, Heidelberg: Springer Berlin Heidelberg.
9. Daemen, J., Rijmen, V. (2002). *The Design of Rijndael*.

Springer.
10. Carlet, C., Crama, Y., & Hammer, P. L. (2010). Boolean Functions for Cryptography and Error-Correcting Codes.
11. Bernstein, D. J. (2005). Cache-timing attacks on AES.
12. Ejaz, A., Shoukat, I. A., Iqbal, U., Rauf, A., & Kanwal, A. (2021). A secure key dependent dynamic substitution method for symmetric cryptosystems. PeerJ Computer Science, 7, e587.
13. N. A. M. Nizam Chew and I. S. Ismail, "A novel systematic byte substitution method using piece-wise-linear chaotic map," *PeerJ Comput. Sci*., vol. 8, e940, 2022.
14. Yang, C., Wei, X., & Wang, C. (2021). S-box design based on 2D multiple collapse chaotic map and their application in image encryption. Entropy, 23(10), 1312.
15. Abdulrazaq, Z. A., Ayoub, H. G., & Zaidan, H. (2025). Synergistic construction of High-Performance S-Boxes based on chaotic systems: a paradigm shift in cryptographic security design. Chaos and Fractals, 2(2), 43-49.
16. Si, Y., Liu, H., & Zhao, M. (2023). Constructing keyed strong S-Box with higher nonlinearity based on 2D hyper chaotic map and algebraic operation. Integration, 88, 269-277.
17. Wang, Y., Zhang, Z., Zhang, L. Y., Feng, J., Gao, J., & Lei, P. (2020). A genetic algorithm for constructing bijective substitution boxes with high nonlinearity. Information Sciences, 523, 152-166.
18. Dutra e Silva Junior, É. C., Cruz, C. A. D. M., Saraiva, I. A. L., Santos, F. G., dos Santos Junior, C. R. P., Indrusiak, L. S., ... & Glesner, M. (2025). Chaos-Based S-Boxes as a Source of Confusion in Cryptographic Primitives. Electronics, 14(11), 2198.
19. Nitaj, A., Susilo, W., & Tonien, J. (2020). A new improved AES S-box with enhanced properties. In Australasian Conference on Information Security and Privacy (pp. 125-141). Cham: Springer International Publishing. Kazlauskas, K., & Kazlauskas, J. (2009). Key-dependent S-box generation in AES block cipher system. Informatica, 20(1), 23-34.
20. Patil, P., Karoshi, A., Marje, A., & Desai, V. (2023). Enhancing S-Box Nonlinearity in AES for Improved Security Using Key-Dependent Dynamic S-Box. In Proceedings of Fourth International Conference on Communication, Computing and Electronics Systems: ICCCES 2022 (pp. 91-102). Singapore: Springer Nature Singapore.
21. Derbez, P., Fouque, P. A., Jean, J., & Lambin, B. (2018). Variants of the AES key schedule for better truncated differential bounds. In International Conference on Selected Areas in Cryptography (pp. 27-49). Cham: Springer International Publishing.
22. Boura, C., Derbez, P., & Funk, M. (2024). Alternative key schedules for the AES. In International Conference on Applied Cryptography and Network Security (pp. 485-506). Cham: Springer Nature Switzerland.
23. Banik, S. (2017). "GIFT: A small present "in *Proc*. CHES. LNCS 10529, pp. 321–345.
24. Nitaj, A., Susilo, W., & Tonien, J. (2024). Enhanced S-boxes for the Advanced Encryption Standard with maximal periodicity and better avalanche property. Computer Standards & Interfaces, 87, 103769.