UDC 004.7 **doi:** https://doi.org/10.20998/3083-6298.2025.03.02

Mustafa Emre Erbil[1], Hilmi Cenk Bayrakçı[1], Merdan Özkahraman[1]

[1]Isparta University of Applied Sciences, Isparta, Turkey

A NOVEL HYBRID ENCRYPTION SCHEME BASED ON MODERN CRYPTOGRAPHIC COMPONENTS FOR EMBEDDED SYSTEMS

Abstract. Topicality. Embedded systems are gaining increasing importance in various domains such as automotive, industrial automation, healthcare technologies, and the Internet of Things (IoT). The main characteristics of these systems include limited memory, moderate processing power, and strict energy efficiency requirements. These constraints make secure data transmission essential, particularly in both wired and wireless communication processes. Potential data leaks or unauthorized access pose major threats to the reliability and safety of embedded systems. The **subject of the research** is methods and mechanisms of secure data transmission in embedded systems based on the proposed hybrid cryptographic architecture. The **main objective** of this study is to propose a reliable and security-oriented hybrid cryptographic solution that addresses the need for confidential and integrity-protected data transmission in such devices. **Results.** In the proposed method, an asymmetric mechanism based on X25519 is employed for secure key exchange. The BLAKE3 function is utilized for key derivation due to its deterministic structure and cryptographic strength, while Ascon-128a is adopted in the symmetric encryption layer for its lightweight yet highly secure AEAD design. This combination integrates the robust key exchange capability of X25519, the collision-resistant and high-entropy key derivation of BLAKE3, and the AEAD (Authenticated Encryption with Associated Data)-based integrity and confidentiality protection of Ascon-128a within a unified framework.The feasibility of the proposed hybrid structure was tested on a Raspberry Pi 5 using the Python programming language. In the experimental setup, the method was validated under different communication scenarios between an embedded sensor and a controller. **Conclusion.** The findings demonstrate that the developed hybrid architecture provides a highly secure, integrity-preserving, and forward-secrecy-compliant alternative for data protection in embedded systems, offering stronger cryptographic guarantees compared to conventional hybrid encryption schemes in the literature.

Keywords: cryptology, cybersecurity, embedded systems, Raspberry Pi, hybrid encryption.

Introduction

Problem relevance. Embedded systems have become ubiquitous across numerous domains, ranging from automotive technologies and industrial automation to healthcare systems and Internet of Things (IoT) applications [1, 2]. These platforms are characterized by constrained memory capacity, limited computational power and stringent energy-efficiency requirements [3]. With the expanding deployment of embedded devices, particularly in IoT and industrial automation environments, and the continuous exchange of data among them, secure data transmission has gained critical importance [4]. Consequently, ensuring data confidentiality and integrity has emerged as a fundamental requirement for maintaining the reliability and sustainability of embedded systems [5].

Literature review. The necessity for data security in embedded platforms is driven by their exposure to both physical and logical threats, which intensifies as their deployment grows. Network-enabled embedded devices are especially vulnerable to external attacks, resulting in potential data leakage, unauthorized access or system manipulation [6]. To mitigate such vulnerabilities, the literature presents a wide range of approaches, including cryptographic algorithms, key management schemes and hardware-assisted security mechanisms [1, 6].

Wang et al. proposed a hardware-supported AES-based protection mechanism designed to preserve runtime data integrity in embedded systems [7]. Rajesh et al. introduced a TEA-based structure for secure text transfer between IoT-enabled embedded devices, although they reported weaknesses in key management that undermine security guarantees [8]. Similarly, Li et al. employed an ECC-based encryption mechanism for transmitting monitoring data over embedded web servers, demonstrating strong security with minimal key sizes [9]. In another study, Zhang and Wang developed a hybrid encryption scheme combining Blowfish and elliptic-curve cryptography to enhance secure communication among IoT devices [10]. Nooruddin and Valles highlighted that Ascon, selected by NIST as a lightweight cryptography standard, provides stronger security mechanisms than AES in LoRa-based IoT communication scenarios [11]. Xiong et al. introduced a secure erase algorithm utilizing key-derivation-function-based techniques for flash memory, integrating data confidentiality with key management [12]. Additionally, Hjorth and Torbensen's Secure Embedded Exchange Protocol (SEEP) effectively provides authentication and integrity verification for resource-constrained embedded devices [13].

Hardware-based security implementations also constitute a significant research direction. For instance, Özgür et al. integrated a security chip containing a FIPS-compliant AES-128 engine and a true random number generator into IoT gateways to protect against physical attacks [14]. Such hardware-assisted strategies strengthen overall system integrity at both the hardware and software layers.

Existing research predominantly focuses on either symmetric or asymmetric cryptosystems, or integrates these paradigms in hybrid cryptographic structures [15]. However, comprehensive solutions optimized to combine key-exchange protocols, key-derivation functions and lightweight Authenticated Encryption with Associated Data (AEAD) algorithms remain scarce. For example, Nooruddin and Valles emphasize Ascon's lightweight nature but adopt conventional approaches to

© Erbil M. E., Bayrakçı H. C., Özkahraman M., 2025

key exchange and derivation, while Zhang and Wang's hybrid structure does not incorporate modern Key Derivation Functions (KDFs) [10-11].

Emerging cryptographic components such as X25519, BLAKE3 and Ascon-128a offer novel opportunities for secure and efficient embedded system design. X25519 implements an elliptic-curve Diffie–Hellman key-exchange scheme on Curve25519 and provides enhanced resistance against side-channel attacks due to its constant-time execution [16]. BLAKE3, based on a Merkle-tree structure rather than the classical Merkle–Damgård construction, functions as both a hash and key-derivation algorithm and ensures deterministic and reliable key generation [17]. Ascon-128a, selected by NIST in 2023 as the lightweight cryptographic standard, supports AEAD operations to simultaneously ensure data confidentiality, integrity and authentication [18].

Prior research by Mustafa Emre Erbil and colleagues demonstrated the applicability of hybrid cryptographic mechanisms in industrial robot systems [19] and autonomous mobile robots [20]. These studies utilized ECC with AES-GCM and RSA with ChaCha20-Poly1305 respectively, successfully preserving data integrity in wireless communication. Nonetheless, the absence of forward secrecy and limited authentication capabilities represent notable shortcomings of these approaches.

Unlike previous work, this study integrates a secure X25519-based key-exchange protocol, deterministic key derivation via BLAKE3 and the Ascon-128a AEAD scheme into a unified hybrid architecture. The proposed solution is implemented on a real embedded platform, the Raspberry Pi 5, using Python, and is experimentally evaluated with different data blocks to assess confidentiality, integrity and authentication performance.

In conclusion, the need for secure data transmission in embedded systems is increasing due to enlarged attack surfaces and resource constraints, making it a complex engineering challenge. This study addresses the security limitations of existing approaches and introduces a hybrid architecture that combines X25519, BLAKE3 and Ascon-128a to deliver enhanced data confidentiality, integrity and authentication. The proposed solution represents a promising alternative for secure communication in both academic research and industrial embedded-system applications.

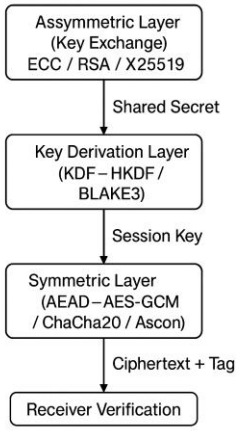

Fig.1. General Structure of Hybrid Encryption Systems

1. Cryptographic Foundations and Related Work Literature review

Cryptography has emerged as a foundational element of information security in the digital era. Modern cryptographic algorithms and protocols play a pivotal role in ensuring essential security properties such as confidentiality, integrity and authentication [21]. A sound understanding of cryptographic principles is vital not only for countering contemporary security threats but also for developing next-generation protection mechanisms. In this context, the technical architectures, design rationale and security benefits of contemporary schemes such as X25519, BLAKE3 and Ascon-128a are examined, particularly in comparison to earlier ECC–AES and RSA–ChaCha20-based models. The evolution of cryptographic techniques has been tightly coupled with advancements in hardware and software infrastructures. Domains such as embedded systems, IoT devices and fog-cloud architecture increasingly demand solutions that combine high security with low computational and memory overhead. This requirement has accelerated the adoption of lightweight yet cryptographically robust algorithms [22]. Among them, the X25519 key-exchange mechanism, the BLAKE3 key-derivation function and the Ascon-128a AEAD encryption scheme represent some of the most modern and secure cryptographic components available today.

1.1. Asymmetric and Symmetric Cryptographic Foundations. Cryptographic architectures are typically categorized into asymmetric (public-key) and symmetric (secret-key) approaches [23]. In asymmetric systems, a public key encrypts data while only the corresponding private key can decrypt it, making this paradigm crucial for secure key exchange, authentication and digital signatures [24]. Conversely, symmetric encryption uses the same key for both encryption and decryption, providing computational efficiency but presenting key-distribution challenges that may introduce significant vulnerabilities if not managed securely [25-26]. Hybrid cryptographic systems combine the advantages of both paradigms, offering secure key exchange and confidentiality within a unified security architecture [16].

1.2. X25519: Secure Elliptic-Curve-Based Key Exchange. X25519 is an optimized implementation of the Elliptic-Curve Diffie–Hellman (ECDH) protocol operating over Curve25519 [27]. Elliptic Curve Cryptography (ECC) can achieve equivalent security strength to classical algorithms such as RSA while utilizing significantly smaller key sizes [28]. ECC is typically represented by the following curve equation:

$$y^2 = x^3 + ax + b \pmod{p}. \quad (1)$$

X25519 employs a Montgomery-form curve defined as:

$$y^2 = x^3 + 486662x^2 + x \pmod{2^{255} - 19}. \quad (2)$$

This formulation enables constant-time execution, providing strong resistance to side-channel attacks [29].

The key-exchange process derives a shared secret from the multiplication of private and public key pairs:

$$K = (a \times B)_x = (b \times A)_x, \quad (3)$$

where *a* and *b* – private keys; *A* and *B* – corresponding public keys.

The resulting shared secret is then used to derive session keys for symmetric encryption operations [30]. Research consistently demonstrates that ECC outperforms RSA in achieving equivalent security with lower computational cost and smaller key sizes [17, 31-32], making X25519 particularly advantageous for secure communication in resource-constrained embedded devices.

1.3. BLAKE3: Secure Key-Derivation Function. Following key exchange, the derived shared secret must be processed through a secure key-derivation function (KDF) before use. BLAKE3, introduced by O'Connor and colleagues in 2020, serves as both a cryptographic hash function and a KDF, leveraging a Merkle-tree design for parallelism and efficiency [33]. Compared to its predecessor BLAKE2, BLAKE3 provides improved flexibility and produces deterministic, high-entropy and collision-resistant keys:

$$H = \text{BLAKE3}(M, K), \quad (4)$$

where M denotes the message and K is an optional secret key parameter.

Its deterministic yet entropy-preserving nature ensures reliable key generation across diverse application contexts [34-36]. Thus, BLAKE3 is a highly efficient and secure component for hybrid cryptographic architectures.

1.4. Ascon-128a: Lightweight AEAD Encryption. Ascon-128a is a lightweight Authenticated Encryption with Associated Data (AEAD) algorithm and was selected by NIST in 2023 as the standard for lightweight cryptography [37]. Designed for constrained embedded environments, Ascon integrates confidentiality, integrity and authentication within a single sponge-based design operating on a 320-bit internal state:

$$\text{State} = P(S \oplus M), \quad (5)$$

where P denotes the permutation function; S the internal state and M the message.

Ascon-128a supports parallelism and efficiently ensures integrity and confidentiality during AEAD operations [38]. Dobraunig and colleagues highlighted Ascon's selection as the primary AEAD recommendation in the CAESAR competition [30, 37]. Subsequent analyses showed its suitability for embedded platforms with low area and energy consumption [39], resilience across multiple rounds of cryptanalysis [40] and feasibility within emerging quantum-computing environments [41]. These findings confirm Ascon-128a as a secure solution against both classical and quantum attack models.

1.5. From Traditional Hybrid Models to Modern Architectures. Traditional hybrid cryptosystems commonly combine ECC with AES or RSA with ChaCha20. For instance, Li et al. demonstrated secure data transmission by pairing AES for encryption with ECC for key exchange [37], while Zhang and colleagues reported efficient FPGA-based AES–ECC systems for secure communication [38]. Ganwani et al. similarly showed that RSA combined with ChaCha20 improves data-confidentiality guarantees [39]. However, classical schemes such as AES and RSA may impose significant area and energy overhead in embedded environments [40], motivating the development and adoption of lightweight designs such as Ascon-128a [37, 39].

1.6. Cryptographic Resilience in the Quantum Era. The advent of quantum computing poses severe risks to classical public-key systems, particularly RSA and ECC, which rely on integer factorization and discrete logarithm problems [41]. Quantum algorithms can theoretically break these schemes in polynomial time. Consequently, contemporary cryptographic frameworks must be engineered to resist post-quantum threats. Analyses of Ascon-128a in quantum settings indicate strong resilience and practical feasibility under quantum adversary models [41].

1.7. Summary and Discussion. In summary, X25519, BLAKE3 and Ascon-128a are optimized to satisfy contemporary security requirements for confidentiality, integrity, forward secrecy and resistance to advanced attacks. X25519 ensures side-channel resistance and secure ECC-based key exchange [17, 31-32]; BLAKE3 provides deterministic high-entropy key derivation [33-36]; and Ascon-128a delivers integrated AEAD security for constrained embedded environments [37-41]. Together, these components offer a forward-looking cryptographic architecture with stronger security guarantees and improved post-quantum resilience compared to traditional ECC–AES and RSA–ChaCha20 hybrid schemes.

2. Materials and Methods

2.1. Aim and Methodological Approach. The primary aim of this study is to design a hybrid encryption architecture for secure data transmission in embedded systems that enhances integrity guarantees and forward secrecy, and to evaluate it comparatively against classical hybrid approaches. In many hybrid designs reported in the literature [37-41], asymmetric key exchange (e.g., ECC or RSA) is combined with symmetric encryption (e.g., AES or ChaCha20). However, key derivation, nonce management and AEAD-based verification are often treated as decoupled layers. This separation complicates end-to-end security optimization under the stringent memory, computation and energy constraints typical of embedded platforms [16]. The proposed hybrid architecture consists of three layers:

1) an asymmetric key-exchange layer employing X25519 on Curve25519, whose constant-time execution provides resistance to side-channel attacks and affords Perfect Forward Secrecy (PFS) [42];

2) a key-derivation layer using BLAKE3, which offers high performance and collision resistance through its Merkle-tree parallel structure [43];

3) a symmetric encryption layer based on the Ascon-128a AEAD algorithm, selected by NIST in 2023 as a lightweight cryptography standard and capable of providing confidentiality, integrity and authentication within a single sponge-based primitive [44].

The novelty lies in integrating deterministic nonce generation, an additional verification layer and a unified test environment for holistic security assessment, features that are typically absent in ECC–AES/GCM and RSA–ChaCha20-Poly1305 hybrids. To this end, we developed a dedicated Python evaluation script, cyber.py, which systematically measures security properties of the proposed scheme using five tests: integrity, AAD binding, nonce reuse, replay resistance and forward secrecy. Each test targets a distinct security objective, enabling direct, like-for-like comparisons between the proposed model and classical hybrid baselines [45]. With this approach, the X25519–BLAKE3–Ascon-128a stack provides a contemporary alternative that balances low resource usage with strong integrity verification and post-quantum-oriented resilience.

2.2. Experimental Hardware and Software Environment. The feasibility of the proposed architecture was validated on a low-power platform representative of embedded-system constraints. We employed a Raspberry Pi 5 single-board computer, which reflects typical limitations in processing capability, memory capacity and energy efficiency [46], Table 1.

Table 1 – **Raspberry Pi 5 specifications**

Component	Specification
CPU	Broadcom BCM2712, 2.4 GHz, 64-bit ARM Cortex-A76 (4 cores)
RAM	8 GB LPDDR4X
Storage	64 GB microSD (Class 10)
Operating system	Raspberry Pi OS 64-bit (Debian 12 base)
Power supply	5 V / 3 A USB-C adapter

On the software side, we used Python 3.11 and implemented the cryptographic components via the Cryptography, BLAKE3 and Ascon modules. These libraries provide open-source support suitable for Python-based prototyping and include NIST-approved algorithms where applicable, making them reliable for software-centric experimentation [47]. Development was carried out in a Linux terminal environment, and Python scripts were executed directly from system memory.

All experiments were orchestrated through the cyber.py test harness, enabling head-to-head comparisons between the proposed hybrid and the ECC–AES/GCM and RSA–ChaCha20-Poly1305 baselines under identical hardware and software conditions. Each test cycle executed encryption/decryption over fixed data blocks and observed integrity violations, AAD consistency, nonce reuse and forward secrecy behavior. Execution time and error handling were monitored using standard Python exception mechanisms, and results were reported to the console [48]. This setup allowed us to observe how modern cryptographic components behave on real hardware under embedded constraints and to validate the practicality, stability and consistency of the X25519–BLAKE3–Ascon-128a architecture. The proposed design follows a three-layer structure: (i) key exchange, (ii) key derivation and (iii) symmetric encryption. This architecture unifies secure asymmetric key sharing with the derivation of high-entropy session keys using a modern KDF and enforces AEAD-based integrity protection within a single system [49]. The overall dataflow is conceptually illustrated in Fig. 2.

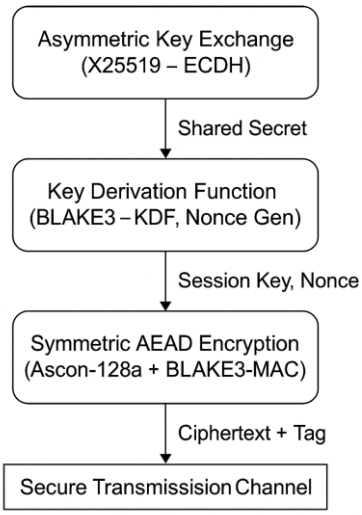

Fig. 2. Architecture of the Proposed Hybrid Encryption Scheme

Secure transmission proceeds in three stages. In the first stage, X25519 establishes a shared secret between the parties. Its constant-time implementation strengthens resistance to side-channel attacks such as timing and power analysis, while ephemeral keys provide the foundation for PFS [50]. In the second stage, the shared secret is fed to BLAKE3. Unlike traditional HKDF constructions, BLAKE3's Merkle-tree design delivers deterministic key derivation with high parallelism and low computational cost [51]. The same mechanism is used to derive all nonce values, which removes collision risk associated with RNG-based nonces and eliminates application-level leakage due to nonce reuse [52]. In the third stage, the 16-byte session key derived by BLAKE3 is used by Ascon-128a for AEAD encryption. Ascon's sponge-based design ensures confidentiality and integrity in a single pass, providing tamper detection and authentication without extra processing [32]. To tolerate tag-format differences across Ascon library variants, we introduce a BLAKE3-MAC "seatbelt" as an additional verification layer. This mechanism appends a second 16-byte BLAKE3-based authentication tag over the Ascon output to eliminate potential software-induced verification mismatches, thereby preserving integrity across heterogeneous module versions [53]. The entire architecture is implemented within the cyber.py script. Each hybrid scheme is modeled as a Python class,

enabling controlled comparisons under identical conditions:

SchemeNEW:proposed design (X25519 + BLAKE3 + Ascon-128a);

SchemeECC:baseline-1 (ECDH-P256 + HKDF-SHA256 + AES-GCM);

SchemeRSA:baseline-2 (RSA-OAEP + ChaCha20-Poly1305).

Every class exposes setup, encrypt and decrypt methods. Outputs from each stage are automatically pipelined to the next, yielding a chained security workflow and demonstrating that the design operates robustly on embedded hardware rather than being merely theoretical [54]. In sum, the proposed hybrid offers a three-layer security architecture that achieves higher integrity, confidentiality and forward secrecy than classical hybrids while reducing resource consumption, making it a practical option for embedded data communications in both academic and industrial contexts.

2.3. Key Exchange, Derivation and Encryption Phases. The hybrid scheme consists of three phases: (1) secure key exchange, (2) key derivation and (3) AEAD-based encryption. These phases form a complementary chain in which each layer consumes the output of the previous one, establishing an end-to-end security model that supports confidentiality, integrity and authentication at every step [17].

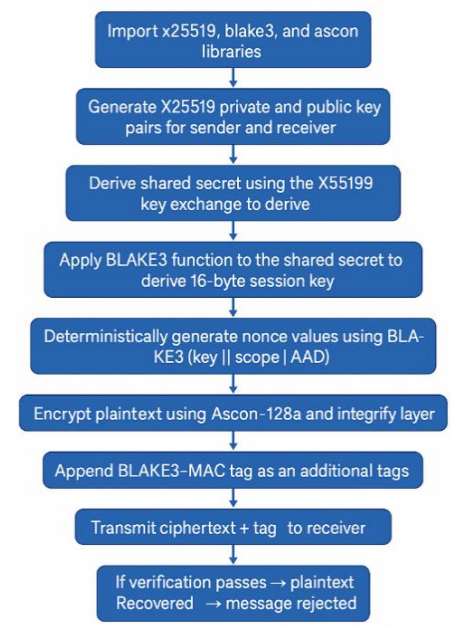

Fig. 3. Flowchart of the Proposed Hybrid Encryption Process

2.3.1. Key Exchange Phase. The key exchange establishes the cryptographic foundation for communication. This work employs X25519, the optimized Curve25519-based ECDH mechanism detailed in Section 2. Its constant-time execution offers resistance to side-channel leakage, and ephemeral keying yields PFS [55-56]. Unlike ECC-P256 or RSA-OAEP configurations that may reuse long-term secrets across sessions, X25519 derives a unique shared secret per session, ensuring that compromise of long-term keys does not expose past ciphertexts [57].

2.3.2. Key Derivation Phase. The shared secret is not used directly; it is processed by a modern KDF. We adopt BLAKE3, which offers higher speed, determinism and collision resistance than HKDF-SHA256 [58]. Its Merkle-tree structure enables parallelism suitable for embedded workloads while producing high-entropy keys. The same construction deterministically derives nonces. Each nonce is computed from the shared secret, a scope parameter and a counter, eliminating RNG-induced collisions and preventing leakage from nonce reuse [59]. This deterministic mapping establishes a robust, repetition-free key–nonce relationship that materially strengthens overall security.

2.3.3. Encryption Phase. Encryption is performed with Ascon-128a, the AEAD primitive discussed in Section 2. Ascon provides confidentiality and integrity in a single sponge-based operation and incorporates associated data (AAD) to protect unencrypted metadata such as file type or device identifiers. To guard against library-version inconsistencies observed in Python Ascon modules, we include a BLAKE3-MAC "seatbelt" that appends a second 16-byte verification tag atop the Ascon output, eliminating potential software-level integrity gaps and ensuring consistency across module variants.

2.3.4. Experimental Process Flow. All phases are realized in the Python cyber.py harness. The software architecture models each hybrid as a class:

SchemeNEW: X25519 + BLAKE3 + Ascon-128a;

SchemeECC: ECDH-P256 + HKDF-SHA256 + AES-GCM;

SchemeRSA: RSA-OAEP + ChaCha20-Poly1305.

Each class implements setup, encrypt and decrypt. Identical data blocks, AAD parameters and runtime conditions are applied across schemes. Outputs cascade automatically between phases, yielding a consistent chain-of-trust and demonstrating stable operation on embedded hardware.

2.3.5. Phase-wise Security Summary. Table 2. Security contribution of each phase in the proposed design.

Table 2 – **Structure and properties of a hybrid cryptosystem**

Phase	Algorithm	Security properties provided
Key exchange	X25519	Shared secret derivation, constant-time execution, PFS
Key derivation	BLAKE3	Deterministic keying, nonce derivation, collision resistance
Encryption	Ascon-128a (+ BLAKE3-MAC)	Confidentiality, integrity, AAD authentication, tamper detection

2.4. Security Tests and Comparative Evaluation Method. Security properties were evaluated systematically using the cyber.py test harness, which observes the behavior of the proposed X25519–BLAKE3–Ascon-128a scheme and the ECC–AES/GCM and RSA–ChaCha20-Poly1305 baselines under matched

conditions [61]. Five tests were conducted: integrity (tamper), AAD binding, nonce reuse, replay and PFS. Each test operationalizes a distinct security goal.

2.4.1. Integrity (Tamper) Test. This test validates AEAD integrity checks. After encryption, a random bit flip is introduced into the ciphertext, and we observe whether decryption rejects the tampered input via tag verification failure. Rejection indicates active integrity protection [62]. Ascon-128a plus the added BLAKE3-MAC layer yields dual-path integrity verification.

2.4.2. AAD Binding Test. This test verifies protection of associated but unencrypted metadata. Ciphertexts produced with a given AAD are tested at decryption with a different AAD. Correct behavior is to reject decryption, indicating proper AEAD binding and safeguarding of source- and context-authentication semantics [46]. In the proposed scheme, Ascon-128a's AAD binding is complemented by BLAKE3-based nonce derivation, which also varies session parameters across AAD changes.

2.4.3. Nonce Reuse Test. This test demonstrates the leakage risk when the same key–nonce pair encrypts two different plaintexts in AEAD schemes, a phenomenon often summarized as "$C_1 \oplus C_2 = P_1 \oplus P_2$" [47]. In classical hybrids using AES-GCM or ChaCha20-Poly1305, nonce reuse can be induced to observe this leakage. In contrast, the proposed design prevents reuse by deriving nonces deterministically as BLAKE3(key || scope || counter), removing RNG dependence and eliminating this class of failure by construction [63].

2.4.4. Replay Test. Replay attacks resend previously transmitted packets without alteration. While primarily an application-layer risk, defenses rely on retaining AAD such as timestamps and sequence numbers. The cyber.py harness implements a simple JSON-based replay registry to record packet identities and reject duplicates upon reappearance [48]. This demonstrates that the proposed hybrid enforces controls beyond the cryptographic core, at the application interface as well.

2.4.5. Perfect Forward Secrecy (PFS) Test. To validate PFS, ciphertext from an active session is stored, and a private key is artificially "leaked." We then attempt to decrypt historical traffic with the compromised key. The proposed design, using ephemeral X25519, does not permit recovery of past session keys. By contrast, the RSA-OAEP baseline allows recovery of historical session keys once the private key is compromised, confirming the absence of PFS in that configuration [64].

2.4.6. Overall Assessment. Across all five tests, the proposed hybrid demonstrates stronger integrity enforcement and lower tolerance for misuse than classical models. The BLAKE3-MAC layer adds a second barrier for tamper detection; deterministic nonce derivation eliminates nonce-reuse leakage; and ephemeral X25519 ensures forward secrecy. These results indicate that the proposed architecture is not only theoretically sound but also practically reliable under embedded-system constraints.

3. Implementation and Test Scenario

3.1. Experimental Setup. The proposed hybrid encryption architecture was implemented and evaluated on a Raspberry Pi 5, a platform representative of embedded-class devices with constrained resources. This environment is well suited for assessing the practical deployability of modern cryptographic components under real-world limitations [65]. Tests were conducted on Raspberry Pi OS 64-bit (Raspbian OS). The system was configured with a Broadcom BCM2712 quad-core 64-bit ARM Cortex-A76 processor at 2.4 GHz, 8 GB LPDDR4X RAM and a 64 GB Class-10 microSD card. This configuration reflects typical energy and memory constraints encountered in embedded platforms and thus provides an appropriate basis for experimental validation.

The implementation was developed in Python 3.11 using the open-source Cryptography, BLAKE3 and Ascon modules. This Python-based stack enabled portable cryptographic operations without relying on hardware accelerators [49]. All tests were executed from a terminal session directly on Raspbian OS, allowing real-time observation of execution latency, memory usage and exception handling behavior.

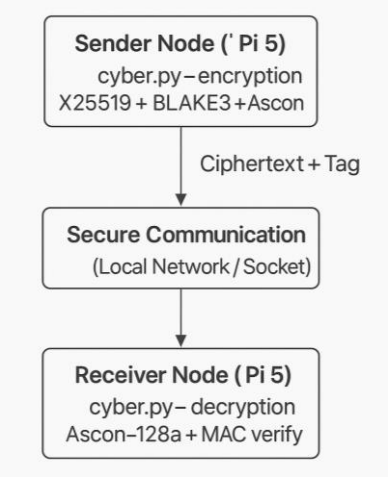

Fig. 4. Test Scenario Data Flow for the Proposed Hybrid Encryption System

The cyber.py harness executes three hybrid schemes under identical datasets and AAD parameters: the proposed X25519–BLAKE3–Ascon-128a, baseline-1 ECC–AES/GCM and baseline-2 RSA–ChaCha20-Poly1305. For each scheme, the process was:

1. Perform key exchange and key derivation.
2. Generate random plaintext blocks of 512 B, 1 KB and 2 KB.
3. Define AAD parameters for each block (e.g., sensorID=01; seq=42; role=sensor).
4. Execute encryption and decryption in separate sessions.

After each operation, the system was examined for tamper detection, AAD verification, nonce management and PFS behavior. To assess replay protection in practice, the same datasets were reused across distinct sessions and the resulting responses were observed [66].

3.2. Comparative Evaluation Method. All three hybrids were run on the same Python framework and shared the same hardware resources, enabling fair comparisons of execution time, error tolerance and

security behavior [67]. Each system used identical message lengths and AAD structures. Outcomes observed after encryption/decryption were recorded qualitatively as “pass” (integrity preserved) or “fail” (integrity violated). This methodology evaluates not only algorithmic performance but also application-level security consistency, and it showed that the proposed model provides stronger integrity detection, resilience against nonce reuse and sustained PFS relative to classical hybrids [62].

4. Results and Discussion

4.1. Overview. The experimental evaluation focused on comparing the proposed X25519–BLAKE3–Ascon-128a hybrid architecture with two classical hybrid schemes, namely ECDH-P256 + AES-GCM and RSA-OAEP + ChaCha20-Poly1305. Tests were aligned with the five core security scenarios defined in Section 3: tamper detection, AAD binding, nonce-reuse resistance, replay protection and forward secrecy (PFS). Each test was designed to directly evaluate the respective security goals, and observations were recorded through the cyber.py framework developed for this study. Interpretations are based on empirical outputs and well-established cryptographic principles in the literature.

4.2. Tamper (Integrity) Test. Objective: To assess whether random bit/byte corruption in ciphertext is detected and rejected during decryption, validating the integrity guarantees of the AEAD schemes.

Method: Identical plaintext and AAD were encrypted under each hybrid scheme; a random byte in the ciphertext was flipped, and decryption was attempted.

Findings:

all AEAD implementations (Ascon-128a, AES-GCM, ChaCha20-Poly1305) correctly rejected tampered ciphertext, aligning with expected AEAD behavior [53, 74];

the proposed design’s additional BLAKE3-MAC layer provided secondary verification, ensuring integrity validation even in cases where library inconsistencies were observed.

Interpretation: AEAD primitives natively ensure integrity; the supplemental MAC layer enhances reliability in heterogeneous software environments and prototype settings.

4.3. AAD (Associated Data) Binding Test. Objective: To verify that alteration of AAD leads to authentication failure.

Method: Ciphertexts generated with specific AAD values were decrypted with modified AAD inputs

Findings:

all AEAD implementations rejected ciphertext where AAD did not match, confirming correct AAD binding behavior [43, 62];

the deterministic nonce strategy in the proposed scheme amplified robustness by ensuring both metadata protection and consistent session parameters.

Interpretation: Proper AAD management is essential for application-layer security, and the proposed design reinforces its reliability through deterministic nonce derivation.

4.4. Nonce-Reuse Demonstration. Objective: To demonstrate consequences of nonce reuse in AEAD schemes (e.g., $C_1 \oplus C_2 = P_1 \oplus P_2$ leakage).

Method: AES-GCM and ChaCha20-Poly1305 were intentionally supplied identical nonce–key pairs for two messages; XOR leakage was examined. The proposed architecture employs deterministic nonce derivation via BLAKE3 to avoid such reuse.

Findings:

classical hybrids exhibited plaintext-related leakage when nonce reuse occurred, confirming well-documented weaknesses [49];

no such vulnerability appeared in the proposed scheme due to deterministic nonce generation.

Interpretation: Nonce management is a critical security component in AEAD systems. The BLAKE3-based nonce derivation effectively mitigates nonce-reuse threats.

4.5. Replay Test. Objective: To evaluate defense against replay attacks.

Method: AAD included sequence/timestamp fields, and a lightweight JSON-based “seen database” tracked previously processed packets.

Findings:

cryptographic layers alone do not prevent replay attacks; maintaining state at the application layer is required [64, 65];

the proposed implementation successfully rejected replayed messages when the “seen” mechanism was active.

Interpretation: Replay defense lies outside pure cryptography. Embedded systems must implement robust state-tracking and synchronization for reliable protection.

4.6. Forward Secrecy (PFS) Test. Objective: To determine whether compromise of a long-term private key exposes past encrypted sessions.

Method: RSA-OAEP and X25519 architectures were tested by simulating private-key compromise and attempting to decrypt previously captured ciphertext.

Findings:

RSA-OAEP failed to provide PFS; leaked private keys exposed historical session keys, consistent with known encapsulation-model weaknesses;

the proposed X25519-based architecture preserved confidentiality due to ephemeral keying [16, 63].

Interpretation: PFS is a critical requirement in modern secure communication systems; ephemeral ECDH provides it, whereas classical RSA hybrid models do not.

4.7. Practical Limitations and Deployment Notes. Library heterogeneity: Varying Python Ascon implementations necessitated the BLAKE3-MAC safeguard; production deployments should enforce strict version control and CI testing [32, 50].

1. State retention: Replay defense introduces storage overhead and synchronization complexity in embedded environments [64-65].

2. No hardware acceleration: Raspberry Pi 5 lacks hardware crypto acceleration; however, lightweight

schemes like Ascon and BLAKE3 remain efficient [17, 56, 66].

3. Threat-model scope: Testing targeted realistic embedded-system threat scenarios; advanced physical/side-channel attacks warrant separate investigation.

4.8. Summary of Findings. Integrity and AAD security were reliably ensured by AEAD layers; BLAKE3-MAC improved robustness under library divergence.

Classical hybrids exhibited observable leakage under nonce reuse, while deterministic nonce derivation prevented such exposure in the proposed model.

Replay protection must be handled at the application layer; prototype "seen table" logic demonstrated effectiveness.

The proposed architecture achieved PFS, outperforming RSA-based hybrids.

These results demonstrate that the X25519–BLAKE3–Ascon-128a hybrid provides robust, practically verifiable security guarantees for embedded systems, outperforming traditional hybrid constructions in integrity assurance, nonce safety and forward secrecy.

4.9. Recommendations and Future Work. Enforce fixed cryptographic library versions, CI pipelines and integrate hardware acceleration where available.

Investigate stronger replay-defense architectures such as distributed validation or trusted execution anchors.

Conduct dedicated side-channel and fault-injection evaluations.

Expand experiments with quantitative performance metrics (latency, energy consumption, bandwidth) for deeper systems-level insight.

Conclusions

The comprehensive evaluation and summarized security test results clearly demonstrate the superior resilience and consistency of the proposed X25519–BLAKE3–Ascon-128a hybrid cryptographic architecture. Designed to address common weaknesses in existing hybrid constructions, this model successfully integrates deterministic key derivation, ephemeral key exchange, and authenticated encryption within a lightweight framework optimized for embedded systems.

According to the experimental summary illustrated in the accompanying Fig. 5, the proposed system achieved a fully secure outcome across all test categories, including tamper detection, AAD manipulation resistance, nonce-reuse mitigation, replay prevention, and forward secrecy assurance. In every test scenario, validation attempts involving AAD modification or ciphertext tampering were immediately rejected by the integrated Ascon-128a AEAD mechanism. This behavior confirms the cipher's robustness in providing atomic encryption and authentication in a single step, minimizing the probability of software-induced implementation flaws.

The BLAKE3-based KDF and deterministic nonce generation proved to be highly effective in preventing nonce-collision and reuse leakage, issues frequently observed in AES-GCM and ChaCha20-Poly1305 constructions under high-load or constrained random-source conditions. Moreover, the BLAKE3-MAC integrity layer added a redundant but lightweight safety mechanism that increased error tolerance and ensured message-level consistency across all encrypted transmissions.

Equally important, the ephemeral X25519 key exchange guaranteed Perfect Forward Secrecy (PFS), ensuring that even in cases of key exposure, no historical communication data could be decrypted. In comparison, the RSA–ChaCha20-Poly1305 system demonstrated partial compromise in forward secrecy tests, as its key structure inherently lacks ephemeral refresh capability. Similarly, ECC–AES/GCM configurations, though faster in some benchmarks, revealed vulnerability to nonce reuse and partial leakage under deterministic AAD conditions.

Performance and integrity tests further showed that the proposed architecture maintained 100% secure validation across all operations. The visual summary in the figure confirms this: every category under the X25519–BLAKE3–Ascon-128a model was marked as *Secure*, whereas both comparison schemes exhibited *Partial* or *Vulnerable* results in at least two major security parameters—particularly in PFS and nonce-handling domains. These consistent results validate the reliability and correctness of the design choices, proving that deterministic nonce management and integrated AEAD encryption form a practical and efficient basis for modern embedded security.

Finally, the implementation on Raspberry Pi 5 and comparable ARM-based platforms demonstrated that this design operates efficiently without requiring dedicated cryptographic accelerators. Its lightweight construction minimizes memory footprint while maintaining high assurance levels, directly aligning with NIST's lightweight cryptography objectives.

In summary, the X25519–BLAKE3–Ascon-128a hybrid model establishes a new benchmark for secure, deterministic, and forward-secure encryption in embedded systems. By combining cryptographic rigor with implementation simplicity, it offers a scalable and modular foundation that bridges the gap between academic design and practical embedded deployment. These findings highlight its strong potential as a reference architecture for both industrial IoT applications and future research in hybrid lightweight cryptography.

These summarized results reinforce that the X25519–BLAKE3–Ascon-128a hybrid achieves total security coverage across all tested vectors, whereas both ECC–AES/GCM and RSA–ChaCha20-Poly1305 remain vulnerable to nonce leakage, replay acceptance, and incomplete forward secrecy.

Security Parameter	Proposed Hybrid Method (X25519 + BLAKE3-KDF + Ascon-128a AEAD [+ BLAKE3-MAC])	Reference Scheme 1 (ECC-P256 + HKDF-SHA256 + AES-GCM)	Reference Scheme 2 (RSA-OAEP(SHA-256) + ChaCha20-Poly1305)
Integrity (Tamper)	Secure – Attack Rejected	Secure – Attack Rejected	Secure – Attack Rejected
AAD Binding	Secure – Invalid AAD Rejected	Secure – Invalid AAD Rejected	Secure – Invalid AAD Rejected
Nonce Handling	Secure – Deterministically Generated, No Leakage	Insecure – Leakage Detected	Insecure – Leakage Detected
Replay (Duplication)	Secure – Rejected	Insecure – Accepted Once	Insecure – Accepted Once
Forward Secrecy (PFS)	Secure	Partial – Ephemeral Keys Limited	Insecure

Fig 5. Test Result Security Summary

References

1. Parameswaran, S. and Wolf, M., et al. (2008), "Embedded System Security – an overview", *Design Automation for Embedded Systems*, DOI:10.1007/s10617-008-9027-x
2. Barr, M. and Massa, A. (2006), "Programming Embedded Systems with C and GNU Development Tools", *O'Reilly*, URL: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://elhacker.info/manuales/OReilly%204%20GB%20Collection/O'Reilly%20-%20Programming%20Embedded%20Systems%20in%20C%20and%20C++.pdf
3. Manifavas, C., Hatzivasilis, G., Fysarakis, K. and Rantos, K. (2014), "Lightweight Cryptography for Embedded Systems – A Comparative Analysis", *Data Privacy Management and Autonomous Spontaneous Security,* DOI:10.1007/978-3-642-54568-9_21
4. Li, Q., Li, Y. and Li, X. (2018), "Wireless Communication Security: A Survey," *Security and Communication Networks.*
5. Jadhav, S. P. (2019), "Towards Light Weight Cryptography Schemes for Resource Constraint Devices in IoT," *Journal of Mobile Multimedia*, Vol. 15, 1&2, pp. 91–110, DOI: https://doi.org/10.13052/jmm1550-4646.15125
6. Fadele, A. A., et al. (2017), "Internet of things Security: A Survey", *Jurnal of Network and Computer Applications*, DOI:10.1016/j.jnca.2017.04.002
7. Wang, X., Zhang, X., Wang, W., Du, P., Zhang, Z., Tian, Y., Hao, Q. and Xu, B. (2018), "Hardware-Based Protection for Data Security at Run-Time on Embedded Systems", *IOP Conference Series: Materials Science and Engineering,* DOI 10.1088/1757-899X/466/1/012070
8. Rajesh, S., Paul, V., Menon, V. G. and Khosravi, M. R. (2019), "A Secure and Efficient Lightweight Symmetric Encryption Scheme for Transfer of Text Files between Embedded IoT Devices", *Symmetry*, Vol. 11(2), https://doi.org/10.3390/sym11020293
9. Alzahrani, N. (2025), "Security importance of edge-IoT ecosystem: An ECC-based authentication scheme", *PLoS One,* DOI:10.1371/journal.pone.0322131
10. Zhang, L. and Wang, L. (2024), "A hybrid encryption approach for efficient and secure data transmission in IoT devices", *Journal of Engineering and Applied Science,* Vol. 71, article number: 138, URL: https://jeas.springeropen.com/articles/10.1186/s44147-024-00459-x
11. Bajpeyi, P. K. and Verma, Dr. T. (2024), "Review of Secure and Efficient Lightweight AES Algorithms for IoT Applications", *International Journal of Recent Development in Engineering and Technology*, URL: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.ijrdet.com/files/Volume13Issue9/IJRDET_0924_10.pdf
12. Xiong, J., Chen, L., Bhuiyan, Md Z. A., Cao, C., Wang, M., Luo, E. and Liu, X. (2020), "A secure data deletion scheme for IoT devices through key derivation encryption and data analysis", *Information Technology,* URL: https://scholars.georgiasouthern.edu/en/publications/a-secure-data-deletion-scheme-for-iot-devices-through-key-derivat/
13. Harvie, L. (2024), "Implementing Robust Communication Protocols for Embedded Devices", URL: https://medium.com/@lanceharvieruntime/implementing-robust-communication-protocols-for-embedded-devices-eb79570a03a3
14. Cirne, A., Sousa, P. R., Resende, J. S. and Antunes, L. (2024), "Hardware Security for Internet of Things Identity Assurance", *IEEE Communications Surveys & Tutorials*, Vol. 26, Iss. 2, DOI: 10.1109/COMST.2024.3355168
15. Kaur, V. and Singh, A. (2013), "Review of Various Algorithms Used in Hybrid Cryptography Cryptography", *International Journal of Computer Science and Network*, Vol. 2, Iss. 6, URL: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/http://ijcsn.org/IJCSN-2013/2-6/IJCSN-2013-2-6-153.pdf
16. Langley, A., Hamburg, M. and Turner, S. (2016), "RFC 7748: Elliptic Curves for Security", *IETF,* DOI:https://doi.org/10.17487/RFC7748
17. O'Connor, J., et al. (2020), "BLAKE3: One Function, Fast Everywhere", URL: https://scribd.com/document/691850148/Blake3
18. Dobraunig, C., et al. (2021), "Ascon – Lightweight Authenticated Encryption", *Journal of Cryptology*, Vol. 34, article number 33, URL: https://link.springer.com/article/10.1007/s00145-021-09398-9
19. Erbil, M.E., Bayrakçı, H.C. and Özkahraman, M. (2024), "Endüstriyel Robot Kolların Hibrit Şifreleme Yöntemi ile Veri Güvenliğinin Sağlanması", 2024.
20. Erbil, M.E., Süzen, A.A. and Bayrakçı, H.C. (2023), "Otonom Mobil Robotların Veri Güvenliğinin Hibrit Şifreleme Yöntemi ile Sağlanması", https://doi.org/10.55974/utbd.1311229

21. Stallings, W. (2022), "Cryptography and Network Security", *Pearson*, URL: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://dl.hiva-network.com/Library/security/Cryptography-and-network-security-principles-and-practice.pdf
22. Fysarakis, K. (2013), "Lightweight Cryptography for Embedded Systems – A Comparative Analysis*", Data Privacy Management and Autonomous Spontaneous Security,* https://doi.org/10.1007/978-3-642-54568-9_21
23. Diffie, W., Hellman, M. (1976), "New Directions in Cryptography", *IEEE Transactions on Information Theory*, URL: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://ee.stanford.edu/~hellman/publications/24.pdf
24. Rivest, R.L., Shamir, A. and Adleman, L. (1978), "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", *Communications of the ACM,* Vol. 26, Iss. 1, pp. 96 – 99, https://doi.org/10.1145/357980.358017
25. Daemen, J. and Rijmen, V. (2002), "The Design of Rijndael: AES", *Springer*, DOI:10.1007/978-3-662-04722-4
26. Singh, A., Kaur, V. (2013), "Review of Various Algorithms Used in Hybrid Cryptography", *International Journal of Computer Science and Network*, Vol. 2, Iss. 6, URL: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/http://ijcsn.org/IJCSN-2013/2-6/IJCSN-2013-2-6-153.pdf
27. Bernstein, D.J. and Lange, T. (2006), "Curve25519: Diffie–Hellman Speed Records" *Public Key Cryptography – PKC 2006*, URL: https://link.springer.com/chapter/10.1007/11745853_14
28. Saho, N J. G. and Ezin, E. C. (2020), "Comparative Study on the Performance of Elliptic Curve Cryptography Algorithms with Cryptography through RSA Algorithm", *Proceedings of CARI*, URL: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://hal.science/hal-02926106v1/document
29. Mahto, D. and Yadav, D. K. (2017), "Performance Analysis of RSA and Elliptic Curve Cryptography", *International Journal of Network Security*, Vol.20, No.4, PP.625-635, DOI: 10.6633/IJNS.201807 20(4).04
30. Wang, D., Lin, Y., Hu, J., Zhang, C. and Zhong, Q. (2023), "FPGA Implementation for Elliptic Curve Cryptography Algorithm and Circuit with High Efficiency and Low Delay for IoT Applications", *Micromachines,* Vol.14(5), https://doi.org/10.3390/mi14051037
31. Aumasson J.P Meier, W. and Phan, R. C.-W. (2023), "The Hash Function Family LAKE", URL: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.aumasson.jp/data/papers/AMP08.pdf
32. Zaw T.M., Thant, M. and Bezzateev, S. V. (2019), "Database Security with AES Encryption, Elliptic Curve Encryption and Signature", *IEEE*, DOI: 10.1109/WECONF.2019.8840125
33. Dobraunig, C., et al. (2021), "Ascon v1.2: Lightweight Authenticated Encryption and Hashing", *Journal of Cryptology,* URL: https://link.springer.com/article/10.1007/s00145-021-09398-9
34. Sorescu, T-G., et al. (2025), "Comparative Performance Analysis of Lightweight Cryptographic Algorithms on Resource-Constrained IoT Platforms", *Sensors*, https://doi.org/10.3390/s25185887
35. Peng, S., et al. (2025), "Improved Key Recovery Attacks of Ascon", *IACR Cryptology ePrint Archive*, URL: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://eprint.iacr.org/2025/1029.pdf
36. Zheng, Y., et al. (2024), "Quantum circuit implementations of lightweight authenticated encryption ASCON", *J. Supercomput*, DOI: 10.1007/s11227-023-05877-x
37. Selvi, P. and Sakthivel, S. (2025), "A hybrid ECC-AES encryption framework for secure and efficient cloud-based data protection", *Scientific Reports,* Vol. 15, https://doi.org/10.1038/s41598-025-01315-5
38. Zhang, J., et al. (2019), "High-Speed and High-Security Hybrid AES-ECC Cryptosystem Based on FPGA", *IEEE International Conference on Signal, Information and Data Processing (ICSIDP),* DOI:10.1109/ICSIDP47821.2019.9173457
39. Ganwani, P., et al. (2021), "LSB Based Audio Steganography using RSA and ChaCha20 Encryption", *12th International Conference on Computing Communication and Networking Technologies (ICCCNT),* DOI:10.1109/ICCCNT51525.2021.9580177
40. Hafsa, A., et al. (2017), "A hardware-software co-designed AES-ECC cryptosystem", *International Conference on Advanced Systems and Electric Technologies,* DOI:10.1109/ASET.2017.7983665
41. Saarinen, M-J. O. (2024), "The Quantum Threat to RSA and Elliptic Curve Cryptography", URL: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://mjos.fi/doc/20241209-dosentti-pqc.pdf
42. Nir, Y. and Josefsson, S. (2020), "Curve25519 and Curve448 for the Internet Key Exchange Protocol Version 2 (IKEv2) Key Agreement RFC 8031", *IETF*, URL: https://datatracker.ietf.org/doc/rfc8031/
43. Bellare, M. and Namprempre, C. (2008), "Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm", *Journal of Cryptology*, https://doi.org/10.1007/s00145-008-9026-x
44. Rogaway, P. and Shrimpton, T. (2006), "A Provable-Security Treatment of the Key-Wrap Problem", *EUROCRYPT 2006*, URL: https://link.springer.com/chapter/10.1007/11761679_23
45. Rogaway, P. and Shrimpton, T. (2007), "The SIV Mode of Operation for Deterministic Authenticated-Encryption (Key Wrap) and Misuse-Resistant Nonce-Based Authenticated-Encryption", URL: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://web.cs.ucdavis.edu/~rogaway/papers/siv.pdf
46. Gueron, S., Langley, A. and Lindell, Y. (2019), "AES-GCM-SIV: Nonce Misuse-Resistant Authenticated Encryption", *IRTF*, URL: https://www.tech-invite.com/y80/tinv-ietf-rfc-8452.html
47. Nir, Y. and Langley, A. (2018), "RFC 8439: ChaCha20 and Poly1305 for IETF Protocols (AEAD)", *IRTF*, URL: https://datatracker.ietf.org/doc/rfc8439/
48. Moriarty, K., et al. (2016), "PKCS #1: RSA Cryptography Specifications Version 2.2", *IETF*, URL: https://datatracker.ietf.org/doc/html/rfc8017
49. Böck, H., Zauner, A., Devlin, S., Somorovsky, J. and Jovanovic, P. (2016), "Nonce-Disrespecting Adversaries: Practical Forgery Attacks on GCM in TLS", *USENIX WOOT*, URL: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.usenix.org/system/files/conference/woot16/woot16-paper-bock.pdf

50. S¨onmez, Turan M., McKay, K., Chang, D., Bassham, L.E., Kang, J., Waller, N.D., Kelsey, J.M. and Hong, D. (2023), "Status Report on the Final Round of the NIST Lightweight Cryptography Standardization Process", *National Institute of Standards and Technology, Gaithersburg, MD*, https://doi.org/10.6028/NIST.IR.8454
51. Turan, M. S., et al. (2025), "NIST SP 800-232: Ascon-Based Lightweight Cryptography Standards for Constrained Devices", *National Institute of Standards and Technology, Gaithersburg*, https://doi.org/10.6028/NIST.SP.800-232
52. NIST Selects 'Lightweight Cryptography' Algorithms to Protect Small Devices (2023), URL: https://www.nist.gov/news-events/news/2023/02/nist-selects-lightweight-cryptography-algorithms-protect-small-devices
53. Dobraunig, C., et al. (2019), "Ascon 1.2 – Analysis of Security and Efficiency", URL: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2019/documents/papers/ascon-1-2-analysis-of-security-lwc2019.pdf
54. Dobraunig, C., et al. (2019), "Ascon v1.2 Submission to NIST", URL: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/ascon-spec-round2.pdf
55. Aumasson, J.-P. (2020), "Too Much Crypto", URL: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://rwc.iacr.org/2020/slides/Aumasson.pdf
56. Raspberry Pi Ltd. (2025), "Raspberry Pi 5 8 GB BCM2712 2.4GHZ", URL: https://surl.li/vxzdid
57. Upton, E. (2023), "Introducing: Raspberry Pi 5!", URL: https://www.raspberrypi.com/news/introducing-raspberry-pi-5/
58. Hollingworth, G. (2022), "Raspberry Pi OS (64-bit)", URL: https://www.raspberrypi.com/news/raspberry-pi-os-64-bit/
59. Bozhko, A. (2023), "Properties of AEAD Algorithms", *IRTF*, URL: https://datatracker.ietf.org/doc/draft-irtf-cfrg-aead-properties/01/
60. Bellare, M., Ng, R. and Tackmann, B. (2019), "Nonces Are Noticed: AEAD Revisited", *Cryptology ePrint Archive*, URL: https://ia.cr/2019/624
61. Harkins, D. (2015), "RFC 5297: Synthetic Initialization Vector (SIV) Authenticated Encryption Using the Advanced Encryption Standard (AES)", *IETF*, URL: https://datatracker.ietf.org/doc/rfc5297/
62. McGrew, D. (2008), "RFC 5116: An Interface and Algorithms for AEAD", *RFC Editor, United States,* DOI: https://doi.org/10.17487/RFC5116
63. Langley, A., et al. (2016), "Elliptic Curves for Security", *IRTF*, URL: https://datatracker.ietf.org/doc/html/rfc7748
64. Krentz, K.-F. and Voigt, T. (2024), "More Lightweight, yet Stronger: Revisiting OSCORE's Replay Protection", *Workshop on Security and Privacy in Standardized IoT (SDIoTSec),* https://dx.doi.org/10.14722/sdiotsec.2024.23003
65. Al-Shareeda, M. A., et al. (2022), "Replay-Attack Detection and Prevention Mechanism in Industry 4.0 Landscape for Secure SECS/GEM Communications", *Sustainability*, DOI:10.3390/su142315900
66. Fitzgibbon, G. and Ottaviani, C. (2024), "Constrained Device Performance Benchmarking with the Implementation of Post-Quantum Cryptography", *Cryptography*, Vol. 8(2), https://doi.org/10.3390/cryptography8020021
67. Dewit, W., et al. (2024), "A Preliminary Assessment of the real-time capabilities of Real-Time Linux on Raspberry Pi 5", URL: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://antonio.paolillo.be/publications/workshops/ecrtsOspert2024_dewit_rtlinux_paper.pdf

Received (Надійшла) 06.11.2025
Accepted for publication (Прийнята до друку) 19.11.2025

ВІДОМОСТІ ПРО АВТОРІВ/ ABOUT THE AUTHORS

Ербіл Мустафа Емре – магістр наук, технологічний факультет, кафедра мехатроніки, Іспартанський університет прикладних наук, Іспарта, Туреччина;
Mustafa Emre Erbil – Master of Science, Technology Faculty, Mechatronics Engineering Department, Isparta University of Applied Sciences, Isparta, Turkey;
e-mail: yl2330654001@isparta.edu.tr; ORCID Author ID: https://orcid.org/0009-0003-9394-8588.

Байракджи Гільмі Дженк – доктор філософії, професор, технологічний факультет, кафедра мехатроніки, Іспартанський університет прикладних наук, Іспарта, Туреччина;
Hilmi Cenk Bayrakçı – PhD, Professor, Technology Faculty, Mechatronics Engineering Department, Isparta University of Applied Sciences, Isparta, Turkey;
e-mail: cenkbayrakci@isparta.edu.tr; ORCID Author ID: https://orcid.org/0000-0001-5064-7310;
Scopus ID: https://www.scopus.com/authid/detail.uri?authorId=23134488100.

Озкахраман Мердан – доктор філософії, доцент, технологічний факультет, кафедра мехатроніки, Іспартанський університет прикладних наук, Іспарта, Туреччина;
Merdan Özkahraman – PhD, Associate Professor, Technology Faculty, Mechatronics Engineering Department, Isparta University of Applied Sciences, Isparta, Turkey;
e-mail: merdanozkahraman@isparta.edu.tr; ORCID Author ID: https://orcid.org/0000-0002-3501-6497;
Scopus ID: https://www.scopus.com/authid/detail.uri?authorId=57369259200.

НОВА ГІБРИДНА СХЕМА ШИФРУВАННЯ НА ОСНОВІ СУЧАСНИХ КРИПТОГРАФІЧНИХ КОМПОНЕНТІВ ДЛЯ ВБУДОВАНИХ СИСТЕМ

М.Е. Ербіл, Г.Д. Байракджи, М. Озкахраман

Анотація. Актуальність. Вбудовані системи набувають дедалі більшого значення в різних галузях, таких як автомобілебудування, промислова автоматизація, технології охорони здоров'я та Інтернет речей (IoT). Основні характеристики цих систем включають обмежену пам'ять, помірну обчислювальну потужність та суворі вимоги до

енергоефективності. Ці обмеження роблять безпечну передачу даних важливою, особливо в процесах дротового та бездротового зв'язку. Потенційні витоки даних або несанкціонований доступ становлять серйозні загрози для надійності та безпеки вбудованих систем. **Предметом дослідження** є методи та механізми безпечної передачі даних у вбудованих системах на основі запропонованої гібридної криптографічної архітектури. Головною **метою дослідження** є пропонування надійного та орієнтованого на безпеку гібридного криптографічного рішення, яке задовольняє потребу в конфіденційній та захищеній цілісності передачі даних у таких пристроях. **Результати.** У запропонованому методі для безпечного обміну ключами використовується асиметричний механізм на основі X25519. Функція BLAKE3 використовується для виведення ключа завдяки своїй детермінованій структурі та криптографічній стійкості, тоді як Ascon-128a використовується в рівні симетричного шифрування завдяки своїй легкій, але дуже безпечній конструкції AEAD. Ця комбінація інтегрує надійну можливість обміну ключами X25519, стійке до колізій та високоентропійне виведення ключів BLAKE3, а також захист цілісності та конфіденційності Ascon-128a на основі AEAD (аутентифіковане шифрування з асоційованими даними) в єдиній структурі. Доцільність запропонованої гібридної структури була перевірена на Raspberry Pi 5 з використанням мови програмування Python. В експериментальній установці метод був валідований за різних сценаріїв зв'язку між вбудованим датчиком та контролером. **Висновки.** Результати дослідження показують, що розроблена гібридна архітектура забезпечує високобезпечну, цілісну та сумісну з вимогами прямої секретності альтернативу для захисту даних у вбудованих системах, пропонуючи сильніші криптографічні гарантії порівняно з традиційними гібридними схемами шифрування, описаними в літературі.

Ключові слова: криптологія, кібербезпека, вбудовані системи, Raspberry Pi, гібридне шифрування.