IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151189

An Effective and Optimized Lightweight Encryption Algorithm for Cloud Based IoT

Majoju Kiran Kumar[1], Prof. Arathi Chitla[2]

Research Scholar, Department of Computer science and Engineering, Telangana University, Nizamabad, Telangana, India[1]

Professor Of Computer science and Engineering, Department of Computer science and Engineering, Telangana University, Nizamabad, Telangana, India[2]

Abstract: Cloud-based Internet of Things (IoT) systems transfer sensitive data from resource-constrained devices through gateways into untrusted cloud storage. However, no single lightweight cipher is optimal for every device, data type, or threat level. Published benchmarks demonstrate that lightweight cipher rankings vary depending on the platform, payload size, and the metric being optimized. Recent reviews highlight the absence of a standardized, evidence-based method for selecting among these ciphers. This work introduces OLEF-CIoT, an Optimized Lightweight Encryption Framework for Cloud-based IoT. OLEF-CIoT (i) profiles the device, data sensitivity, and application quality-of-service requirements; (ii) eliminates candidate ciphers that do not meet strict security and resource constraints; (iii) ranks the remaining candidates using a normalized, weighted multi-criteria utility function across execution time, throughput, memory, energy, hardware area, and security strength; and (iv) integrates the selected cipher into a layered device–edge–cloud security pipeline, combining authenticated lightweight symmetric encryption at the device, elliptic-curve key agreement and dual encryption at the edge, and pairing-free attribute-based access control in the cloud. An adaptive feedback loop re-executes the selection process when battery, latency, or threat conditions change. The framework is evaluated using published hardware (UMC 90 nm) and software (8-bit ATmega328) measurements of PRESENT, AES-128, SPECK, and ISL-LWC. Case studies indicate that the optimal cipher depends on the deployment profile: PRESENT-128 is preferred for area- and power-constrained hardware, SPECK for latency-critical software, and a low-cost key schedule is advantageous when frequent re-keying is required. The paper concludes with a security analysis of the integrated pipeline and outlines a research agenda for standardized benchmarking.

Keywords: Internet of Things, cloud computing, lightweight cryptography, block ciphers, multi-criteria optimization, attribute-based encryption, elliptic curve cryptography, ASCON, PRESENT, SPECK.

I. INTRODUCTION

The IoT connects sensors, actuators, RFID tags, wearables, and embedded controllers to cloud platforms that store, analyse, and share the data they generate. Security is consistently the top issue in IoT deployments because most devices are physically exposed and have very limited energy, memory, processing power, and silicon area [1]. The traditional primitives (AES, RSA and SHA-256) were all designed for platforms with lots of resources and are typically too heavy for the embedded nodes and sensor networks in question [2], [3]. Lightweight cryptography (LWC) addresses this problem with ciphers that sacrifice some security or generality to achieve small footprints, low power, and low latency [1], [2].

Today, the difficulty is no longer a shortage of lightweight ciphers but choosing among them. Thakor *et al.* report more than fifty lightweight ciphers in use, plus 57 submissions to the NIST lightweight cryptography process, and conclude that none satisfies every hardware and software metric; each performs best only in its own target environment [1]. A 2025 systematic review of 22 recent proposals identifies the "profound lack of standardized frameworks for both performance benchmarking and security validation" as its most critical finding, because methodological fragmentation makes evidence-based algorithm selection hard [4]. Empirical studies confirm that rankings vary by platform and payload: on a Raspberry Pi 3, XTEA and SIMON lead in throughput for small payloads, while on an 8-bit Arduino Mega 2560, Skipjack and HIGHT lead and PRESENT ranks last in software [5]. On Arduino Nano and Micro boards, SPECK gives the best throughput and latency, while ASCON gives the best energy efficiency and memory use [6]. In hardware, by contrast, PRESENT-128 needs only 2272 gate equivalents (GE) and 0.98 µW/MHz at 90 nm, far below AES-128 [7].

Cloud integration adds another layer of requirements. Data leaving the device must remain confidential at the gateway and in the untrusted cloud. Keys must be agreed and refreshed without expensive public-key operations on the device, and stored data must be shared only with the appropriate users. Some of the prior work addresses this: lightweight block ciphers [8], domain-specific selection advice for healthcare [9], elliptic curve dual encryption for cloud based IoMT [10],

DOI: 10.17148/IJARCCE.2026.151189

pairing free ABE for IoT [11], a hybrid AES-ECC-ABE framework for body area networks [12] and lightweight image cryptosystems for cloud assisted IoT [13], [14]. Most of this work is evaluated in isolation, using different metrics and platforms, and none provides a repeatable process that takes a given device and data profile, maps it to the optimal cipher, and then integrates that cipher end-to-end with the cloud.

A. Research Objective

In this work we aim to develop a framework or model to identify an optimized, effective lightweight encryption algorithm for cloud-based IoT and integrate it into the device–edge–cloud data path. In this context, effectiveness means meeting a specified security threshold based on data sensitivity, while optimization maximizes a weighted utility across measurable performance and cost metrics, subject to the device's resource constraints.

B. Contributions

1. A consolidated taxonomy and comparison of lightweight ciphers, hybrid schemes, attribute-based encryption (ABE) schemes, and image cryptosystems, synthesized from fourteen reference studies and organized by the metrics reported in each (Section II).
2. OLEF-CIoT, a four-stage framework comprising profiling, constraint filtering, multi-criteria optimization, and adaptive re-selection, supported by a formal utility model and selection algorithm (Section III).
3. An integration architecture that embeds the selected cipher within a layered pipeline: authenticated lightweight encryption at the device, elliptic-curve cryptography (ECC)-based key agreement and dual encryption at the edge, and pairing-free attribute-based encryption (ABE) in the cloud (Section IV).
4. Two case studies applying the model to published hardware and software measurements, demonstrating how the optimal cipher selection varies with the deployment profile (Section V), followed by a security analysis (Section VI) and a discussion of limitations and future work (Section VII).

C. Paper Organization

Section II reviews our related work. Section III presents the OLEF-CIoT framework and its optimization model. Section IV describes the optimized encryption scheme and its integration with the cloud. Section V reports the evaluation. Section VI provides a security analysis. Section VII discusses limitations and future directions, and Section VIII concludes.

II. RELATED WORK

The reference literature is divided into five groups: surveys of lightweight primitives, empirical benchmarks, new cipher designs, hybrid and cloud-oriented schemes and image cryptosystems. Table I provides a summary of each of the studies and the gap each one leaves in terms of an integrated selection framework.

A. Surveys of Lightweight Cryptography

Singh *et al.* [2] survey lightweight block ciphers, stream ciphers and hash functions, and compare them by key size, block size, number of rounds and structure (SPN, Feistel, ARX). They also propose a smart-home security scheme in which each device parameter has a threshold, and name an algorithm for computing those thresholds as future work. That idea—selecting security behavior from device parameters—is the starting point for the profiling stage of OLEF-CIoT. Shah *et al.* [3] describe a four-stage IoT architecture (sensors and actuators, gateways and data acquisition, edge IT, and data centre/cloud) and compare AES, EC-DH and related methods, recommending a combination of AES and NTRU for stronger security. Thakor *et al.* [1] compare lightweight ciphers by implementation cost, hardware and software performance, and resistance to attacks. They observe that PRESENT and CLEFIA are standardized for their security with acceptable cost, that SIMON and SPECK have the most compact implementations, and that three open problems remain: reducing S-box cost, lightening key schedules, and choosing the number of rounds without losing security. Silva *et al.* [4] review 22 recent ciphers (e.g., SPNRX, MBRISI, QLW, GFSPX), find that execution speed, memory and energy are the dominant performance criteria, and that security is usually assessed with differential and linear cryptanalysis, the avalanche effect, histogram analysis and NIST statistical tests. Their central conclusion is that there is no standard evaluation framework.

B. Empirical Benchmarks

Panahi *et al.* [5] measure ten block ciphers (AES, PRESENT, LBlock, Skipjack, SIMON, XTEA, PRINCE, Piccolo, HIGHT, RECTANGLE) on a Raspberry Pi 3 and an Arduino Mega 2560 for RAM, ROM, energy, throughput and

DOI: 10.17148/IJARCCE.2026.151189

execution time with the data being sent to the cloud. The order of the ciphers is different on the two devices and varies with payload size (from 16B to 2048B). Radhakrishnan *et al.* [6] benchmark AES-128, SPECK and ASCON on Arduino Nano and Micro boards. SPECK has the highest throughput and lowest latency, AES-128 has the best key scheduling throughput and ASCON has the best energy efficiency, the lowest memory requirement and the best resistance to attacks. Goyal *et al.* [7] implement PRESENT, AES, ECDH, DH and RSA in a UMC 90nm standard cell library. The PRESENT-128 design uses 2272 GE, achieves 1.86 Gb/s and consumes 0.98 µW/MHz as compared to 3900 GE, 0.23 Gb/s and 62 µW/MHz for the AES-128 design that they compare it to. PRESENT is also standardised in ISO/IEC 29192-2:2012.

C. New Lightweight Cipher Designs

Khompysh *et al.* [8] present ISL-LWC, a 16 round SPN cipher with a 64 bit block and an 80 bit key. It passes both the NIST and Knuth statistical test suites and has a good avalanche effect. On an Arduino Uno R3 (ATmega328, 16 MHz) it encrypts a block in 108.59 µs (faster than PRESENT at 2111.56 µs, slower than SPECK at 16.90 µs) and sets up its round keys in 275.12 µs (roughly 5-6x faster than both). Oladipupo *et al.* [13] get rid of the S-box altogether with their HXP cryptosystem which uses SHA-256, XOR and bit permutation within a Feistel structure. Unlike ARX designs which typically diffuse half the block per round, HXP diffuses the entire block in one round. The authors note that the memory, time and power costs have not yet been measured.

D. Hybrid, Dual and Attribute-Based Schemes for Cloud IoT

Kumar *et al.* [10] present Elliptic Curve and Dual Cryptography (ECDC) for cloud-based IoMT, in which the cloud server and hospital server agree on a curve and a second key layer strengthens ECC. Their scheme requires fewer elliptic-curve operations than four compared schemes for 160-, 192-, and 256-bit keys. Yao *et al.* [11] propose a pairing-free key-policy ABE (KP-ABE) scheme built on ECC, with security reduced to the elliptic-curve decisional Diffie–Hellman (ECDDH) assumption in the selective-set model. It reduces data transmission and processing cost relative to pairing-based ABE, but lacks attribute revocation, expands linearly with the number of attributes, and supports only a single authority. Karunkuzhali *et al.* [12] combine AES for data, ECC for key management and ABE for dynamic access control (HLCA-ABE) in wireless body-area sensor networks. They tune a deep-learning monitoring model with metaheuristic optimizers and report 99.9% accuracy, while noting that combining AES, ECC and ABE may raise computing load at scale. Sabri *et al.* [9] map ciphers to healthcare data sensitivity: AES-128, LEA, ASCON and GIFT for high-sensitive data, HIGHT and PRINCE for medium, and RC5-32/12/16 only for low-risk data.

E. Lightweight Image Encryption

Images are a significant proportion of the IoT data that ends up in the cloud. Jain *et al.* [14] combine the Hénon and 2D-logistic chaotic maps (MMCBIE) and report an NPCR of 99.603% and an UACI of 32.8828%, with the caveat that the computing complexity may make it unsuitable for real time use on resource constrained devices. HXP [13] looks at the same cloud based image scenario but with an S-box free design. The upshot of all this is that the best primitive is very much dependent on the data type in question, a parameter which is rarely included in general cipher benchmarking.

TABLE I. Summary of reference studies and the gap each leaves for an integrated selection framework

Ref.	Focus	Algorithms studied	Platform / metrics	Key finding	Gap
[1]	Review and comparison	50+ ciphers, NIST LWC candidates	GE, throughput, attacks	No cipher wins on all metrics	No selection procedure
[2]	Survey and scheme	Block, stream, hash primitives	Key/block size, rounds	Threshold-based smart-home scheme	Threshold algorithm left open
[3]	Study	AES, EC-DH, SDE	Qualitative	Suggests AES + NTRU	No quantitative model
[4]	Systematic review	22 recent ciphers	Speed, memory, energy	No standard evaluation framework	Calls for unified benchmarks
[5]	Benchmark	10 block ciphers	RPi 3, Arduino Mega; RAM, ROM, energy, time	Ranking depends on	No integrated decision rule

DOI: 10.17148/IJARCCE.2026.151189

				device and payload	
[6]	Benchmark	AES-128, SPECK, ASCON	Arduino Nano/Micro; time, memory, latency	SPECK fastest; ASCON most efficient	Three ciphers only
[7]	Hardware	PRESENT, AES, ECDH, DH, RSA	UMC 90 nm; GE, power, Gb/s	PRESENT-128 best area/power	Hardware only
[8]	New cipher	ISL-LWC vs PRESENT, SPECK	Arduino Uno; time, NIST tests	Fast key schedule	80-bit key; cryptanalysis pending
[9]	Healthcare review	AES, LEA, ASCON, GIFT, HIGHT, PRINCE, RC5	Speed, throughput, security level	Sensitivity-based mapping	Qualitative tiers
[10]	Dual encryption	ECC + dual cryptography	ECC operation counts	Fewer EC operations	IoMT-specific
[11]	ABE	Pairing-free ECC KP-ABE	Communication/computation cost	Lightweight access control	No revocation; single authority
[12]	Hybrid framework	AES + ECC + ABE	WBASN simulation	99.9% monitoring accuracy	Overhead at scale
[13]	Image cryptosystem	HXP (S-box-free)	Entropy, NPCR/UACI, EQ	S-box removal keeps security	Cost not yet measured
[14]	Image encryption	MMCBIE (chaotic maps)	NPCR 99.603%, UACI 32.88%	Strong statistical security	High computational cost

III. PROPOSED OLEF-CIOT FRAMEWORK

OLEF-CIoT turns the problem of cipher selection into a constrained multi-criteria optimisation problem which is solved on a per-deployment-profile basis and re-solved when the situation changes. It comprises four stages: (1) profiling, (2) constraint filtering, (3) multi-criteria optimisation and (4) integration with adaptive re-selection. Figure 1 illustrates the way the stages relate to the device, edge and cloud layers.

Fig. 1. OLEF-CIoT architecture: four-stage selector and the device–edge–cloud data path

The selector runs off-device; it assigns the chosen cipher a† to the device layer and re-runs when gateway telemetry signals a trigger.

A. Stage 1: Profiling

Each deployment is defined by three profiles. The device profile details the platform (software on a microcontroller or hardware ASIC/FPGA), word size, clock frequency, available RAM and ROM and energy budget. The data profile details the data type (scalar telemetry, stream or image) and its sensitivity tier (high, medium or low) as per the healthcare mapping in [9]. The quality of service (QoS) profile details the typical payload size, the maximum tolerable latency and the expected re-keying frequency. Payload size is important as the cipher rankings vary between 16 byte and 2048 byte payloads [5].

$$D = (\tau, w, f_{dk}, R_{RAM}, R_{ROM}, E_{max}), \quad S = (\delta, \ell), \qquad Q = (p, L_{max}, \rho)$$

Here τ is the platform type, w the word size, δ the data type, $\ell \in$ {high, medium, low} the sensitivity tier, p the payload size, L_{max} the latency bound and ρ the re-keying rate.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151189

B. Stage 2: Constraint Filtering

The A candidate pool consists of the standard, well-understood ciphers (AES-128, PRESENT, SPECK, SIMON, ASCON, GIFT, LEA, HIGHT, PRINCE, RECTANGLE, XTEA, Skipjack, LBlock, Piccolo) plus some domain specific designs such as ISL-LWC [8], and for image data HXP [13] and MMCBIE [14]. A candidate 'a' is only admissible if it satisfies all the hard constraints:

$$A^* = \{a \in \mathcal{A}: \sigma(a) \geq \lambda_l, M_{RAM(a)} \leq R_{RAM}, M_{ROM(a)} \leq R_{ROM}, t(a,p) \leq L_{max}, AE(a) \geq \alpha_l\}$$

The security strength σ(a) is the base-2 logarithm of the complexity of the best-known attack, as used for the PRESENT, AES and ECDH comparison in [7]. The thresholds are λ_high = 128 bits with authenticated encryption required (α_high = 1), λ_medium = 128 bits with integrity provided by the protocol, and λ_low = 80 bits. A cipher with a published practical attack, or one that has not passed statistical randomness and avalanche tests [4], [8], is excluded at this stage regardless of its performance. Filtering before scoring makes certain that no amount of speed can buy back an inadequate security level.

C. Stage 3: Multi-Criteria Optimization

For each admissible candidate OLEF-CIoT collects a metric vector x_a = (x_a1,..., x_am) on the target platform (execution time, throughput, RAM, ROM, energy per byte, key setup time and for hardware area in GE and power per MHz). The cost metrics (time, memory, energy, area, power) and the benefit metrics (throughput, security strength) are all scaled to [1] via min–max scaling over the admissible set:

$$n_{aj} = \begin{cases} \dfrac{x_{aj} - min_b x_{bj}}{max_b x_{bj} - min_b x_{bj}}, \text{j is a benefit metric} \\ \dfrac{max_b x_{bj} - x_{aj}}{max_b x_{bj} - min_b x_{bj}}, \text{j is a cost metric} \end{cases}$$

Dominated candidates (i.e. those no better on any metric and worse on at least one than another candidate) are eliminated to give the Pareto set P. The remaining candidates are then ranked according to a weighted utility and the optimal cipher is the maximiser:

$$U(a) = \sum_{j=1}^{m} w_j \, n_{aj}, \quad \sum_{j=1}^{m} w_j = 1, \quad a^{\dagger} = arg\, max_{a \in p}\, U(a)$$

The weights represent the deployment's priorities. OLEF-CIoT provides default weight templates for the standard profiles (Table II); an architect can of course override these or derive them via pairwise comparison. A sensitivity analysis is carried out for every selection (each weight is perturbed by ±0.1 and the effect on a† is checked) and the results of this are reported so that a fragile choice is apparent.

TABLE II. Default weight templates for common cloud-IoT deployment profiles

Profile	Example	Time	Throughput	Memory	Energy / power	Area	Security
P1 Battery sensor	Environmental node, 8-bit MCU	0.15	0.05	0.20	0.30	—	0.30
P2 Real-time	Actuator control, video gateway	0.35	0.25	0.10	0.10	—	0.20
P3 Hardware tag	RFID, ASIC sensor	—	0.10	—	0.30	0.30	0.30
P4 High-sensitivity	Medical wearable, IoMT [9], [10]	0.15	0.05	0.15	0.20	—	0.45

D. Stage 4: Integration and Adaptive Re-selection

The chosen cipher is part of the layered pipeline as described in Section IV. A monitoring agent on the gateway will be looking at battery level, observed latency, payload mix and threat intelligence (e.g. a new attack on a candidate has just been published). A re-selection will be triggered if the battery level drops below threshold θ_E, if the latency bound is breached for a sustained period, or if the threat level or sensitivity tier changes. To prevent oscillation between ciphers which are very close in utility a re-selection will only be made if the new utility is at least Δ greater than the current one:



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.151189

$$"switch\ to\ "\ a' \ \Leftrightarrow\ U(a') - U(a_(cur)) > \Delta\ "\ and\ "\ a' \in A^*$$

This takes the device threshold idea of Singh *et al.* [2] and which lead us turn it into a quantitative, repeatable decision rule. Based on required weight templates which can help to decide the suitable cipher, we proposed the following algorithm.

E. Selection Algorithm

Algorithm 1: OLEF-CIoT cipher selection

1. Input: device profile D, data profile S, QoS profile Q, candidate pool A, weight vector w, margin Δ.
2. Set the security floor λ and authenticated-encryption requirement α from the sensitivity tier ℓ.
3. For each a in A, look up or measure x_a on the target platform for payload size p.
4. Form A* by removing every candidate that violates a hard constraint; if A* is empty, report the binding constraint and recommend a higher device class or offloading to the edge.
5. Normalize the metrics over A*, remove dominated candidates to form P, and compute U(a) for each a in P.
6. Return a† = argmax U(a), its runner-up, and the weight-sensitivity report.
7. At run time, on any trigger event, repeat steps 2–6 and switch only if the utility gain exceeds Δ.

The cost of Algorithm 1 is O(|A| · m) for filtering and scoring plus O(|A|² · m) for Pareto filtering. With fewer than twenty candidates and about eight metrics, this runs at the gateway or in the cloud in negligible time; constrained devices never run it themselves.

IV. OPTIMIZED LIGHTWEIGHT ENCRYPTION SCHEME AND CLOUD INTEGRATION

The cipher in question is only effective if it is part of a data path that also deals with keys, integrity and access control without the device having to do any of that work. OLEF-CIoT therefore places each cryptographic function at the layer that is best positioned to do it: device level symmetric lightweight encryption, elliptic curve at the edge and attribute based access control in the cloud. This is a direct application of the four stage IoT architecture of [3] and the hybrid patterns of [10]-[12].

A. Device Layer: Authenticated Lightweight Encryption

The device encrypts each reading or frame with the selected cipher (a†) under a session key K_s. For very sensitive data, Stage 2 already mandates authenticated encryption: ASCON provides this natively [6], [9]; a block cipher such as PRESENT-128 or SPECK-64/128 would be used in an encrypt-then-MAC construction. Each packet has a small header containing the cipher ID, a key epoch and a monotonic counter which is used as the nonce:

$$C_i = AEnc_{a^\dagger}\left(K_s, N_i = ctr_i, H_i = \left(id_{a^\dagger}, e, dev\right), M_i\right)$$

The cipher identifier provides the system with crypto-agility: on re-selection the gateway knows which algorithm decrypts each packet. The device does not perform any public key operations at all during normal operation so the per-packet cost remains at the level measured in the benchmarks [5], [6].

B. Edge Layer: Key Agreement and Dual Encryption

The gateway (edge node) does all the public key work. At provisioning and at every key epoch the device and gateway perform an ECC key exchange (ECDH) followed by a key derivation function to produce K_s. Devices that are constrained in terms of compute (i.e. can't afford a full scalar multiplication on every epoch) will have used a pre-shared root key at manufacture which is then ratcheted forward via hashing. ECC is used rather than RSA as it provides the same level of security with significantly shorter keys (a 256 bit ECC key versus a 3072 bit RSA key for 128 bits of security) [7].

The gateway then applies a second outer layer of encryption prior to upload, in accordance with the ECDC dual encryption principle [10]. Two modes are available:



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.151189

1. End to end mode (high sensitivity). The gateway validates the device tag but does not decrypt. It wraps C_i with a fresh data encryption key (DEK) and uploads the doubly encrypted object. A compromised gateway therefore exposes nothing in the clear.
2. Aggregation mode (medium and low sensitivity). The gateway decrypts, filters and aggregates the readings (or in the case of images applies an S-box free image cryptosystem such as HXP [13]) and then encrypts the aggregate under the DEK. This reduces cloud traffic at the expense of trusting the gateway.

$$O = \left(AEnc\left(DEK, C_i \text{“ or “ } Agg(M)\right), ABE.Enc(PK_{ABE}, DEK, \gamma)\right)$$

The DEK itself is never stored in the clear. It is encrypted under an access structure as described below.

C. Cloud Layer: Pairing-Free Attribute-Based Access Control

The cloud stores O but cannot read it. The DEK is encrypted using the pairing-free, ECC-based KP-ABE scheme of Yao *et al.* [11] under an attribute set γ (e.g. {cardiology, hospital-A, 2026-Q4}). A data user (a doctor, an analytics service or a maintenance engineer) obtains from the attribute authority a private key which embodies an access policy; only a user whose policy is satisfied by γ will be able to recover the DEK and hence the data. The use of a pairing-free scheme keeps the encryption cost manageable at the gateway. HLCA-ABE [12] demonstrates the benefit of this level of fine-grained, multi-user access control in a healthcare context. Since [11] does not support attribute revocation, OLEF-CIoT achieves revocation by rotating the DEK at each key epoch and re-encrypting it under the updated attributes so that a revoked user will be unable to read any data written after the rotation.

D. Key Hierarchy

TABLE III. Key hierarchy of the integrated pipeline

Key	Holder	Established by	Protects	Lifetime
Root key K_root	Device, gateway	Provisioning (pre-shared or ECDH)	Derivation of session keys	Device lifetime
Session key K_s	Device, gateway	ECDH + KDF, or hash ratchet from K_root	Device-to-gateway packets	One key epoch
Data-encryption key DEK	Gateway, authorized users	Random, per epoch	Objects stored in the cloud	One key epoch
ABE public key PK_ABE	Gateway (public)	Attribute authority	Encryption of DEK under γ	Until system re-key
ABE user key SK_U	Data user	Attribute authority	Decryption of DEK if policy holds	Until revoked

E. Integration Workflow

1. Provision: the device is given K_root and the gateway's public key; the OLEF-CIoT selector (Algorithm 1) runs at the gateway or in the cloud and selects a†.
2. Sense and encrypt: the device encrypts and authenticates each payload under K_s with a†.
3. Verify and wrap: the gateway checks the tag and counter and then applies the outer layer in either end-to-end or aggregation mode.
4. Protect the key: the gateway encrypts the epoch's DEK under the attribute set γ with pairing-free ABE.
5. Store: the cloud stores O and its metadata; it never stores K_s (the DEK) or the plaintext.
6. Access: an authorized user downloads O, recovers the DEK with SK_U, and decrypts.
7. Adapt: the gateway monitors the triggers and if Algorithm 1 selects a new cipher then it pushes the new identifier with the next key epoch.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151189

V. EXPERIMENTAL SETUP AND PERFORMANCE EVALUATION

The framework was evaluated by applying Algorithm 1 to measurements published in the reference studies, so that every input number can be traced to its source. This isolates the behaviour of the selection model from implementation differences. Section V-E specifies the testbed protocol for the follow-up experimental phase of this research.

A. Evaluation Metrics

The metrics used are those most commonly found in the literature [4]–[6]: execution time per block (µs), key setup time (µs), throughput (B/s or Gb/s), RAM and ROM (bytes), energy per byte (µJ/B), hardware area (GE), power (µW/MHz) and security strength σ (bits). Security is also assessed using the NIST statistical suite, the avalanche criterion and for images specifically, entropy, NPCR and UACI [13], [14][15].

B. Case Study A: Hardware Implementation (Profile P3)

Table IV applies the P3 weights (area 0.30, power 0.30, throughput 0.10, security 0.30) to the 90nm hardware results of Goyal *et al.* [7]. ECDH is not a bulk cipher and is used for key agreement in the edge layer so is not scored. Note that the AES-128 design cited in [7] was implemented in 130nm technology (which somewhat inflates its area) and the comparison is as published.

TABLE IV. OLEF-CIoT applied to published hardware results (P3 weights; data from [7])

Cipher	Area (GE)	Throughput (Gb/s)	Power (µW/MHz)	σ (bits)	U, low tier (λ = 80)	Admissible, high tier (λ = 128)
PRESENT-128	2272	1.86	0.98	128	0.946	Yes; Pareto-optimal
PRESENT-80	1921	1.86	0.89	80	0.700	No (σ < 128)
AES-128	3900	0.23	62	128	0.300	Yes; dominated by PRESENT-128

In the low tier PRESENT-128 wins as it only gives up 18% of PRESENT-80's normalised area score and takes the full security score. In the high tier PRESENT-80 is filtered out and AES-128 is dominant on all metrics so PRESENT-128 is the choice for any set of weights. We have observed that this is consistent with the conclusion of [7] that PRESENT-128 provides the best trade-off between area, power, throughput and security of all the designs considered.

C. Case Study B: Software on an 8-bit Microcontroller (Profiles P2 and P1)

Table V applies two weight vectors to the Arduino Uno R3 (ATmega328, 16MHz) measurements of Khompysh *et al.* [8]. As the cryptanalysis results are not the same for the three ciphers, key length is used as a proxy for sigma. Profile P2 (latency critical) gives weights of 0.5 to encryption time, 0.2 to key setup time and 0.3 to key length. A re-keying heavy variant of P1 gives the weights 0.3, 0.5 and 0.2 respectively.

TABLE V. OLEF-CIoT applied to published 8-bit software results (data from [8])

Cipher	Key (bits)	Encryption time (µs)	Key setup (µs)	U, latency-critical	U, frequent re-keying
SPECK	96	16.90	1320.69	0.835	0.587
ISL-LWC	80	108.59	275.12	0.678	0.787
PRESENT	80	2111.56	1541.31	0.000	0.000

The optimal cipher changes with the profile. SPECK wins when per-packet latency dominates, consistent with [6], whereas ISL-LWC wins when keys are refreshed often, because its key schedule is about 4.8 times faster than SPECK's. PRESENT is dominated in this software setting, consistent with its last-place throughput ranking on the Arduino Mega in [5], even though it is the best hardware choice in Case Study A. The two utilities cross when the key-setup weight satisfies 0.826 w_ks > 0.044 w_enc + w_key, which gives the architect an explicit, auditable rule.

DOI: 10.17148/IJARCCE.2026.151189

Case Study B also illustrates the value of Stage 2. In the high sensitivity tier all three of the configurations tested fail the 128-bit floor so Algorithm 1 returns an empty admissible set and names the binding constraint. The recommended action is to evaluate the 128-bit key variants (SPECK-64/128, PRESENT-128) or ASCON which [6] and [9] both identify as good, efficient options for sensitive data.

D. Profile-Level Recommendations

Combining the evidence from [5] to [9], [13] and [14] with the framework yields the initial recommendations in Table VI. These are defaults which will be confirmed by Algorithm 1 on the actual hardware, not hard and fast rules.

TABLE VI. Starting recommendations produced by OLEF-CIoT for typical cloud-IoT profiles

Profile	Sensitivity	Device-layer cipher	Edge / cloud integration	Evidence
Medical wearable, IoMT	High	ASCON-128 (AEAD)	End-to-end dual encryption, ECC key agreement, pairing-free ABE	[6], [9]–[12]
Real-time control on 8/16-bit MCU	Medium	SPECK-64/128 + MAC	Aggregation mode, ECDH at gateway	[6], [8]
RFID tag, ASIC sensor	Medium–high	PRESENT-128	Gateway re-encryption	[1], [7]
Frequent re-keying, 8-bit MCU	Low–medium	Cipher with low key-setup cost (e.g., ISL-LWC, pending further cryptanalysis)	Hash-ratchet session keys	[8]
Raspberry Pi–class gateway	Medium	XTEA, SIMON or RECTANGLE ranked by payload size	ABE-protected DEK in cloud	[5], [11]
Camera / image node	Medium–high	S-box-free image cryptosystem (HXP) or chaotic scheme (MMCBIE) if compute allows	Encrypted images stored in cloud under ABE	[13], [14]

E. Testbed Protocol for Experimental Validation

The next stage of this work will be to measure all the candidates under a single protocol, fulfilling the requirement for a standardised evaluation as per [4]. The testbed we are going to use will be an Arduino Uno/Mega (8 bit AVR), an ARM Cortex-M class board, a Raspberry Pi gateway and a cloud object store. All ciphers will be compiled with the same compiler and optimisation level, measured over payloads of 16B to 2048B as in [5], with energy being measured via a shunt resistor power monitor and memory obtained from the linker map. The end to end latency from device to cloud, the gateway CPU load for the ECDH and ABE operations and the overhead of re-selection will all be reported in addition to the per cipher metrics.

VI. SECURITY ANALYSIS

The integrated pipeline is assessed against an adversary that can: eavesdrop on and modify the wireless traffic, compromise the cloud provider's storage, capture individual devices and collude with users who possess some of the attribute keys. Table VII shows the mapping of each threat to the mechanism that mitigates it.

A. Confidentiality

On the device link confidentiality is based on the cipher that is selected in Stage 2 which will only accept a cipher if its security strength is at the floor for the data's level. For highly sensitive data the floor is 128 bits (equivalent to AES-128 [7]). In the cloud the DEK protects the data and is encrypted under pairing-free KP-ABE whose security reduces to the ECDDH assumption in the selective-set model [11]. An honest-but-curious cloud provider therefore only sees ciphertext. In end-to-end mode the device layer ciphertext remains in the outer layer so even if a gateway is compromised it learns nothing about the high sensitivity of the payloads in question and the dual encryption protection of [10] is extended.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151189

B. Integrity and Authenticity

Authenticated encryption on the device link will detect any modification of the payload or header (including an attempt to rewrite the cipher identifier in order to force a weaker cipher). The cipher identifier is in the authenticated header and only the permitted ciphers can be selected so a downgrade attack would require a tag forge. The outer layer at the edge is also authenticated so any object which has been tampered with in storage will be rejected on download.

C. Replay and Key Compromise

The monotonic counter used as the nonce allows the gateway to reject replayed packets. Session keys are constrained to a single key epoch (with ECDH based refresh the compromise of one session key does not expose the earlier epochs). With the hash ratchet option for the most constrained devices, you can't recompute an earlier key from a later one, but if the root key is captured then all future keys are exposed until the device is re-provisioned. That trade-off is made visible to the architect at profiling time.

D. Access Control and Collusion

Only users whose policy is satisfied by the stored attribute set will be able to recover the DEK. Collusion resistance (i.e. users cannot collude to satisfy a policy which no individual user can satisfy on their own) is inherited from the ABE construction [11]. There is no direct revocation in [11] so OLEF-CIoT achieves revocation via DEK rotation at each epoch; data written prior to revocation will remain readable by a user who had downloaded it previously. That's the normal limit of any revocation scheme.

E. Statistical and Cryptanalytic Screening

Stage 2 rejects any candidate that fails the NIST and Knuth statistical tests or the avalanche criterion [8], or that has a published practical attack. Thakor *et al.* point out that as new ciphers are proposed new attacks are continually being found [1]; the adaptive loop in Stage 4 allows the system to remove a cipher from the pipeline when such an attack is published, without the need to redesign the pipeline. For image data the selected cryptosystem must achieve an entropy level very close to 8 bits per pixel and NPCR and UACI values very close to the ideal (approximately 99.6% and 33.4% respectively) as demonstrated by MMCBIE [14] and HXP [13].

TABLE VII. Threats and countermeasures in the OLEF-CIoT pipeline

Threat	Countermeasure	Layer
Eavesdropping on the device link	Admissible lightweight cipher, $\sigma \geq \lambda_\ell$	Device
Packet modification or forgery	Authenticated encryption (ASCON or encrypt-then-MAC)	Device
Cipher downgrade	Cipher identifier inside authenticated header; admissible set only	Device, edge
Replay	Monotonic counter as nonce	Edge
Gateway compromise	End-to-end dual encryption for high-sensitivity data	Edge
Curious or breached cloud	DEK under pairing-free ABE; cloud holds no keys	Cloud
Colluding users	Collusion-resistant ABE keys	Cloud
Revoked user	DEK rotation each key epoch	Edge, cloud
Newly published attack	Adaptive re-selection (Stage 4)	All



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.151189

VII. DISCUSSION, LIMITATIONS AND FUTURE WORK

The case studies validate the paper's main point: the 'best' lightweight cipher is a function of the deployment scenario rather than the cipher itself. PRESENT-128 is optimal in 90nm hardware but outperformed in 8-bit software. SPECK is optimal in terms of latency but not for the scenario of frequent re-keying. A framework which makes the profile, the constraints and the weights explicit turns these disparate findings into a repeatable decision process (which is exactly what [4] points out is missing).

The design also addresses the open issues listed in [1]. Lighter key schedules and fewer S-boxes are directly rewarded in the key setup and memory metrics so ISL-LWC [8] or the S-box free HXP [13] are the designs that emerge in the regions where those are important. Putting the ECC and ABE at the edge and in the cloud rather than on the device is the way to deal with the overhead issue raised for HLCA-ABE [12] and still maintains the fine grained access control.

The work has four drawbacks.

1. The evaluation makes use of published measurements (taken from a variety of platforms, compilers and technology nodes) so the AES-128 hardware figure for example is from a 130nm design. The cross-study numbers are therefore indicative and the planned unified testbed (Section V-E) is required to validate the rankings.
2. Weighted sum utility is transparent but very sensitive to normalisation when the candidate set is small (with just two candidates min-max scaling will map every metric to either 0 or 1). Larger candidate sets should be considered as should alternative ranking methods such as TOPSIS.
3. Security strength is reduced down to a single figure (sigma) in Case Study B to key length. A more comprehensive security score which incorporates differential and linear bounds, side channel resistance and the maturity of the public cryptanalysis would be a much better indicator in Stage 2.
4. The ABE scheme that is pairing free (taken from [11]) has a single attribute authority and scales linearly with the number of attributes. Any cross domain deployments would require a lightweight multi-authority scheme.

The next phase of work will be to (i) implement the full pipeline on the testbed of Section V-E and report the end-to-end delay and energy; (ii) add the NIST-standardised ASCON variants and post-quantum KEMs for the edge layer as recommended in [4] and [9]; (iii) learn the weight vectors from operator feedback; and (iv) extend the image path with a cost measured comparison of HXP and MMCBIE on the camera nodes.

VIII. CONCLUSION

This paper presents OLEF-CIoT, a framework for identifying an optimal, effective lightweight encryption algorithm for cloud-based IoT and for integrating it end-to-end. The framework takes the device, data and QoS requirements into account, filters the candidate ciphers against strict security and resource constraints, ranks the survivors using a normalised multi-criteria utility and re-selects when the situation changes. The selected cipher is then embedded into a layered pipeline comprising authenticated lightweight encryption on the device, ECC key agreement and dual encryption at the edge and pairing-free attribute based access control in the cloud. When applied to published hardware and software measurements the model selects PRESENT-128 for constrained hardware, SPECK for latency critical 8-bit software and a low cost key schedule for frequent re-keying and correctly identifies those configurations that fall below the 128-bit floor for sensitive data. OLEF-CIoT thus replaces the ad hoc selection of a cipher with an auditable process and provides the basis for the unified experimental testing that will form the next stage of this work.

REFERENCES

[1] V. A. Thakor, M. A. Razzaque, and M. R. A. Khandaker, "Lightweight cryptography algorithms for resource-constrained IoT devices: A review, comparison and research opportunities," *IEEE Access*, vol. 9, pp. 28177–28193, 2021, doi: 10.1109/ACCESS.2021.3052867.

[2] S. Singh, P. K. Sharma, S. Y. Moon, and J. H. Park, "Advanced lightweight encryption algorithms for IoT devices: Survey, challenges and solutions," *J. Ambient Intell. Humanized Comput.*, 2017, doi: 10.1007/s12652-017-0494-4.

[3] P. Shah, M. Arora, and K. Adhvaryu, "Lightweight cryptography algorithms in IoT – A study," in *Proc. 4th Int. Conf. I-SMAC (IoT in Social, Mobile, Analytics and Cloud)*, 2020, pp. 332–336.

[4] C. Silva, N. Tenório, and J. Bernardino, "Lightweight encryption algorithms for IoT," *Computers*, vol. 14, no. 12, Art. no. 505, 2025, doi: 10.3390/computers14120505.

 This work is licensed under a Creative Commons Attribution 4.0 International License

DOI: 10.17148/IJARCCE.2026.151189

[5] P. Panahi, C. Bayılmış, U. Çavuşoğlu, and S. Kaçar, "Performance evaluation of lightweight encryption algorithms for IoT-based applications," *Arabian J. Sci. Eng.*, vol. 46, pp. 4015–4037, 2021, doi: 10.1007/s13369-021-05358-4.

[6] I. Radhakrishnan, S. Jadon, and P. B. Honnavalli, "Efficiency and security evaluation of lightweight cryptographic algorithms for resource-constrained IoT devices," *Sensors*, vol. 24, no. 12, Art. no. 4008, 2024, doi: 10.3390/s24124008.

[7] T. K. Goyal, V. Sahula, and D. Kumawat, "Energy efficient lightweight cryptography algorithms for IoT devices," *IETE J. Res.*, 2019, doi: 10.1080/03772063.2019.1670103.

[8] A. Khompysh, N. Kapalova, O. Lizunov, D. Dyusenbayev, and K. Sakan, "Development of a new lightweight encryption algorithm," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 5, pp. 452–459, 2023.

[9] O. Sabri, B. Al-Shargabi, A. Abuarqoub, and T. A. Hakami, "A lightweight encryption method for IoT-based healthcare applications: A review and future prospects," *IoT*, vol. 6, no. 2, Art. no. 23, 2025.

[10] S. Kumar, K. Abhishek, R. H. Jhaveri, A. Alabdulatif, and R. Gaur, "An efficient dual encryption of IoMT data using lightweight security scheme for cloud based IoT environment," in *Proc. 38th ACM/SIGAPP Symp. Appl. Comput. (SAC '23)*, 2023, pp. 1782–1788.

[11] X. Yao, Z. Chen, and Y. Tian, "A lightweight attribute-based encryption scheme for the Internet of Things," *Future Gener. Comput. Syst.*, vol. 49, pp. 104–112, 2015.

[12] D. Karunkuzhali, A. A. Shaikh, R. Suguna, and M. Venkatesan, "Hybrid lightweight cryptography with attribute-based encryption for secure health monitoring in IoT-wireless body area sensor network," *Biomed. Mater. Devices*, vol. 4, pp. 2466–2482, 2026, doi: 10.1007/s44174-025-00402-5.

[13] E. T. Oladipupo, O. C. Abikoye, and J. B. Awotunde, "A lightweight image cryptosystem for cloud-assisted Internet of Things," *Appl. Sci.*, vol. 14, Art. no. 2808, 2024.

[14] K. Jain, B. Titus, P. Krishnan, S. Sudevan, P. Prabu, and A. S. Alluhaidan, "A lightweight multi-chaos-based image encryption scheme for IoT networks," *IEEE Access*, vol. 12, pp. 62118–62148, 2024, doi: 10.1109/ACCESS.2024.3377665.

[15]Ahamad MM, Abdullah MI (2016) Comparison of encryption algorithms for multimedia. Rajshahi Univ J Sci Eng 44:131–139

[16]Al Salami S, Baek J, Salah K, Damiani E (2016) Lightweight encryption for smart home. In: Proceeding of 2016 11th International Conference on Availability, Reliability and Security (ARES), IEEE, pp 382–388

[17]Aumasson JP, Henzen L, Meier W, Naya-Plasencia M (2013) Quark: a lightweight hash. J Crypto 26(2):313–339

[18]Babbage S, Dodd M (2008) The MICKEY stream ciphers. In: Proceeding of New Stream Cipher Designs, Springer, Berlin, pp 191–209

[19]Badel S, Dağtekin N, Nakahara JJ, Ouafi K, Reffé N, Sepehrdad P, Vaudenay S (2010) ARMADILLO: a multi-purpose crypto graphic primitive dedicated to hardware. In: Proceeding of International Workshop on Cryptographic Hardware and Embedded Systems, Springer, Berlin. pp 398–412

[20]Baharon MR, Shi Q, Llewellyn-Jones D (2015) A new lightweight homomorphic encryption scheme for mobile cloud computing. In: Proceeding of 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/ PICOM), IEEE. pp 618–625