JCTA Journal of Computing Theories and Applications ISSN:3024-9104

Research Article

A Side-Channel-Aware Cryptographic Framework for Secure Interactive, Embedded, IoT, and Edge Communication Systems

Walid W. Souror [1,*], Mohamed Fouad [2,3], Fahmi Khalif [3,4], and Ali E. Takieldeen [5]

[1] Department of Electronics and Communications Engineering, Faculty of Engineering, Zagazig University, Zagazig 44519, Sharkia, Egypt; e-mail : w.wageeh22@eng.zu.edu.eg
[2] Higher Institute of Technology in Beheira, Abu El Matamir, Beheira Governorate, Egypt; e-mail : mfouad@zu.edu.eg
[3] Department of Electronics and Communications Engineering, Faculty of Engineering, Mansoura University, Mansoura 35516, Dakahlia, Egypt; e-mail : fahmikhalifa@mans.edu.eg
[4] Department of Electrical and Computer Engineering, Morgan State University, Baltimore, MD 21251, United States
[5] Department of Cyber Security, Faculty of Artificial Intelligence, Delta University for Science and Technology, Gamasa 35712, Dakahlia, Egypt; e-mail : a_takieldeen@deltauniv.edu.eg
* Corresponding Author : Walid W. Souror

Abstract: Secure interactive, embedded, and edge communication systems are often deployed in physically exposed environments where algorithmically secure ciphers may exhibit implementation-specific leakage. This paper presents a simulation-based evaluation of a configurable hybrid AES-Blowfish framework comprising AES-Hybridization, Combined Blowfish Trilogy, and cascaded Blowfish-to-AES modes. The AES-Hybridization path combines AES-256-CBC, Argon2id-derived whitening material, HMAC-based integrity binding, plaintext masking, and modeled randomized hiding activity. The hiding activity is represented solely within the leakage simulation and does not modify the plaintext or ciphertext. The Blowfish path employs session-dependent P-array randomization, dynamic S-box initialization, and a three-stage Feistel-like structure. The framework is evaluated using representative IoT/edge workload proxies, component-level ablation studies, modeled leakage assessment, non-ideal leakage scenarios, parameter-sensitivity analysis, and a software-level performance model. Compared with the simulated baseline configurations, the proposed modes reduce modeled TVLA, CPA, and DPA distinguishability, with AES-Hybridization providing the most balanced security-performance trade-off and the cascaded mode achieving the lowest modeled distinguishability at the highest computational cost. All findings are derived exclusively from simulation; no validation using physical power traces, electromagnetic traces, FPGA implementations, or microcontroller platforms is claimed.

Keywords: AES; Blowfish; CPA; Hybrid cryptography; IoT cryptography; Side-channel security; TVLA; Simulation-based leakage assessment.

Received: June, 20th 2026
Revised: July, 16th 2026
Accepted: July, 17th 2026
Published: July 26th 2026

1. Introduction

Cryptographic protection is essential for modern communication systems. However, the mathematical security of a cryptographic primitive does not necessarily guarantee the security of its physical or software implementation. Interactive systems, intelligent edge devices, wearables, smart gateways, embedded human-machine interfaces, and IoT communication nodes often operate in environments where an adversary can observe timing, power consumption, electromagnetic emanations, memory behavior, or other implementation-dependent signals. Consequently, side-channel analysis is directly relevant to the security of interactive and applied communication technologies [1]–[4].

AES remains one of the most widely used symmetric-key primitives, while Blowfish and related designs continue to be adopted in lightweight, image-encryption, and comparative cryptographic studies [5]–[9]. Nevertheless, conventional implementations may expose stable

intermediate computations, S-box activity, key-schedule behavior, or mode-dependent data patterns. Accordingly, countermeasures such as masking, hiding, shuffling, key whitening, constant-time programming, dynamic substitution, and randomized internal states have been extensively investigated [1]–[4], [10]–[17].

The present work investigates a configurable cryptographic framework for secure interactive and edge communication systems while maintaining a clearly defined simulation-based evidence boundary. The framework is not intended to replace hardware-validated side-channel countermeasures. Instead, it is designed as a programmable cryptographic module whose operating mode can be selected according to the threat level, latency requirements, workload size, and resource constraints. Representative application scenarios include smart sensing interfaces, medical and wearable communication devices, mobile edge nodes, industrial human-machine interfaces (HMIs), IoT gateways, and smart-grid communication environments [5].

Rather than claiming novelty for the individual cryptographic primitives or side-channel countermeasure families, this work contributes a configurable framework that coordinates two independently hardened cipher paths together with an optional cascaded mode. The proposed architecture comprises an AES-Hybridization path that combines AES-256-CBC, Argon2id-derived whitening material, HMAC-based integrity binding, plaintext masking, and modeled randomized hiding activity; a Combined Blowfish Trilogy path that employs session-dependent P-array randomization, dynamic S-box initialization, and a three-stage Feistel-like structure; and a sequential Blowfish-to-AES mode for higher-risk communication scenarios. The evaluation separately examines ciphertext randomness, modeled leakage characteristics, and performance overhead, thereby ensuring that each result is interpreted within its appropriate evidence boundary. The main contributions of this work are summarized as follows:

- A configurable hybrid cryptographic framework that integrates AES-Hybridization, Combined Blowfish Trilogy, and an optional cascaded Blowfish-to-AES mode to support different security and computational requirements in interactive, embedded, IoT, and edge communication systems.
- A unified simulation-based evaluation framework that systematically assesses the proposed operating modes through modeled leakage analysis, component-level ablation studies, parameter-sensitivity analysis, representative IoT/edge workload proxies, and software-level performance estimation within a clearly defined simulation boundary.
- A comparative analysis of the proposed operating modes, highlighting their security-performance trade-offs under representative deployment scenarios while explicitly limiting all conclusions to simulation-derived evidence.

The proposed framework is evaluated exclusively through simulation; therefore, all results should be interpreted within this evidence boundary. The remainder of this paper is organized as follows. Section 2 reviews related studies on side-channel attacks, cryptographic countermeasures, and hybrid encryption frameworks. Section 3 presents the proposed cryptographic framework and its operating modes. Section 4 describes the experimental setup, simulation environment, and evaluation methodology. Section 5 presents and discusses the experimental results, including security, leakage, performance, scalability, and comparative analyses. Finally, Section 6 concludes the paper and outlines directions for future work.

2. Related Work

Side-channel security research has evolved from the statistical evaluation of cryptographic outputs to implementation-level leakage analysis. Recent studies have investigated AES leakage datasets, modern hardware attack surfaces, provable side-channel security, hybrid post-quantum monitoring, Argon2 adoption, FPGA-oriented AES protection, synthesizable AES-256 signature attenuation, and data augmentation for protected AES implementations [1]–[4], [12]–[15]. Collectively, these studies demonstrate that practical cryptographic security should be evaluated with respect to a specific implementation and attack model rather than inferred solely from the mathematical properties of the underlying cipher.

Masked AES implementations randomize sensitive intermediate values before operations such as S-box evaluation, thereby reducing the statistical dependence between physical leakage and secret data. Although effective, these countermeasures generally introduce additional randomness requirements, implementation complexity, and computational overhead [3], [13]–[15]. Hardware-oriented approaches, including FPGA-based protection and

signature attenuation, provide stronger implementation-level evidence; however, their effectiveness remains platform-specific and requires validation using physical measurements [13], [14].

Research on Blowfish has explored dynamic substitution techniques, modified internal lookup tables, and generalized Feistel structures. Because Blowfish derives its security-critical P-arrays and S-boxes from the encryption key, session-dependent diversification of these structures can alter memory-access behavior and diffusion characteristics. However, these techniques have rarely been integrated with AES-oriented side-channel countermeasures or evaluated comparatively under a common simulation protocol [6]–[9], [11]. Despite these advances, a gap remains in the availability of a configurable cryptographic framework that systematically compares AES-based and Blowfish-based hardening strategies under a common simulation protocol while explicitly distinguishing ciphertext randomness, modeled leakage behavior, and implementation-level security.

Accordingly, the proposed framework is positioned as a complementary research platform rather than a replacement for formal masking schemes, threshold implementations, hardware-level countermeasures, standardized lightweight authenticated encryption, or previously reported hybrid and cascaded cryptographic constructions [18], [19]. Those approaches provide stronger theoretical guarantees or implementation-specific assurance within their respective security models. In contrast, the present work addresses the need for a transparent, configurable, and simulation-based evaluation framework that systematically quantifies the contributions of individual security components, analyzes workload scalability and non-ideal leakage behavior, and characterizes the associated security-performance trade-offs. By explicitly limiting all conclusions to simulation-derived evidence, the framework establishes a reproducible foundation for future implementation and hardware-validation studies.

3. Proposed Method

3.1 Threat Model and Design Objectives

The proposed framework assumes an adversary capable of observing implementation-dependent leakage, including timing behavior and modeled power or electromagnetic signals, from software-based or embedded deployment environments. The adversary is not assumed to compromise the mathematical security of AES or Blowfish; instead, the objective is to exploit statistical correlations between secret-dependent intermediate states and observable leakage. Accordingly, the design objectives are to reduce stable key-dependent regularities, provide selectable protection modes, support deployment in interactive and edge communication systems, and maintain clearly defined evidence boundaries for simulation-based evaluation.

The adopted threat model explicitly distinguishes modeled leakage from physical leakage. The adversary is assumed to exploit correlations represented in simulated leakage traces. These traces are not intended to reproduce oscilloscope-acquired power measurements, electromagnetic measurements, microarchitectural leakage, or compiler-dependent execution behavior. Consequently, all security-related observations presented in this work should be interpreted within the defined simulation boundary rather than as evidence of hardware-validated side-channel resistance.

3.2. Framework Architecture and Operating Modes

Figure 1 presents the overall workflow of the proposed framework. Representative IoT/edge workload proxies are first processed by a mode-selection layer that selects the operating mode according to the threat level, latency requirements, workload size, and available computational resources. The AES-Hybridization path targets deployments requiring standardized AES-based protection with moderate computational overhead. The Combined Blowfish Trilogy path emphasizes session-dependent internal-state diversification, whereas the cascaded Blowfish-to-AES mode is intended for higher-risk communication scenarios in which additional latency, memory consumption, and setup cost are acceptable.

Rather than introducing new cryptographic primitives, the proposed framework integrates established cryptographic mechanisms into a configurable side-channel-aware architecture. AES, Blowfish, HMAC, Argon2id, CBC chaining, masking, whitening, dynamic substitution, and cascaded encryption are well-established techniques. Their contribution in this

work lies in their coordinated integration within a unified framework whose operating mode can be selected according to communication requirements and systematically evaluated under a common simulation protocol.

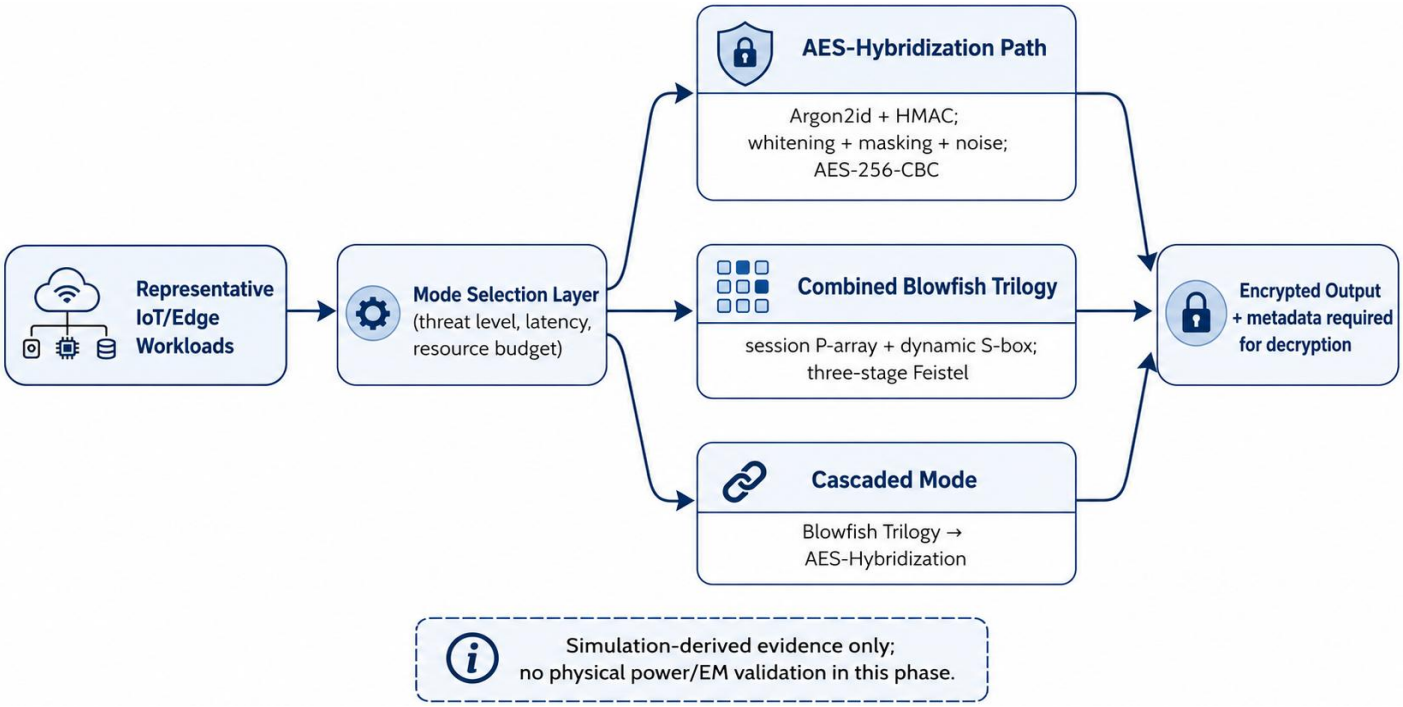

Figure 1. Overall architecture of the proposed hybrid AES-Blowfish framework.

The proposed framework is not intended to replace formally verified masking schemes, threshold implementations, or platform-specific hardware countermeasures [20], [21]. Instead, it provides a unified simulation-based evaluation framework that combines AES hardening, session-dependent Blowfish-state diversification, and an optional cascaded encryption mode. This common evaluation protocol supports consistent ablation studies, modeled leakage assessment, scalability analysis, and performance evaluation while explicitly distinguishing ciphertext randomness, modeled leakage behavior, and implementation-level security. Accordingly, the proposed framework should be interpreted as a candidate architecture for future hardware validation rather than as a hardware-validated side-channel-resistant implementation.

3.3. AES-Hybridization Path

The AES-Hybridization path is formulated as a block-level cryptographic transformation that combines AES-256-CBC with input randomization, key-dependent state diversification, integrity-bound metadata handling, and a modeled hiding layer. The mathematical formulation presented below defines the cryptographic transformation evaluated in Section 4. The hiding layer is applied only to simulated implementation activity rather than to the plaintext or ciphertext and therefore does not appear in the encryption or decryption equations. Accordingly, the proposed path is not presented as a formally proven masking scheme or a hardware-validated side-channel countermeasure.

Let P_i denote the i-th plaintext block, IV the initialization vector, K_{AES} the AES-256 encryption key, K_{pre} the pre-whitening key, K_{post} the post-whitening key, R_i the random masking block, and C_i the i-th ciphertext block. Whitening and authentication-related material are first derived from a master secret MS and a salt s using Argon2id with memory cost m, iteration count t, parallelism p, and output length L:

$$K_D = \text{Argon2id}(MS, s; m, t, p, L) \tag{1}$$

The derived material is then divided into independent subkeys to ensure that the same keying material is not simultaneously reused for encryption, whitening, and authentication:

$$(K_{AES}, K_{pre}, K_{post}, K_{HMAC}) = \text{Split}(K_D) \tag{2}$$

HMAC is employed to provide integrity protection for the whitening material and associated encryption metadata. This construction is not intended to represent a formal authenticated-encryption proof. Instead, the whitening keys and ciphertext-related metadata are cryptographically bound to an authentication tag, enabling the receiver to detect accidental or adversarial modifications:

$$\tau_K = \text{HMAC}_{K_{HMAC}}(K_{pre} \| K_{post} \| context) \tag{3}$$

Padding and masking are applied on a block-by-block basis before CBC chaining. The masked plaintext block is defined as:

$$\tilde{P}_i = Pad(P_i) \oplus R_i \tag{4}$$

The masked block is subsequently combined with the previous ciphertext block and the pre-whitening key to produce the CBC input. For the first block, C_{i-1} is replaced with the initialization vector:

$$X_i = \tilde{P}_i \oplus C_{i-1} \oplus K_{pre}, \quad C_o = IV \tag{5}$$

AES encryption is then performed as

$$Y_i = E_{K_{AES}}(X_i) \tag{6}$$

Finally, post-whitening produces the transmitted ciphertext block:

$$C_i = Y_i \oplus K_{post} \tag{7}$$

The ciphertext-level authentication tag is defined as:

$$\tau_C = HMAC_{K_{HMAC}}(IV \parallel C_1 \parallel C_2 \parallel \cdots \parallel C_n \parallel metadata) \tag{8}$$

By modifying the relationships among the plaintext, intermediate states, and transmitted ciphertext, the proposed construction alters the values presented to the modeled leakage functions. Practical deployment would additionally require secure management of initialization vectors, salts, masking information or deterministic mask-regeneration inputs, refresh policies, authentication tags, and session metadata. The present evaluation assumes that the required ciphertext and associated metadata are available for correct decryption and reproducible simulation but does not validate metadata management within an operational communication system.

Within this work, the term noise insertion refers to randomized, nonfunctional hiding activity, such as dummy-operation windows or additive activity in the modeled leakage trace, rather than numerical noise added to the plaintext or ciphertext. Consequently, this hiding mechanism requires neither decryption reversal nor additional recovery metadata. It should also be distinguished from the background leakage noise, timing jitter, signal drift, and memory-access disturbances introduced later as evaluation stressors across all operating modes.

3.4. Combined Blowfish Trilogy Path

The Combined Blowfish Trilogy path is formulated as a session-dependent Feistel-based construction. Its primary design objective is to reduce stable internal-state regularity by refreshing the P-array and S-box structures using a session seed together with a cryptographically secure pseudorandom generator. Similar to the AES-Hybridization path, this construction is intended for comparative simulation within the proposed framework rather than as a formally analyzed lightweight cipher.

Let K_{BF} denote the Blowfish master key, S_j the j-th session seed, and $G(\cdot)$ the state-generation function that produces the session-dependent P-array and S-box structures. The session-specific Blowfish state is defined as:

$$State_j = G(K_{BF}, S_j) = \{PArray_j, SBox_j\} \tag{9}$$

Each 64-bit input block B_l is divided into left and right halves:

$$B_l = L_l \parallel R_l \tag{10}$$

The round function is modeled using a simplified Blowfish-like F-function with four S-box lookups derived from the session-specific state. If R_l is represented by the four bytes a, b, c, and d, then

$$F_j(R_l) = \left(\left(S_{j,1}[a] + S_{j,2}[b]\right) \oplus S_{j,3}[c]\right) + S_{j,4}[d] \tag{11}$$

The trilogy path is implemented as a variation of a Feistel network. Three consecutive Blowfish-like transformations are applied using the same session-specific state generated for the current encryption session. The corresponding block transformation is expressed as:

$$B_l = BF\left(B_0; State_j\right), \qquad B_2 = BF\left(B_1; State_j\right), \qquad C_{BF} = BF\left(B_2; State_j\right) \tag{12}$$

The underlying rationale is that session-to-session variation in the P-array and S-box structures modifies the modeled leakage characteristics associated with table-access operations, thereby reducing the stability measured by CPA- and DPA-based distinguishability metrics. This diversification introduces additional setup overhead because session-state generation precedes encryption. Consequently, the Combined Blowfish Trilogy path should not be interpreted as an intrinsically lightweight construction but rather as an alternative operating mode for comparative security-performance evaluation within the proposed framework.

3.5. Cascaded Blowfish-to-AES Mode

The cascaded operating mode sequentially applies the Combined Blowfish Trilogy path followed by the AES-Hybridization path. This mode is intended for higher-risk communication scenarios in which additional computational overhead, memory consumption, and setup cost are acceptable in exchange for increased cryptographic diversification. The cascaded encryption process is represented as:

$$C_H = AES_{Hybrid}\left(BF_{Trilogy}\left(P; K_{BF}, S_j\right); K_{AES}, IV, R\right) \tag{13}$$

The corresponding decryption process follows the reverse order of encryption:

$$P = BF^{-1}_{Trilogy}\left(AES^{-1}_{Hybrid}(C_H); K_{BF}, S_j\right) \tag{14}$$

This operating mode evaluates whether combining the two independently diversified cryptographic paths further reduces modeled distinguishability compared with either operating mode individually. The cascaded configuration therefore represents the highest computational-cost mode considered in this study and should be interpreted as a comparative simulation scenario rather than as evidence of absolute security or hardware-validated side-channel resistance.

3.6. Parameterization and Implementation Boundaries

The proposed framework distinguishes cryptographic parameters from simulation parameters to preserve clear interpretation boundaries throughout the evaluation. Cryptographic parameters include key size, initialization vector length, authentication-tag length, salt length, mask-refresh policy, session-seed refresh policy, and dynamic table-generation policy. Simulation parameters include workload class, repetition count, background leakage-noise model, timing jitter, leakage observation point, signal drift, and statistical-analysis settings. The design-level hiding amplitude associated with the AES-Hybridization path is treated independently from the background leakage noise introduced to evaluate all operating modes under non-ideal conditions.

The metadata requirements also differ across the operating modes. The AES-Hybridization path requires an initialization vector, salt, masking information or deterministic mask-regeneration inputs, an HMAC authentication tag, and context metadata. The Combined Blowfish Trilogy path requires a session identifier together with the associated seed-handling and state-regeneration policies. In the cascaded operating mode, metadata from both cryptographic paths must be preserved in the appropriate order to ensure successful decryption.

The evaluation assumes that all required cryptographic parameters and metadata are available for reproducible simulation and correct decryption. However, the proposed framework does not validate metadata management, synchronization, storage, transmission, or lifecycle management within an operational communication system. These aspects remain implementation-dependent and are intentionally excluded from the simulation boundary adopted in this work.

4. Experimental Setup

4.1 Evaluation Scope, Configurations, and Workloads

The experimental evaluation is conducted entirely through simulation and encompasses configuration comparison, representative workload behavior, component-level ablation, modeled leakage analysis, parameter sensitivity, scalability, performance evaluation, and broader attack-model screening. No physical power traces, electromagnetic measurements, FPGA implementations, oscilloscope acquisitions, or microcontroller-based experiments were performed. Consequently, all reported results should be interpreted within the simulation boundary defined in Section 3 rather than as evidence of hardware-validated side-channel resistance.

The evaluation methodology follows three guiding principles. First, all directly simulated configurations are evaluated under consistent workload profiles, repetition counts, and modeled leakage assumptions. Second, every reported result is assigned to a clearly defined interpretation boundary, including ciphertext-randomness assessment, modeled leakage distinguishability, performance estimation, or analytical attack-model screening. Third, the raw simulation datasets, figure data, summary outputs, and Python scripts are provided as supplementary material to support reproducibility.

Figure 2 illustrates the overall workflow of the simulation-based evaluation methodology adopted in this study. The workflow begins with representative IoT/edge workload generation, followed by configuration selection, component-level ablation, modeled leakage generation, evaluation under non-ideal conditions, TVLA/CPA/DPA analysis, performance estimation, broader attack-model screening, and supplementary reproducibility support.

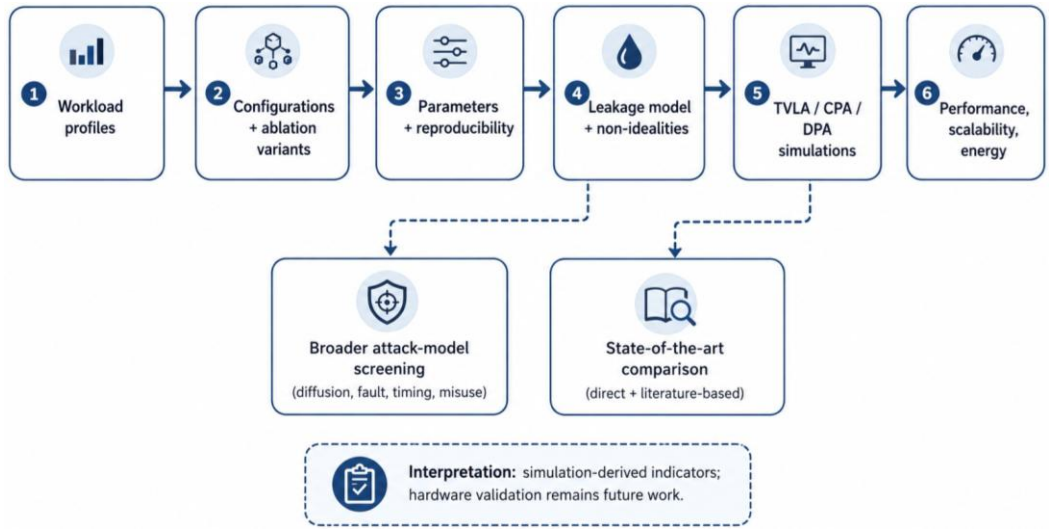

Figure 2. Overall workflow of the proposed simulation-based evaluation methodology.

Configurations C0–C5 were generated directly within a unified simulation workflow. Configurations C0, C1, and C3 serve as conventional baseline implementations, whereas C2, C4, and C5 represent the three proposed operating modes described in Section 3. Configurations C6–C8 are included only as abstract software-level reference proxies. Specifically, C6 and C7 provide AEAD-oriented comparison points under the same workload and performance model, while C8 represents an Ascon-like literature-oriented reference rather than an implementation of standardized Ascon. None of the reference proxies (C6–C8) represents

measured hardware performance, validated cryptographic-library benchmarks, or implementation-level side-channel evidence.

For the directly simulated configurations (C0–C5), identical workload profiles, repetition counts, modeled leakage assumptions, and statistical reporting procedures were employed to ensure fair comparison. Reference-proxy configurations (C6–C8) are interpreted separately and are not used to claim superiority over standardized implementations. Ciphertext randomness, TVLA, CPA, DPA, performance, and broader attack-model screening are therefore reported as independent evaluation dimensions. Accordingly, NIST SP 800-22 results are interpreted solely as indicators of ciphertext randomness rather than evidence of side-channel resistance, consistent with the documented limitations of statistical randomness test suites [10], [17], [22], [23]. Table 1 summarizes the evaluated configurations, their respective roles, nominal performance indicators, and reporting boundaries adopted throughout the simulation-based evaluation.

Table 1. Evaluated configurations, nominal summary values, and reporting boundaries.

Code	Configuration	Role	Throughput (Mbps, 1 MB, nominal)	Peak Memory (MB, nominal)	TVLA Max |t|	CPA Max Corr.	Setup (ms)	Reporting Boundary
C0	Original AES	Baseline	206.6	12.2	8.65	0.538	1.2	Direct simulation; no hardware data
C1	AES-CBC baseline	Baseline with CBC mode	195.0	12.8	8.10	0.500	1.4	Direct simulation; no hardware data
C2	AES-Hybridization	Proposed AES path	169.7	18.6	3.11	0.076	8.6	Direct simulation; no hardware data
C3	Original Blowfish	Baseline	184.1	10.4	7.53	0.386	1.0	Direct simulation; no hardware data
C4	Combined Blowfish Trilogy	Proposed Blowfish path	143.9	20.9	3.41	0.095	6.4	Direct simulation; no hardware data
C5	Cascaded Blowfish-to-AES	High-security cascaded mode	74.7	29.6	2.50	0.055	14.8	Direct simulation; no hardware data
C6	AES-GCM reference	AEAD reference proxy	188.0	14.5	7.90	0.460	2.2	Simulation proxy; no validated benchmark
C7	ChaCha20-Poly1305 reference	AEAD reference proxy	220.0	13.8	6.80	0.340	2.0	Simulation proxy; no validated benchmark
C8	Ascon-like AEAD reference	Ascon-like reference proxy	135.0	11.8	6.20	0.310	2.5	Ascon-like proxy; not standardized Ascon

Representative workload classes were constructed as structured simulation proxies because generic byte streams do not adequately represent communication patterns commonly encountered in IoT and edge-computing environments. The workload profiles include sensor telemetry, wearable-health packets, industrial HMI commands, edge-gateway batches, mixed edge streams, and large stress payloads. These workloads were generated synthetically from representative message structures and payload-size ranges rather than collected from deployed devices. The workload set is defined as:

$$W = \{w_{sensor}, w_{wearable}, w_{HMI}, w_{gateway}, w_{mixed}, w_{stress}\} \tag{15}$$

where each workload $w \in W$ is characterized by a payload-size distribution, a structural communication pattern, and a repetition count. Smaller telemetry workloads emphasize latency and per-packet overhead, whereas gateway batches and stress payloads primarily evaluate throughput and memory behavior. This design enables the evaluation to determine whether performance and security trends remain consistent across different communication profiles and payload sizes. Table 2 summarizes the representative workload profiles used throughout the simulation-based evaluation. Table 3 summarizes the parameter settings adopted throughout the simulation-based evaluation. The rationale and implementation of each parameter group are discussed in detail in Section 4.2.

Table 2. Representative IoT/edge workload profiles.

Code	Workload	Representative Scenario	Minimum Bytes	Maximum Bytes	Mean Bytes	Source / Boundary
W1	Sensor telemetry packet	IoT small-message proxy	32	128	67.2	Simulation-derived structured proxy; not captured from deployed devices
W2	Wearable health packet	Wearable/medical telemetry proxy	128	480	242.7	Simulation-derived structured proxy; not captured from deployed devices
W3	Industrial HMI command packet	Industrial HMI command proxy	64	240	138.1	Simulation-derived structured proxy; not captured from deployed devices
W4	Edge gateway batch	Gateway aggregated payload	5,248	227,904	57,109.3	Simulation-derived structured proxy; not captured from deployed devices
W5	Mixed edge stream	Mixed edge communication proxy	1,152	961,872	217,702.4	Simulation-derived structured proxy; not captured from deployed devices
W6	Stress edge payload	Large-payload stress proxy	10,485,760	10,485,760	10,485,760	Simulation-derived structured proxy; not captured from deployed devices

Table 3. Parameter settings used in the simulation-based evaluation.

Parameter	Levels	Purpose
Argon2id memory cost	16, 32, 64 MB	Sensitivity setting; simulation-derived setup overhead
Argon2id iterations	1, 2, 3	Sensitivity setting for key-derivation cost
Argon2id parallelism	1, 2	Sensitivity setting
Mask refresh policy	Per session, per packet, per block	Lower reuse generally reduces modeled leakage
AES-H hiding-activity amplitude	Low, medium, high	Design-level simulated dummy/additive activity; does not modify message data
Background leakage noise	Low, medium, high; Gaussian, uniform, mixed	Evaluation stressor applied in leakage scenarios
P-array refresh policy	Per session, per packet	Blowfish Trilogy dynamic-state sensitivity
S-box refresh policy	Per session, per batch	Dynamic table sensitivity
Session seed policy	Unique, reused stress scenario	Misuse/stress evaluation only

4.2. Parameter Settings and Reproducibility

Table 3 summarizes the parameter settings adopted throughout the simulation-based evaluation. The following discussion explains the rationale for each parameter group while distinguishing cryptographic parameters from simulation-specific evaluation settings. To support reproducibility and isolate the contribution of each design element, all parameter settings were explicitly specified. The AES-Hybridization path is parameterized by the AES key size, CBC operating mode, initialization-vector length, padding rule, mask-refresh policy, modeled hiding-activity amplitude, Argon2id configuration, and HMAC-tag handling. The Combined Blowfish Trilogy path is parameterized by the session-seed length, P-array refresh policy, S-box refresh policy, and the number of Feistel-like stages. Background leakage-noise amplitude and distribution, temporal jitter, baseline drift, and workload sampling are treated as evaluation parameters rather than cryptographic inputs. Noncryptographic simulation seeds are

fixed only to reproduce modeled disturbances and lookup-table generation, whereas cryptographic session variables are regarded as independent conceptual inputs. The complete parameter vector is defined as:

$$\Theta = \{\theta_{AES}, \theta_{Argon2id}, \theta_{HMAC}, \theta_{mask}, \theta_{noise}, \theta_{BF}, \theta_{sim}\} \tag{16}$$

where θ_{sim} denotes the simulation-only parameter group, including the repetition count, background leakage-noise level, temporal jitter magnitude, drift factor, and workload-sampling policy. These parameters do not represent implementation claims; instead, they define the reproducibility range of the simulation-based evaluation.

4.3. Ablation and Modeled Leakage Analysis Protocol

The ablation study quantifies the marginal contribution of each principal component within the proposed framework. For the AES-Hybridization path, components are progressively activated from the baseline configuration to the complete operating mode. The Combined Blowfish Trilogy path follows the same incremental strategy, while the hybrid evaluation compares the complete AES-Hybridization, Combined Blowfish Trilogy, and cascaded Blowfish-to-AES operating modes. The incremental contribution of each additional component is expressed as:

$$\Delta_S(q) = S(q) - S(q-1) \tag{17}$$

For leakage-oriented metrics in which lower values indicate improved security, such as TVLA |t| and CPA correlation, improvement is defined as a reduction in the corresponding metric. For performance-oriented metrics, including execution time and memory consumption, the same ablation sequence is used to quantify the additional computational cost introduced by each component. This evaluation strategy avoids treating the proposed framework as a black-box combination of existing techniques by explicitly identifying the contribution of each design element.

The baseline modeled leakage follows the Hamming-weight and Hamming-distance assumptions. For AES-based configurations, the first-round S-box output corresponding to a key hypothesis k is selected as the target intermediate state. For plaintext byte P_i, the modeled intermediate value is defined as:

$$V_{i,k} = SBox(P_i \oplus k) \tag{18}$$

The modeled leakage combines Hamming-weight and Hamming-distance terms with additive background disturbance. For AES-Hybridization configurations, a separate configuration-controlled hiding component represents the nonfunctional activity introduced in Section 3.3:

$$L_i(t) = \alpha_t\, HW\big(V_{i,k}\big) + \beta_t\, HD\big(V_{i,k}, V_{i-1,k}\big) + \eta_i(t) \tag{19}$$

To approximate non-ideal operating conditions, the leakage model further incorporates temporal jitter, trace misalignment, amplitude scaling, baseline drift, memory-access leakage, dummy-operation windows, and multipoint leakage windows:

$$L'_i(t) = g_i L_i(t - \delta_i) + d_i(t) + \mu_i(t) + \nu_i(t) \tag{20}$$

where g_i, δ_i, $d_i(t)$, $\mu_i(t)$, and $\nu_i(t)$ denote amplitude scaling, temporal misalignment, baseline drift, memory-access leakage, and background random noise, respectively. These evaluation stressors are applied independently of the design-level hiding activity introduced by the AES-Hybridization path to assess whether the observed simulation trends remain stable under less ideal operating conditions.

Fixed-versus-random Welch's t-tests were employed for TVLA following established side-channel assessment practice [24], [25]. The fixed and random trace groups are compared at each sampling point t using Equation (21).

A first-order leakage indication is recorded when the maximum absolute t-value exceeds the commonly adopted threshold of $|T| > 4.5$. Within this work, the threshold serves solely as a simulation-screening indicator and does not constitute evaluation under ISO/IEC 17825.

$$T(t) = \frac{\bar{x}_F(t) - \bar{x}_R(t)}{\sqrt{\frac{s_F^2(t)}{n_F} + \frac{s_R^2(t)}{n_R}}} \tag{21}$$

Correlation Power Analysis (CPA) correlates hypothetical leakage values with the simulated traces to evaluate key distinguishability [26], [27]. The maximum correlation for each key hypothesis k is computed using:

$$\rho_k(t) = corr\left(HW(V_{i,k}), L_i(t)\right) \tag{22}$$

The correct key hypothesis k^* is ranked according to the maximum absolute correlation value:

$$rank(k^*) = 1 + \left|\left\{k: \max_t |\rho_k(t)| > \max_t |\rho_{k^*}(t)|\right\}\right| \tag{23}$$

Differential Power Analysis (DPA) is evaluated by partitioning traces according to the predicted intermediate-bit values and computing the corresponding difference of means [27], [28]:

$$D_k(t) = mean(L_i(t) \mid b_i(k) = 1) - mean(L_i(t) \mid b_i(k) = 0) \tag{24}$$

Accordingly, CPA and DPA are interpreted solely as modeled distinguishability metrics within the adopted simulation framework.

4.4. Sensitivity, Scalability, and Performance Evaluation

Parameter sensitivity analysis evaluates whether the observed security and performance trends remain stable across the parameter settings summarized in Table 3. The evaluated factors include Argon2id memory and iteration settings, mask-refresh policy, modeled hiding-activity amplitude, background leakage noise, P-array refresh frequency, S-box update policy, and session-seed reuse conditions. Security-oriented and performance-oriented indicators are analyzed jointly to determine the relative influence of each parameter on the proposed framework. The relative sensitivity of metric Y with respect to parameter value θ_r is calculated as:

$$Sensitivity(\theta_r) = \frac{Y(\theta_r) - Y(\theta_{base})}{|Y(\theta_{base})|} \tag{25}$$

This metric quantifies the relative effect of each parameter on the evaluated security and performance indicators and identifies whether the observed trends remain consistent across different parameter settings.

Scalability is evaluated using payload sizes ranging from small telemetry packets to large edge-style batches (32 bytes to 10 MB). These payloads distinguish latency-sensitive communication from throughput-oriented processing, allowing the evaluation to determine whether computational overhead remains stable across representative IoT and edge communication scenarios. Throughput is calculated as:

$$Throughput = \frac{8N}{T_{enc}} \tag{26}$$

where N is the payload size in bytes and T_{enc} is the encryption time in seconds. The relative computational overhead is calculated with respect to the corresponding baseline:

$$Overhead = \frac{T_{mode} - T_{base}}{T_{base}} \times 100 \tag{27}$$

The deployment analysis considers three representative classes of IoT platforms: constrained devices, moderate embedded or wearable devices, and edge gateways. Deployment feasibility is evaluated using simulated latency, memory consumption, and estimated energy behavior.

The performance model distinguishes runtime cost from setup cost. Runtime cost includes packet-level encryption and decryption, whereas setup cost includes Argon2id key derivation, HMAC preparation, session-state generation, and dynamic P-array/S-box

initialization. This distinction is important because several operations are performed once per communication session rather than for every transmitted packet. The estimated packet-level energy consumption is approximated as:

$$E_{packet} = P_{CPU} \times T_{packet} \tag{28}$$

where P_{CPU} denotes the assumed processor power profile and T_{packet} is the simulated packet-processing time. The resulting energy estimates are intended solely for comparative simulation across configurations and should not be interpreted as hardware measurements.

4.5. Broader Attack-Model Screening and Statistical Analysis

Broader attack-model screening complements the ciphertext-randomness, modeled leakage, and performance evaluations by examining additional analytical security scenarios. The screening includes chosen-plaintext diffusion, chosen-ciphertext and integrity behavior, modeled fault propagation, timing-variance indicators, replay and session-reuse stress, IV and mask reuse, and implementation-oriented vulnerability analysis. These assessments provide complementary analytical evidence within the adopted simulation framework rather than formal security proofs. Chosen-plaintext diffusion is summarized using the ciphertext Hamming-distance ratio.

$$Avalanche(P, P') = \frac{HD\big(C(P), C(P')\big)}{|C|} \tag{29}$$

where a value approaching 0.5 indicates favorable bit-level diffusion in the simulated ciphertext. Fault-propagation screening introduces perturbations at selected input or intermediate-state locations and measures the resulting output-change ratio. Timing analysis evaluates the variance of simulated execution time and its correlation with input patterns. Together, these metrics provide comparative analytical screening of implementation-oriented behaviors within the simulation environment.

All stochastic analyses involving workload sampling, modeled leakage generation, disturbance generation, and performance estimation were repeated 30 times for each directly simulated configuration and applicable ablation variant. Deterministic configuration descriptions and analytical screening definitions were evaluated once because they do not depend on stochastic processes. Nominal reference-setting values are reported in Tables 1, 4, and 5, whereas repeated-run statistics are presented in Tables 6 and 7. Minor differences between nominal summaries and repeated-run averages therefore reflect the reporting basis rather than contradictory experiments.

Repeated experimental results are summarized using the mean, standard deviation, and 95% confidence interval (CI). For a metric with mean $\bar{x}$, sample standard deviation s, and repetition count n, the confidence interval is calculated as:

$$CI_{95} = x^{-} \pm 1.96 \frac{s}{\sqrt{n}} \tag{30}$$

5. Results and Discussion

This section presents the component ablations, non-ideal leakage scenarios, parameter-sensitivity findings, performance and scalability trade-offs, comparative security indicators, and broader security screening. Reductions in TVLA, CPA, and DPA are interpreted only as lower modeled distinguishability under the adopted simulation assumptions.

5.1. Ablation and Modeled Leakage Results

Tables 4 and 5 summarize the ablation results for the proposed AES-Hybridization, Blowfish Trilogy, and hybrid operating modes under the nominal reference setting. These experiments evaluate the incremental contribution of each protection component to the modeled leakage indicators and performance characteristics. Figure 3 provides a normalized visual comparison of the relative contributions of the evaluated configurations across the considered security and performance metrics.

Table 4. AES-Hybridization ablation results under the nominal reference setting.

Code	AES Variant	NIST (%)	TVLA max |t|	CPA Corr.	DPA DoM	Throughput (Mbps)	Memory (MB)	Setup (ms)
A0	Original AES	98.7	8.65	0.538	1.17	206.6	12.2	1.2
A1	AES-CBC only	99.0	8.10	0.500	1.08	195.0	12.8	1.4
A2	AES-CBC + pre-whitening	99.2	6.55	0.365	0.79	188.0	14.5	2.6
A3	AES-CBC + pre/post-whitening	99.4	5.72	0.285	0.63	181.0	15.3	3.4
A4	AES + whitening + masking	99.7	4.05	0.145	0.36	175.0	16.8	4.9
A5	AES + whitening + masking + modeled hiding activity	99.9	3.42	0.092	0.25	171.0	17.4	5.7
A6	Full AES-Hybridization	100.0	3.11	0.076	0.21	169.7	18.6	8.6

Table 5. Blowfish Trilogy and hybrid-mode ablation results under the nominal reference setting.

Family	Code	Variant	NIST (%)	TVLA max |t|	CPA Corr.	DPA DoM	Throughput (Mbps)	Memory (MB)	Setup (ms)
Blowfish ablation	B0	Original Blowfish	90.0	7.53	0.386	0.89	184.1	10.4	1.0
	B1	Blowfish + dynamic P-array	94.2	6.12	0.270	0.68	169.0	13.5	2.5
	B2	Blowfish + dynamic S-box	95.7	5.78	0.245	0.59	164.0	14.9	2.9
	B3	Blowfish + dynamic P-array + dynamic S-box	98.8	4.30	0.148	0.37	154.0	17.2	4.4
	B4	Two-stage Blowfish-like Feistel	99.4	3.88	0.124	0.31	150.0	19.0	5.1
	B5	Full Combined Blowfish Trilogy	100.0	3.41	0.095	0.25	143.9	20.9	6.4
Hybrid ablation	H1	Full AES-Hybridization	100.0	3.11	0.076	0.21	169.7	18.6	8.6
	H2	Full Combined Blowfish Trilogy	100.0	3.41	0.095	0.25	143.9	20.9	6.4
	H3	Cascaded Blowfish-to-AES	100.0	2.50	0.055	0.17	74.7	29.6	14.8

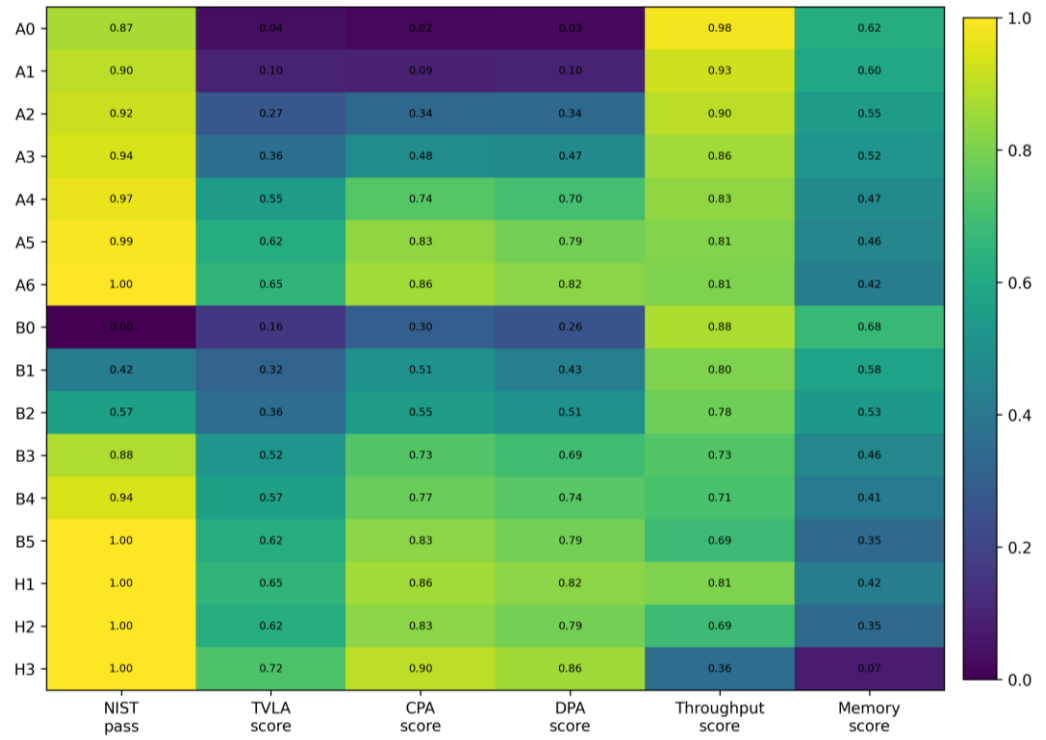

Figure 3. Normalized ablation-contribution heatmap of the evaluated configurations. All indicators are normalized for visualization, with larger values representing more favorable overall performance after metric-direction normalization.

The ablation results identify the contribution of each protection component. In the AES path, pre- and post-whitening, masking, and modeled hiding activity progressively reduce the simulated leakage indicators while increasing setup and runtime costs. In the Blowfish path, dynamic P-array and S-box behavior reduce modeled leakage stability. The cascaded mode yields the lowest nominal CPA and TVLA values but incurs the largest throughput penalty. Overall, the results indicate that the observed improvements arise from the cumulative contribution of multiple protection mechanisms rather than any single enhancement. The progressive reductions in TVLA, CPA, and DPA therefore support the design rationale of combining whitening, masking, dynamic-state diversification, and modeled hiding activity while illustrating the expected trade-off between reduced modeled leakage indicators and additional computational overhead.

5.2. Parameter Sensitivity and Non-Ideal Leakage

Table 6 summarizes the results obtained under non-ideal modeled leakage conditions. The evaluated scenarios introduce progressively more challenging disturbance conditions, including Gaussian noise, temporal misalignment, memory-access leakage, and combined perturbations, to assess the robustness of the proposed configurations within the adopted simulation framework.

Table 6. Summary of non-ideal modeled leakage scenarios (30-run means).

Scenario	Description	AES TVLA	AES-H TVLA	BF TVLA	BF-Trilogy TVLA	Cascaded TVLA	AES-H CPA	BF-Trilogy CPA	Cascaded CPA
Ideal modeled leakage	HW/HD + low additive noise	8.766	3.051	7.593	3.329	2.398	0.084	0.088	0.048
High noise	HW/HD + high Gaussian noise	7.058	2.529	6.189	2.810	2.085	0.074	0.093	0.051
Clock jitter	Temporal shifts/misalignment	8.067	2.923	6.949	3.158	2.274	0.068	0.082	0.044
Memory-influenced	Additional lookup/table-access leakage	9.868	3.510	8.701	3.907	2.861	0.094	0.117	0.074
Combined non-ideal	Noise + jitter + memory + baseline drift + amplitude variation	10.648	3.792	9.235	4.153	3.012	0.107	0.128	0.075

The baseline AES and Blowfish configurations remain more distinguishable than the proposed modes under modeled noise, jitter, memory-access leakage, and baseline drift. Table 6 reports scenario-specific 30-run means, accounting for the small differences from the nominal reference-setting values reported in Tables 4 and 5. These disturbances make the simulation less idealized but do not convert it into physical side-channel evidence. Despite the increasing disturbance levels, the relative performance ranking of the evaluated configurations remains largely consistent across all scenarios. This observation suggests that the proposed protection mechanisms maintain their comparative effectiveness under a range of modeled leakage conditions, although the results should be interpreted only within the assumptions of the adopted simulation framework.

5.3. Performance and Scalability Trade-Offs

Table 7 summarizes the performance estimates of the evaluated configurations for the 1 MB workload based on 30 repeated simulation runs. Figure 4 illustrates the scalability and performance trade-offs across different payload sizes, highlighting the relationship between computational overhead and the modeled leakage reductions achieved by the proposed operating modes.

The 30-run mean performance estimates identify AES-Hybridization as the most balanced proposed mode, while the Combined Blowfish Trilogy incurs greater dynamic-state overhead. The cascaded mode provides the greatest modeled leakage reduction but requires substantially higher estimated latency, memory, and energy consumption. Its values differ slightly from the rounded nominal screening values reported in Table 1 because the two tables use different reporting bases. These results demonstrate the expected trade-off between

computational efficiency and reduced modeled leakage indicators. Among the proposed configurations, AES-Hybridization provides the most balanced compromise between security-oriented improvements and estimated computational cost.

Table 7. Extended performance summary for the 1 MB workload (30-run means).

Code	Configuration	Enc (ms)	Dec (ms)	Throughput (Mbps)	Latency (ms)	CPU est. (%)	RAM (MB)	Energy (mJ)	Setup (ms)
C0	Original AES	40.783	42.119	205.688	40.783	44.63	12.09	17.129	1.2
C1	AES-CBC baseline	43.228	44.089	194.053	43.228	47.05	12.28	18.156	1.4
C2	AES-Hybridization	50.722	52.367	165.384	50.722	54.05	19.47	21.303	8.6
C3	Original Blowfish	45.715	45.903	183.496	45.715	49.47	10.82	19.201	1.0
C4	Combined Blowfish Trilogy	59.255	60.792	141.569	59.255	61.91	21.92	24.887	6.4
C5	Cascaded Blowfish-to-AES	114.517	119.035	73.252	114.517	96.00	30.43	48.097	14.8

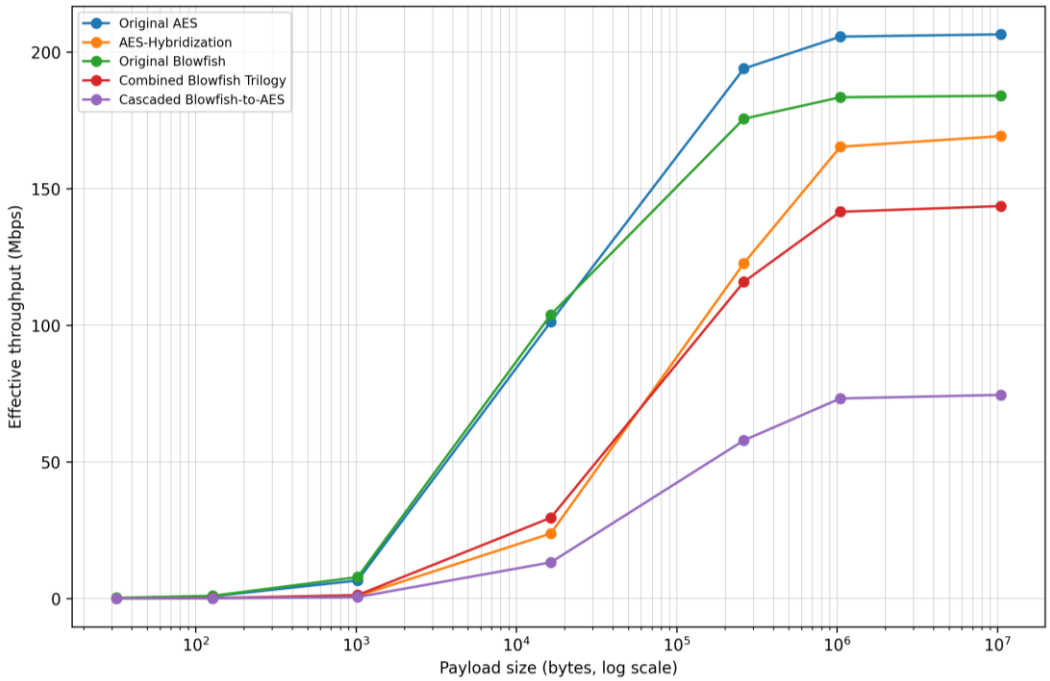

Figure 4. Simulation-derived scalability and performance trade-offs across payload sizes and evaluated cryptographic configurations.

5.4. Comparative Security Indicators

Figures 5–7 compare the principal security indicators obtained under the nominal reference setting, including the NIST pass rate, maximum simulated TVLA statistic, and maximum simulated CPA correlation. Together, these indicators provide complementary perspectives on ciphertext randomness and modeled leakage distinguishability within the adopted simulation framework.

Figure 5 shows the highest simulated pass rates for AES-Hybridization, Combined Blowfish Trilogy, and cascaded Blowfish-to-AES, whereas Original Blowfish achieves the lowest pass rate among the evaluated configurations. These results describe ciphertext-output randomness only and do not establish side-channel resistance. Figure 6 shows that the proposed modes remain below the commonly used first-order TVLA screening threshold of $|t|=4.5$ under the adopted leakage model, whereas Original AES, AES-CBC, and Original Blowfish exceed that threshold. This comparison should be interpreted as simulation-based screening rather than standards-based hardware validation.

Figure 7 shows lower maximum simulated CPA correlations for the proposed modes than for the baseline configurations. The cascaded Blowfish-to-AES mode produces the lowest modeled correlation among the evaluated configurations, but this result does not demonstrate physical key-recovery resistance. Because C6–C8 are abstract reference proxies, their

positions in Figure 7 are descriptive only and must not be interpreted as measured comparisons with production AES-GCM, ChaCha20-Poly1305, or standardized Ascon implementations. Taken together, the results presented in Figures 5–7 indicate that the proposed configurations improve ciphertext randomness while reducing modeled leakage distinguishability relative to the evaluated baselines. These observations should, however, be interpreted only within the adopted simulation assumptions and not as evidence of implementation-level side-channel resistance.

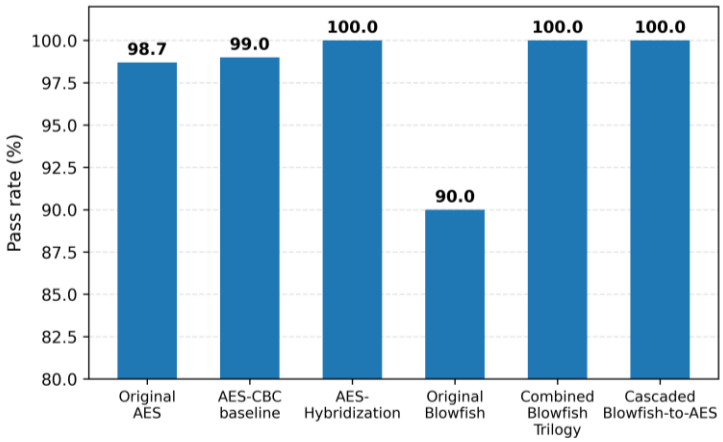

Figure 5. NIST pass-rate comparison across the evaluated configurations under the nominal reference setting.

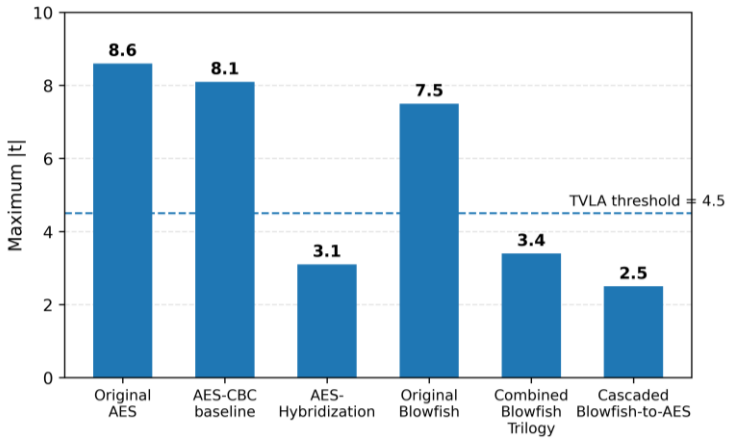

Figure 6. Maximum simulated TVLA |t| across the evaluated configurations under the nominal reference setting.

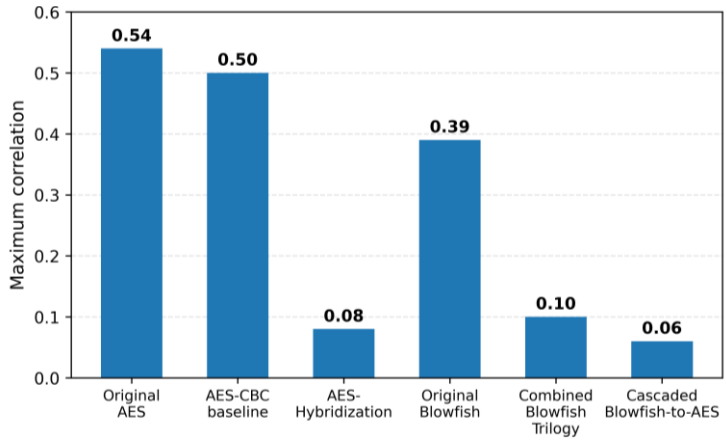

Figure 7. Maximum simulated CPA correlation across the evaluated configurations under the nominal reference setting.

5.5. Broader Attack-Model Screening and Security Interpretation

Table 8 summarizes the broader attack-model screening results using normalized heuristic scores. These indicators complement the CPA, DPA, and TVLA analyses by providing additional insight into implementation-oriented security considerations within the adopted simulation framework.

Table 8. Broader attack-model screening matrix (normalized heuristic scores; interpretation directions are defined in the text).

Code	Configuration	Avalanche (%)	Integrity Score	Fault-Propagation Score	Timing Risk	Reuse Risk	Misuse Risk	Implementation Risk
C0	Original AES	49.87	0.45	0.54	0.79	0.76	0.82	0.82
C2	AES-Hybridization	50.45	0.82	0.86	0.72	0.38	0.42	0.83
C4	Combined Blowfish Trilogy	50.59	0.72	0.84	0.66	0.40	0.44	0.89
C5	Cascaded Blowfish-to-AES	50.72	0.92	0.84	0.89	0.33	0.38	0.95

Table 8 presents a normalized heuristic screening matrix rather than a formal security proof, and its entries should not be interpreted as probabilities. Avalanche behavior is favorable when it approaches 50%, whereas higher values are favorable for the integrity and fault-propagation scores, and lower values are favorable for timing, reuse, misuse, and implementation risk. Under these definitions, the proposed modes improve integrity and fault-propagation screening while reducing reuse and misuse risk relative to Original AES. The cascaded mode, however, exhibits the highest timing and implementation-risk scores because of its greater complexity. Taken together, these heuristic indicators highlight the expected trade-off between enhanced protection mechanisms and increased implementation complexity, thereby complementing the leakage-oriented metrics presented in the previous subsections. The matrix therefore identifies implementation aspects that require further attention during deployment rather than establishing certified security.

Under a chosen-plaintext scenario, an adversary can submit structured or repeated plaintexts and observe the resulting ciphertexts. Within the adopted simulation framework, the proposed architecture addresses this scenario through diffusion, avalanche behavior, ciphertext Hamming-distance variation, workload-dependent output regularity, and the use of fresh IVs, masks, session seeds, and dynamic internal states to reduce deterministic regularities. These indicators, however, should be interpreted only as simulation-based screening and do not constitute a formal proof of chosen-plaintext security beyond the assumptions of the underlying primitives and operating modes.

AES-CBC alone does not provide authenticated encryption, which is important under chosen-ciphertext conditions. The AES-Hybridization path therefore incorporates HMAC-based integrity binding and requires successful tag verification before decryption and plaintext release. This design reduces exposure to unauthenticated ciphertext manipulation but is not presented as a formal proof of chosen-ciphertext security.

Timing resistance requires separate implementation-level assessment. The Python simulation estimates execution-time variation across workloads and configurations but cannot establish constant-time behavior. Branching, memory-access patterns, compiler optimization, cache behavior, table lookups, and platform-specific runtime effects all influence timing leakage [16]. The reported timing results should therefore be interpreted as variance screening rather than validation against timing attacks.

Fault-injection resistance must likewise be interpreted cautiously. The simulation models fault propagation across multiple stages of the cryptographic process and reports its resulting effects. HMAC verification may detect manipulated ciphertext or metadata before plaintext release, but it cannot guarantee detection of all internal computational or transient hardware faults. A comprehensive evaluation would require dedicated instruction-skip, clock- and voltage-glitch, electromagnetic-fault, and memory-corruption experiments in emulation and on physical platforms.

Session reuse and misuse remain important practical concerns. Reusing IVs, masks, session seeds, or dynamically generated P-array and S-box states can reduce diversification and reintroduce stable leakage or ciphertext regularities. Any practical implementation should therefore enforce strict freshness policies, robust state management, and appropriate misuse detection.

The simulation does not capture all implementation-specific leakage sources. Physical devices may exhibit additional leakage arising from power consumption, electromagnetic emanations, memory hierarchy, compiler transformations, clock instability, power-supply variation, and platform-specific hardware behavior. Future validation should therefore include public or newly collected real-trace datasets, including simultaneous power and electromagnetic measurements of AES implementations [29]. These screening results support the proposed framework as a candidate for future physical implementation and real-trace evaluation; they should not be interpreted as demonstrating deployment-level side-channel resistance.

5.6. Comparison with the State of the Art

Table 9 compares the proposed framework with representative state-of-the-art approaches discussed in the literature. The comparison emphasizes differences in design objectives, implementation assumptions, and evaluation scope rather than establishing superiority over existing techniques.

Table 9. Comparison with representative state-of-the-art approaches.

Approach	Category	Main Feature	Comparison Type	Hardware Validation	Main Interpretation
Standard AES-CBC	Conventional block-cipher baseline	No built-in SCA countermeasure	Algorithmic baseline	No	Low computational overhead but stable intermediate values
AES-GCM	Authenticated encryption baseline	Authenticated encryption with nonce-based AEAD	Reference comparison	No	Provides authentication but not physical SCA resistance
ChaCha20-Poly1305	Lightweight AEAD baseline	Software-oriented stream-cipher AEAD	Reference comparison	No	Efficient software implementation but not equivalent to SCA validation
Ascon Lightweight AEAD	NIST lightweight cryptography standard	Authenticated encryption for constrained devices	Literature-oriented comparison	Depends on implementation	Well suited for IoT but based on a different cryptographic primitive
AES Masking / Higher-Order Masking	Formal SCA countermeasure	Sensitive values split into randomized shares	State-of-the-art discussion	Implementation-specific	Strong theoretical protection with additional randomness and implementation overhead
Threshold Implementations	Hardware SCA countermeasure	Glitch-resistant shared hardware design	State-of-the-art discussion	Usually requires hardware testing	Strong hardware-oriented protection with implementation-specific complexity
Dynamic S-box / Modified Blowfish Variants	Dynamic substitution approaches	Diversified internal table behavior	Related-work comparison	Usually limited	Improved diversification but generally not integrated with AES-level protection
R-STELLAR / Protected AES Hardware	Hardware signature attenuation	Physical implementation-oriented protection	State-of-the-art discussion	Yes (target implementation)	Strong physical security evidence but reduced portability
Proposed AES-Hybridization	Proposed framework mode	Whitening, masking, modeled hiding activity, CBC, and integrity-bound key derivation	Simulation-derived evaluation	No	Moderate computational overhead with reduced modeled leakage indicators
Proposed Blowfish Trilogy	Proposed framework mode	Dynamic P-array, dynamic S-box, and three-stage Feistel diversification	Simulation-derived evaluation	No	Higher computational overhead than the baseline with reduced modeled leakage indicators
Proposed Cascaded Mode	Proposed framework mode	Blowfish Trilogy followed by AES-Hybridization	Simulation-derived evaluation	No	Greatest modeled leakage reduction with the highest computational cost

The results should be interpreted alongside recent advances in protected AES implementations, lightweight authenticated encryption, and physical side-channel analysis. Formal masking and threshold implementations reduce the statistical dependence between sensitive intermediate computations and observable leakage by splitting or transforming sensitive values [20], [21], [30], [31]. Within their stated assumptions, these approaches provide stronger theoretical guarantees than the present framework, although they also introduce additional

randomness requirements, implementation complexity, and platform-specific overhead. In contrast, the proposed framework emphasizes configurable algorithm-level diversification rather than formal higher-order masking security.

Recent protected-AES studies further demonstrate the importance of implementation-level evidence. Hardware-oriented countermeasures such as R-STELLAR provide signature attenuation and attack-on-countermeasure detection, while balanced encoding targets near-zero correlation in AES implementations [14], [32]. Because these approaches are evaluated directly on target implementations, they support stronger physical security claims. The proposed architecture offers greater algorithmic configurability but does not claim an equivalent level of implementation-level validation.

Deep-learning and electromagnetic side-channel studies likewise motivate a cautious interpretation of the reported results. Modern profiling attacks can exploit complex leakage characteristics that are not fully represented by conventional CPA or DPA assumptions [33]–[36]. Furthermore, amplitude-modulated electromagnetic analysis has shown that implementations protected under formal masking assumptions may remain vulnerable under alternative acquisition modalities and learning-based profiling [37]. Consequently, favorable simulated CPA, DPA, and TVLA indicators should not be interpreted as confirmation of resistance on physical devices.

Lightweight authenticated encryption provides another important comparison for constrained IoT systems. NIST SP 800-232 standardizes Ascon-based authenticated encryption, hashing, and extendable-output functions for resource-constrained environments [38]. Surveys of lightweight cryptography and Ascon implementations emphasize that area, power, throughput, energy consumption, and side-channel behavior remain implementation-dependent evaluation criteria [39], [40]. The proposed framework investigates hybrid AES-Blowfish diversification under simulation and should therefore be regarded as complementary to, rather than a replacement for, standardized lightweight cryptographic primitives.

Broader work on secure multimedia, cloud, and medical-data systems illustrates the increasing adoption of hybrid encryption, authentication, privacy protection, and application-specific security architectures [41]–[43]. These developments provide valuable application context for secure edge communication but do not eliminate the need for implementation-level side-channel validation. Overall, the comparison indicates that the proposed framework occupies a distinct research position by providing a configurable, simulation-based platform for evaluating algorithm-level protection strategies prior to hardware implementation and real-trace validation. Accordingly, the framework is positioned as a complementary approach to formal masking, threshold implementations, hardware countermeasures, and standardized lightweight AEAD rather than as their replacement.

6. Conclusions

This study presented a hybrid AES-Blowfish framework designed to investigate the combined effects of whitening, masking, modeled hiding activity, dynamic P-array generation, dynamic S-box initialization, and cascaded operation on modeled side-channel leakage and computational performance. Within the adopted simulation framework, the ablation analysis demonstrated that progressively integrating these protection mechanisms reduced the modeled TVLA, CPA, and DPA distinguishability relative to the baseline AES and Blowfish configurations. Among the evaluated operating modes, the cascaded configuration achieved the greatest modeled leakage reduction at the highest computational cost, whereas AES-Hybridization provided the most balanced trade-off between modeled leakage reduction and computational efficiency.

The findings should be interpreted within the scope of the adopted simulation methodology. The framework was evaluated using modeled leakage rather than physical measurements; consequently, no FPGA or microcontroller implementation, power or electromagnetic traces, oscilloscope acquisitions, or hardware-based side-channel experiments were included. Accordingly, the proposed framework should be regarded as a simulation-evaluated design candidate rather than a validated side-channel-resistant implementation.

Future work will focus on validating the proposed framework in practical environments through FPGA and microcontroller implementations with real power and electromagnetic trace acquisition. Additional evaluation should include TVLA, CPA/DPA, key-rank evolution, constant-time behavior, fault-injection resilience, energy consumption, memory

footprint, and hardware area overhead. Such implementation-level studies will provide stronger evidence regarding the practical applicability of the proposed framework for secure embedded, IoT, and edge communication systems.

Author Contributions: Conceptualization: W.W.S. and A.E.T.; Methodology: W.W.S. and A.E.T.; Software: F.K.; Validation: W.W.S., A.E.T., and M.F.; Formal analysis: W.W.S.; Investigation: M.F.; Resources: M.F.; Data curation: M.F.; Writing—original draft preparation: W.W.S.; Writing—review and editing: W.W.S., A.E.T., and M.F.; Visualization: F.K.; Supervision: A.E.T.; Project administration: M.F.; Funding acquisition: N/A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets generated and analyzed during this study are available in Supplementary File S1 DOI 10.5281/zenodo.21600205, which contains the raw simulation-derived datasets, summary tables, CSV files, high-resolution figures DOI 10.5281/zenodo.21600837, and Python code DOI 10.5281/zenodo.21600720 required to reproduce the reported analyses. No physical FPGA, microcontroller, IoT-node, power-trace, or electromagnetic measurement data were generated or analyzed in this study, as the reported results are based entirely on simulation.

Acknowledgments: The authors acknowledge the use of GenAI as an assistive tool for language editing and manuscript preparation. All scientific content, analyses, interpretations, and conclusions were verified and approved by the authors.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] W. Liu *et al.*, "Full-Element Analysis of Side-Channel Leakage Dataset on Symmetric Cryptographic Advanced Encryption Standard," *Symmetry (Basel).*, vol. 17, no. 5, p. 769, May 2025, doi: 10.3390/sym17050769.

[2] J. Mishra and S. K. Sahay, "Modern Hardware Security: A Review of Attacks and Countermeasures," *ArXiv*. Jan. 08, 2025. [Online]. Available: http://arxiv.org/abs/2501.04394

[3] S. Belaïd *et al.*, "SoK: A Methodology to Achieve Provable Side-Channel Security in Real-World Implementations," *IACR Commun. Cryptol.*, vol. 2, no. 1, 2025, doi: 10.62056/aebngy4e-.

[4] D. Ramakrishna and M. A. Shaik, "PSESV: A hybrid post-quantum encryption Framework with real-time thermal and EM side-channel attack detection," *Ain Shams Eng. J.*, vol. 16, no. 12, p. 103776, Dec. 2025, doi: 10.1016/j.asej.2025.103776.

[5] N. S. Dokku, R. David Amar Raj, S. K. Bodapati, A. Pallakonda, Y. R. M. Reddy, and K. Krishna Prakasha, "Resilient cybersecurity in smart grid ICS communication using BLAKE3-driven dynamic key rotation and intrusion detection," *Sci. Rep.*, vol. 15, no. 1, p. 32754, Sep. 2025, doi: 10.1038/s41598-025-17530-z.

[6] R. K. Muhammed *et al.*, "Comparative Analysis of AES, Blowfish, Twofish, Salsa20, and ChaCha20 for Image Encryption," *Kurdistan J. Appl. Res.*, vol. 9, no. 1, pp. 52–65, May 2024, doi: 10.24017/science.2024.1.5.

[7] V. T. Hoang and P. Rogaway, "On Generalized Feistel Networks," in *Lecture Notes in Computer Science*, 2010, pp. 613–630. doi: 10.1007/978-3-642-14623-7_33.

[8] R. C and K. G. N., "An optimized encryption algorithm and F function with dynamic substitution for creating S-box and P-box entries for blowfish algorithm," *Comput. Sci. Inf. Technol.*, vol. 2, no. 1, pp. 16–25, Mar. 2021, doi: 10.11591/csit.v2i1.p16-25.

[9] S. S. Abdul-Jabbar, A. E. Abed, S. G. Mohammed, and F. G. Mohammed, "Fast 128-bit Multi-Pass Stream Ciphering Method," *Iraqi J. Sci.*, pp. 2589–2600, May 2023, doi: 10.24996/ijs.2023.64.5.40.

[10] P. Kietzmann, T. C. Schmidt, and M. Wählisch, "A Guideline on Pseudorandom Number Generation (PRNG) in the IoT," *ACM Comput. Surv.*, vol. 54, no. 6, pp. 1–38, Jul. 2022, doi: 10.1145/3453159.

[11] Y. Chinnam and B. Sambana, "Artificial Intelligence enhanced Security Problems in Real-Time Scenario using Blowfish Algorithm," *arXiv*. Apr. 14, 2024. [Online]. Available: http://arxiv.org/abs/2404.09286

[12] P. Tippe and M. P. Berner, "Evaluating Argon2 Adoption and Effectiveness in Real-World Software," in *Lecture Notes in Computer Science*, 2025, pp. 25–46. doi: 10.1007/978-3-032-00627-1_2.

[13] A. S. Tushar, "Enhancement of AES Security On FPGA: Mitigating Side Channel Attacks," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 13, no. 6, pp. 2605–2611, Jun. 2025, doi: 10.22214/ijraset.2025.72696.

[14] A. Ghosh, D.-H. Seo, D. Das, S. Ghosh, and S. Sen, "R-STELLAR: A Resilient Synthesizable Signature Attenuation SCA Protection on AES-256 With Built-In Attack-on-Countermeasure Detection," *IEEE Open J. Solid-State Circuits Soc.*, vol. 5, pp. 167–179, 2025, doi: 10.1109/OJSSCS.2025.3571334.

[15] H. Li and G. Perin, "A systematic study of data augmentation for protected AES implementations," *J. Cryptogr. Eng.*, vol. 14, no. 4, pp. 649–666, Nov. 2024, doi: 10.1007/s13389-024-00363-3.

[16] G. Barthe, G. Betarte, J. Campo, C. Luna, and D. Pichardie, "System-level Non-interference for Constant-time Cryptography," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, Nov. 2014, pp. 1267–1279. doi: 10.1145/2660267.2660283.
[17] E. Almaraz Luengo and J. Román Villaizán, "Cryptographically Secured Pseudo-Random Number Generators: Analysis and Testing with NIST Statistical Test Suite," *Mathematics*, vol. 11, no. 23, p. 4812, Nov. 2023, doi: 10.3390/math11234812.
[18] C. Chevalier, G. Lebrun, and A. Martinelli, "Spilling-Cascade: An Optimal PKE Combiner for KEM Hybridization," in *Lecture Notes in Computer Science*, 2025, pp. 455–485. doi: 10.1007/978-3-031-95761-1_16.
[19] Et. al., Pravin Soni, "Performance Analysis of Cascaded Hybrid Symmetric Encryption Models," *Turkish J. Comput. Math. Educ.*, vol. 12, no. 2, pp. 1699–1708, Apr. 2021, doi: 10.17762/turcomat.v12i2.1506.
[20] M. Rivain and E. Prouff, "Provably Secure Higher-Order Masking of AES," in *Lecture Notes in Computer Science*, 2010, pp. 413–427. doi: 10.1007/978-3-642-15031-9_28.
[21] S. Nikova, C. Rechberger, and V. Rijmen, "Threshold Implementations Against Side-Channel Attacks and Glitches," in *Lecture Notes in Computer Science*, 2006, pp. 529–545. doi: 10.1007/11935308_38.
[22] L. E. Bassham *et al.*, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," Gaithersburg, MD, 2010. doi: 10.6028/NIST.SP.800-22r1a.
[23] M.-J. O. Saarinen, "SP 800–22 and GM/T 0005–2012 Tests: Clearly Obsolete, Possibly Harmful," in *2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, Jun. 2022, pp. 31–37. doi: 10.1109/EuroSPW55150.2022.00011.
[24] G. Goodwill, B. Jun, J. Jaffe, and P. Rohatgi, "A Testing Methodology for Side-Channel Resistance Validation," 2011. [Online]. Available: https://csrc.nist.gov/csrc/media/events/non-invasive-attack-testing-workshop/documents/08_goodwill.pdf
[25] International Organization for Standardization, "ISO/IEC 17825:2024 Information Technology -- Security Techniques -- Testing Methods for the Mitigation of Non-Invasive Attack Classes Against Cryptographic Modules," 2024.
[26] E. Brier, C. Clavier, and F. Olivier, "Correlation Power Analysis with a Leakage Model," in *Lecture Notes in Computer Science*, 2004, pp. 16–29. doi: 10.1007/978-3-540-28632-5_2.
[27] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Boston, MA: Springer US, 2007. doi: 10.1007/978-0-387-38162-6.
[28] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," in *Lecture Notes in Computer Science*, 1999, pp. 388–397. doi: 10.1007/3-540-48405-1_25.
[29] Y. Bai, R. Y. Acharya, and D. Forte, "SPERO: Simultaneous Power/EM Side-channel Dataset Using Real-time and Oscilloscope Setups," *ArXiv*. May 10, 2024. [Online]. Available: http://arxiv.org/abs/2405.06571
[30] J. Ming, Y. Zhou, H. Li, and Q. Zhang, "A secure and highly efficient first-order masking scheme for AES linear operations," *Cybersecurity*, vol. 4, no. 1, p. 14, Dec. 2021, doi: 10.1186/s42400-021-00082-w.
[31] A. R. Shahmirzadi, D. Božilov, and A. Moradi, "New First-Order Secure AES Performance Records," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2021, no. 2, pp. 304–327, Feb. 2021, doi: 10.46586/tches.v2021.i2.304-327.
[32] S. Lee and J.-N. Kim, "Balanced Encoding of Near-Zero Correlation for an AES Implementation," *IEEE Trans. Inf. Forensics Secur.*, vol. 19, pp. 6589–6603, 2024, doi: 10.1109/TIFS.2024.3420101.
[33] R. Benadjila, E. Prouff, R. Strullu, E. Cagli, and C. Dumas, "Deep learning for side-channel analysis and introduction to ASCAD database," *J. Cryptogr. Eng.*, vol. 10, no. 2, pp. 163–188, Jun. 2020, doi: 10.1007/s13389-019-00220-8.
[34] L. Masure and R. Strullu, "Side-channel analysis against ANSSI's protected AES implementation on ARM: end-to-end attacks with multi-task learning," *J. Cryptogr. Eng.*, vol. 13, no. 2, pp. 129–147, Jun. 2023, doi: 10.1007/s13389-023-00311-7.
[35] S. Picek, G. Perin, L. Mariot, L. Wu, and L. Batina, "SoK: Deep Learning-based Physical Side-channel Analysis," *ACM Comput. Surv.*, vol. 55, no. 11, pp. 1–35, Nov. 2023, doi: 10.1145/3569577.
[36] M. Panoff, H. Yu, H. Shan, and Y. Jin, "A Review and Comparison of AI-enhanced Side Channel Analysis," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 18, no. 3, pp. 1–20, Jul. 2022, doi: 10.1145/3517810.
[37] H. Wang, "Amplitude-modulated EM side-channel attack on provably secure masked AES," *J. Cryptogr. Eng.*, vol. 14, no. 3, pp. 537–549, Sep. 2024, doi: 10.1007/s13389-024-00347-3.
[38] M. S. Turan, K. A. McKay, D. Chang, J. Kang, and J. Kelsey, "Ascon-based lightweight cryptography standards for constrained devices :," Aug. 2025. doi: 10.6028/NIST.SP.800-232.
[39] C. Silva, N. Tenório, and J. Bernardino, "Lightweight Encryption Algorithms for IoT," *Computers*, vol. 14, no. 12, p. 505, Nov. 2025, doi: 10.3390/computers14120505.
[40] J. Kaur, A. Cintas Canto, M. Mozaffari Kermani, and R. Azarderakhsh, "A Survey on the Implementations, Attacks, and Countermeasures of the Current NIST Lightweight Cryptography Standard," *TechRxiv*. May 22, 2023. doi: 10.36227/techrxiv.22970855.
[41] S. Deb, A. A.-A. Gutub, and A. K. Sahu, Eds., *Fortressing Pixels: Information security for images, videos, audio and beyond*. United Kingdom: The Institution of Engineering and Technology, 2025. doi: 10.1049/PBSE030E.
[42] A. K. Chandanan, V. K. Sarathe, A. Dwivedi, R. Chandrasekaran, V. Roy, and A. K. Sahu, "Cloud-based analysis with quantum cryptography-based cloud security model (QC-CSM) for enhanced data security in storage and access," in *Fortressing Pixels*, United Kingdom: The Institution of Engineering and Technology, 2025, pp. 93–115. doi: 10.1049/PBSE030E_ch6.
[43] B. V. S. S. Praneeth, R. Ch, C. N. Manikanta, D. Pavan Kumar, M. Sahu, and A. K. Sahu, "Privacy protection of medical data using NTRU-based post-quantum cryptography," in *Fortressing Pixels*, United Kingdom: The Institution of Engineering and Technology, 2025, pp. 197–211. doi: 10.1049/PBSE030E_ch10.