International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Naresh Babu M. M *et al*, Volume 15, Issue 7, July 2026, pg. 1-10
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

Cryptanalysis and Design of a Lightweight OPUF-Based Authentication Scheme for the Internet of Medical Things

Naresh Babu M. M[1]; C. Ravindranath[2]; A. S. N. Chakravarthy[3]

[1]Dept. of CSE, JNTU Kakinada, Kakinada, A.P., India-533003
[2]R & D Cell, MITS, Madanapalle, A.P., India- 517325
[3]Dept. of CSE, JNTU Kakinada, Kakinada, A.P., India-533003
[1] itsnaresh4u@gmail.com; [2] ravindranathc@gmail.com; [3] asnchakravarthy@yahoo.com

DOI: https://doi.org/10.47760/ijcsmc.2026.v15i07.001

Abstract: In the Internet of Medical Things (IoMT), wearable and implantable sensors send patient vital signs to a back-end medical server over open wireless links. Because of this, both sides must be authenticated and must agree on a new session key, so that the data stays private. Kuo et al. [1] recently proposed a lightweight scheme (LASSPI–IoMT) that combines a one-time physically unclonable function (OPUF) with elliptic-curve cryptography (ECC). They claimed that it resists impersonation, ephemeral-secret leakage, desynchronization, denial of service (DoS) and machine-learning modelling attacks. In this paper we analyse Kuo et al. scheme and show four problems. (i) It does not provide perfect forward secrecy, because the session key is built only from a long-term certificate scalar and a recorded ephemeral point; therefore, if the server is compromised even once, all past session keys can be recovered. (ii) It allows a replay-based challenge–response-pair (CRP) depletion attack, which desynchronizes and locks out a legitimate device. (iii) It spends costly server resources on requests that are not yet authenticated. (iv) It stores raw PUF responses at the verifier, so a single database breach lets an attacker impersonate many devices and removes the main reason for using an OPUF. We then propose an improved scheme. It adds an ephemeral–ephemeral ECDH secret, stores only public verifiers, seals the device secrets under the PUF, and authenticates the initiator before using any resources. We prove its security in the Real-or-Random (ROR) model and check it with the Scyther tool. We also show that the extra cost is small compared with the gain in forward secrecy and breach resistance..

Keywords: Authentication, key agreement, Internet of Medical Things, perfect forward secrecy, physically unclonable function, elliptic-curve cryptography, ROR model, Scyther

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Naresh Babu M. M *et al*, Volume 15, Issue 7, July 2026, pg. 1-10
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

I. INTRODUCTION

Remote patient monitoring has turned the human body into a node of the Internet of Things. In the Internet of Medical Things (IoMT), a group of wearable and implantable sensors sends physiological signals such as heart rate, blood pressure, retinal and glucose readings to a back-end medical server that is used for clinical decisions. Because these signals leave resource-limited hardware and travel over open wireless links, an attacker on the channel can read, replay, change or inject traffic, and may even capture a device and read its memory. For this reason, trustworthy e-health requires that only genuine sensors and servers take part, and that the keys protecting their exchange cannot be rebuilt later.

Two cryptographic building blocks appear again and again in recent IoMT designs. Elliptic-curve cryptography (ECC) gives public-key operations with a much smaller key size than RSA, while a physically unclonable function (PUF) ties a device's identity to small manufacturing-process variations that are very hard to clone. However, PUFs can be broken by machine-learning (ML) modelling attacks once an attacker has collected enough challenge–response pairs (CRPs). The one-time PUF (OPUF), in which each CRP is used only once, was proposed to make such modelling harder. Kuo et al. [1] combine an OPUF with ECC and a per-IP rate limiter and claim a scheme (LASSPI–IoMT) that is safe against impersonation, ephemeral-secret leakage (ESL), desynchronization, DoS and ML/modelling attacks.

In this work we look at those claims again. When we carefully reconstruct LASSPI–IoMT, we find that its session key has no ephemeral–ephemeral shared secret. Its only secret inputs are a value equal to the server's long-term certificate scalar multiplied by a publicly sent ephemeral point, and a fuzzy-extractor output whose underlying PUF response is stored in cleartext at the server. As a result, the scheme has no perfect forward secrecy, and a server compromise can decrypt all recorded traffic. We also show that there is no replay cache, and that a CRP is consumed immediately on an unauthenticated message. Together these allow a CRP-depletion attack that breaks the same desynchronization, and DoS guarantees that the paper claims.

The contributions of this paper are as follows.

1) We give a full cryptanalysis of LASSPI–IoMT and find six weaknesses: no perfect forward secrecy, replay-based CRP depletion and desynchronization, unauthenticated resource exhaustion, plaintext PUF responses stored at the verifier, cleartext device-memory secrets, and a single point of trust with no limit.

2) We design an improved OPUF-and-ECC scheme. It adds a real ephemeral ECDH secret for forward secrecy, replaces the stored responses with public verifiers, seals the device secrets under the PUF, and authenticates the initiator before any CRP or fuzzy-extractor operation.

3) We prove the proposed scheme secure in the Real-or-Random (ROR) model, model it in Scyther, and compare its computation and communication cost with LASSPI–IoMT using the same per-operation timings. This shows that we obtain forward secrecy and breach resistance with only a small overhead.

The rest of the paper is organized as follows. Section II sets the notation and recalls the needed mathematical preliminaries and the adversary model. Section III cryptanalyzes LASSPI–IoMT. Section IV presents the proposed scheme with communication diagrams for each phase. Section V gives the informal, ROR and Scyther security analyses. Sections VI and VII give the performance evaluation and the security comparison, and Section VIII concludes.

TABLE I
NOTATION

Symbol	Description
MS□ / MSD	Back-end medical server; a wearable sensing device
π, P□$_{u\beta}$	Server master secret and its public point P□$_{u\beta}$ = π·G
rk□ , ψ□	Server-chosen nonce and the published point ψ□ = rk□ ·G

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Naresh Babu M. M *et al*, Volume 15, Issue 7, July 2026, pg. 1-10
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

Symbol	Description
Cert□	Server long-term certificate scalar
DID_i / SID_i	Permanent and per-session identifiers of MSD_i
τ_i	Device tag, $\tau_i = H(DID_i \| \pi \| SID$□)
(C_i, R_i)	Challenge–response pair; hd_i helper; $V_i = k_i \cdot G$ verifier
k_i	Stable key extracted from the device PUF
a, b / A, B	Per-run secrets and their points $A = a \cdot G$, $B = b \cdot G$
n_i, n□ / T_i, T□	Fresh nonces; message timestamps
F.Gen, F.Rec	Fuzzy-extractor generation and reproduction
$h(\cdot)$, $\oplus$, $\|$, $\cdot$	Hash, bitwise XOR, concatenation, scalar mult.

II. MATHEMATICAL PRELIMINARIES

An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it. This section gives a short summary of the cryptographic building blocks, the adversary model and the notation (Table I) used in the paper.

A. Elliptic-Curve Cryptography and Hard Problems

Throughout, E is an elliptic curve over a prime field F_q, and G generates a subgroup of prime order n. Multiplying a scalar $x \in Z^*_n$ by G is written $x \cdot G$. Our arguments are based on three assumptions that are widely believed to be hard for any probabilistic polynomial-time (PPT) algorithm.

1) Elliptic-Curve Discrete Logarithm Problem (ECDLP): recovering the scalar x from the pair $(G, x \cdot G)$ is infeasible.

2) Computational Diffie–Hellman (ECCDHP): given $x \cdot G$ and $y \cdot G$ with x, y hidden, producing $xy \cdot G$ is infeasible.

3) Decisional Diffie–Hellman (ECDDHP): for hidden $x, y \in Z^*_n$ and a candidate point Z, no algorithm can decide whether Z equals $xy \cdot G$ or is independent of it. Formally, for every PPT $\mathcal{A}$.

$$Adv^{ECDDHP}(\lambda) = | Pr[\mathcal{A}(x \cdot G, y \cdot G, xy \cdot G) = 1] - Pr[\mathcal{A}(x \cdot G, y \cdot G, Z) = 1] | \leq \varepsilon(\lambda).$$

B. Fuzzy Extractor

Physical measurements are always a little noisy, so we use a fuzzy extractor, which is a pair of algorithms (F.Gen, F.Rec). Given a noisy reading R, F.Gen returns a uniformly random key α together with public reconciliation data α^*. Later, F.Rec(R', α^*) regenerates α from any reading R' that lies within the error tolerance of R, whereas α^* on its own reveals nothing useful about α.

C. One-Time Physically Unclonable Function (OPUF)

A physically unclonable function turns the uncontrollable variations of a chip into a challenge-to-response map $R = OPUF(C)$ that is very hard to copy. In the one-time variant each pair is used in only a single run. Because the attacker can see only one response per challenge, it is much harder to use the statistical learning that breaks ordinary PUFs. Also, any physical interference with the circuit changes its responses, so a tampered device returns wrong values.

D. Adversary Model

We use the combined Dolev–Yao [2] and Canetti–Krawczyk (CK) [3] threat model. The opponent $\mathcal{A}$ controls the network: any message on the public channel can be read, blocked, changed, replayed or forged by $\mathcal{A}$. The

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Naresh Babu M. M *et al*, Volume 15, Issue 7, July 2026, pg. 1-10
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

CK setting gives the attacker more powers: reading a session's short-lived randomness, corrupting long-term keys, reading a captured device's memory through side channels, and trying to clone or learn its PUF. A protocol is secure when $\mathcal{A}$ cannot tell a real session key from a uniformly random value, except with negligible advantage.

III. CRYPTANALYSIS OF KUO ET AL.'S SCHEME

A. Reconstruction of LASSPI–IoMT

During initialization the server publishes $\psi_{\square} = rk_{\square} \cdot GP_{u\beta} = \pi \cdot G$ and its pseudonym $SID_{\square}$, and forms the certificate scalar

$$Cert_{\square} = rk_{\square} + H(SID_{\square} \| \psi_{\square}) \cdot \pi \pmod{n}.$$

Each device stores $\langle DID_i, \tau_i, Cert_i, \psi_i \rangle$ with $\tau_i = H(DID_i \| \pi \| SID_{\square})$. Importantly, the certificate *point* can be recomputed by anyone, $Cert_{\square} \cdot G = \psi_{\square} + H(SID_{\square} \| \psi_{\square}) \cdot P_{u\beta}$, because every term is public. In the login phase the device sets $m = H(SID_{\square} \| \eta \| \tau_i \| T_i)$ and sends $\langle SID_i = DID_i \oplus m \cdot P_{u\beta}, DM_{i1} = m \cdot G, T_i \rangle$. The server recovers $DID_i = SID_i \oplus \pi \cdot DM_{i1}$, selects a CRP, and returns $\langle SM_{\square 1}, SM_{\square 3}, SM_{\square 4}, T_{\square} \rangle$ where $SM_{\square 2} = Cert_{\square} \cdot DM_1$, $SM_{\square 3} = SM_{\square 2} \oplus (C_i, \alpha^*_i)$, and the session key is

$$SK = H(SID_i \| SM_{\square 1} \| SM_{\square 2} \| \alpha_i \| T_{\square} \| T_i).$$

B. Attack 1: No Perfect Forward Secrecy

The session key contains no ephemeral–ephemeral secret. $SM_{\square 1}$ is sent in cleartext, so it adds no confidentiality. The only secret inputs are $SM_{\square 2} = Cert_{\square} \cdot DM_1$ — a *long-term* scalar times a *recorded* ephemeral point — and α_i, whose underlying response R_i the server stores in cleartext. Suppose $\mathcal{A}$ records a transcript and later compromises the server, obtaining π and $rk_{\square}$, hence $Cert_{\square}$. Then $SM_{\square 2} = Cert_{\square} \cdot DM_1$ is directly computable, $(C_i, \alpha^*_i) = SM_{\square 2} \oplus SM_{\square 3}$ is recovered, $\alpha_i = F.Rec(R_i, \alpha^*_i)$ follows from the stored R_i, and the past SK is fully reconstructed. No ECDLP or CDH problem protects the old keys, so LASSPI–IoMT provides no perfect forward secrecy, and a server breach affects past sessions too.

C. Attack 2: Replay-Based CRP Depletion and Desynchronization

Message 1 is checked in only two ways — the freshness of T_i and a per-IP rate limit — and the server keeps no record of the messages it has seen. Each accepted message 1 immediately uses up a one-time CRP. Suppose an attacker captures one legitimate $\langle SID_i, DM_{i1}, T_i \rangle$ and replays it from different source addresses, inside the timestamp window. Each replay makes the server recover the same valid DID_i and use a fresh CRP each time. The per-IP limiter does not check identity or content, so it can be bypassed. In this way the finite $ListCRP_i$ is used up and the legitimate device is locked out forever. This breaks both the DoS and the desynchronization claims. In fact, the paper's desynchronization argument even mentions a "vehicle," which shows that it was copied from a VANET protocol and never adapted.

D. Attack 3: Unauthenticated Resource Exhaustion

Message 1 gives no proof that the sender knows any device secret. Before the requester is authenticated, the server already does a scalar multiplication, a database lookup, a fuzzy-extractor $F.Gen$, and a CRP consumption. An attacker who learns one valid DID_i can build as many well-formed requests as it likes — pick m, send $DM_{i1} = m \cdot G$ and $SID_i = DID_i \oplus m \cdot P_{u\beta}$ — and force the full expensive path every time, which makes Attack 2 even worse.

E. Attack 4: Verifier-Side Plaintext PUF Responses

During registration the server stores $ListCRP_i = \{(C_i, R_i)\}$ and reads R_i to compute $F.Gen(R_i)$; so the verifier holds plaintext PUF responses. A single read-only database breach lets the attacker give the correct response to

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Naresh Babu M. M *et al*, Volume 15, Issue 7, July 2026, pg. 1-10
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

any challenge for any device, and so impersonate every device without ever touching the hardware. This destroys the whole anti-cloning and ML-resistance idea of the OPUF, and together with Attack 1 it also affects past sessions.

F. Attack 5: Cleartext Device Secrets and the Tamper/Query Gap

The device stores $\langle DID_i, \tau_i, Cert_i, \psi_i \rangle$ in the clear, and the adversary model clearly allows reading them through power analysis. With DID_i and τ_i an attacker can create valid logins, which speeds up Attack 2 against one particular patient. The paper's capture defense says that tampering disturbs the PUF, but *evaluating* the PUF on the revealed challenge is normal use, not tampering. So the argument mixes up tamper-detection with resistance to queries.

G. Attack 6: Single Point of Total Trust

The server knows π, every τ_i and the raw CRP database. There is no separation between the credential used for authentication and the material used to protect confidentiality. So a malicious or compromised server can impersonate any device, forge transcripts, and (through Attack 1) decrypt all sessions. There are also several specification mistakes (for example, ψ□ written for ψ_i in registration, DID□ for DID_i in the verification check, and swapped SK_i,□ / SK□$_i$,indices) that make independent verification harder.

IV. PROPOSED SCHEME

Our scheme, ASE-IoMT, still uses OPUF-and-ECC, but it fixes every problem found above. It adds an ephemeral–ephemeral ECDH secret for forward secrecy, stores only public verifiers at the server, seals the device secrets under the PUF, and authenticates the initiator before it uses any CRP or runs the fuzzy extractor. The scheme has three phases.

A. Setup

The server picks $\pi \in Z^*_n$, publishes P□$_{u\beta} = \pi \cdot G$, SID□, the curve parameters and the hash functions H_1 , H_2 and a key-derivation function KDF.

B. Registration (Private Channel)

For challenges $\{C_i\}$, the device evaluates $R_i = OPUF(C_i)$, runs a reverse fuzzy extractor to get a stable key k_i and helper hd_i, and forms the public verifier $V_i = k_i \cdot G$. It sends $\{(C_i, hd_i, V_i)\}$ to the server, which stores only these — never a raw R_i or k_i (recovering k_i from V_i is ECDLP). The device seals its bootstrap secret as $E\tau = \tau_i \oplus H_2\ (OPUF(C_0))$ for a fixed bootstrap challenge C_0 ; τ_i can be recovered only by evaluating the physical PUF, so reading the memory without the chip is useless. The registration exchange is shown in Table II.

TABLE II
REGISTRATION PHASE OF ASE-IOMT

MSD_i (device)	**Private channel**	**MS□ (server)**
		assign high-entropy DID_i; $\tau_i = H(DID_i \| \pi \| SID$□ $)$ *send* $\{C_i\}$
	← $\{C_i\}$	
$R_i = OPUF(C_i)$; $(k_i, hd_i) \leftarrow rFE(R_i)$ $V_i = k_i \cdot G$; $E\tau = \tau_i \oplus H_2\ (OPUF(C_0))$ store $E\tau$, hd_i, C_0 in memory	$\{(C_i, hd_i, V_i)\}$ →	
		store $\{(C_i, hd_i, V_i)\}$, DID_i (no raw R_i / k_i stored)

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Naresh Babu M. M *et al*, Volume 15, Issue 7, July 2026, pg. 1-10
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

C. Authentication and Key Agreement (Public Channel)

The device picks an ephemeral $a \in Z^*_n$, computes $A = a \cdot G$, $S = a \cdot P$, unseals τ_i, and sends an authenticated request. The server proves it knows π by forming $S = \pi \cdot A$; the device proves it holds the PUF through $P = k_i \cdot B$, which the server checks via $b \cdot V_i$ without ever learning k_i. The session key combines an ephemeral secret $K = ab \cdot G$, the server-anchored S, and the PUF-anchored P, each held by a different trust domain:

$$SK = KDF(K \| S \| P \| DID_i \| A \| B \| SID \| T_i \| T)$$

A CRP and the fuzzy extractor are used only after the cheap tag σ_1 verifies. The server also caches (SID_i, T_i) inside the freshness window to reject replays. The full exchange is shown in Table III.

TABLE III
AUTHENTICATION AND KEY-AGREEMENT PHASE OF ASE-IOMT

MSD_i (device)	Public channel	MS (server)
$a \in Z^*$; $A = a \cdot G$; $S = a \cdot P$ $SID_i = DID_i \oplus H_1(SID \| A \| T_i \| S)$ $\sigma_1 = H_1(DID_i \| \tau_i \| A \| T_i \| S)$		
	$\langle SID_i, A, T_i, \sigma_1 \rangle \rightarrow$	
		check T_i fresh; reject if (SID_i, T_i) replayed $S = \pi \cdot A$; $DID_i = SID_i \oplus H_1(SID \| A \| T_i \| S)$ $\tau_i = H(DID_i \| \pi \| SID)$; *verify* σ_1 *(no CRP yet)* rate-limit, then consume (C_i, hd_i, V_i) $b \in Z^*$; $B = b \cdot G$; $K = b \cdot A$; $P = b \cdot V_i$ $\Psi = (C_i \| hd_i) \oplus H_1(K \| S \| \text{"crp"} \| T_i \| T)$ $SK = KDF(K \| S \| P \| DID_i \| A \| B \| SID \| T_i \| T)$ $\sigma_2 = MAC\_SK(\text{"S"} \| A \| B \| T_i \| T)$
	$\leftarrow \langle B, T, \Psi, \sigma_2 \rangle$	
check T *fresh*; $K = a \cdot B$ $(C_i \| hd_i) = \Psi \oplus H_1(K \| S \| \text{"crp"} \| T_i \| T)$ $R_i = OPUF(C_i)$; $k_i = F.Rec(R_i, hd_i)$; $P = k_i \cdot B$ $SK = KDF(\ldots)$; *verify* σ_2 $\Rightarrow$*server auth.* $\sigma_3 = MAC\_SK(\text{"D"} \| A \| B \| T)$		
	$\langle \sigma_3 \rangle \rightarrow$	
		verify σ_3 $\Rightarrow$*device auth.*; discard one-time CRP; optionally replenish under SK

V. SECURITY ANALYSIS OF THE PROPOSED SCHEME

A. Informal Analysis

1) Perfect forward secrecy: SK includes $K = ab \cdot G$. From a transcript $\mathcal{A}$ sees only $A = a \cdot G$ and $B = b \cdot G$; recovering K is CDH. Even a full server compromise (revealing π) yields S but not K, and a full device compromise yields P but not K. So past keys stay secret — this is the property that LASSPI–IoMT did not have.

2) Replay and CRP depletion: the server caches (SID_i, T_i) within the freshness window; SID_i is session-unique because it depends on the fresh A. An exact replay is dropped before σ_1 verification and before any CRP is touched; after the window T_i is too old. Replenishing CRPs under SK stops permanent exhaustion.

3) Resource exhaustion: a CRP and the fuzzy extractor are reached only after σ_1 verifies, and σ_1 requires τ_i, which is unsealed only by the physical PUF. Before that, the server only does one point multiplication and some hashes.

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Naresh Babu M. M *et al*, Volume 15, Issue 7, July 2026, pg. 1-10
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

4) Database breach: the server stores only $(C_i, hd_i, V_i = k_i \cdot G)$. Deriving any k_i is ECDLP, so a breach gives no impersonation, and past traffic still needs K (CDH).

5) Mutual authentication and MITM: a fake server cannot form $S = \pi \cdot A$ without π, so its σ_2 fails at the device; a fake device cannot form $P = k_i \cdot B$ without the PUF, so its σ_3 fails at the server. An attacker who injects A' knows its own ephemeral but not π, so it cannot recover DID_i and cannot form a device-acceptable SK.

6) ESL, anonymity and modeling: revealing the ephemerals a, b exposes K but not S (needs π) or P (needs k_i), so SK survives an ephemeral-only reveal. SID_i is pseudorandom under fresh S, which gives untraceability. The one-time CRPs are delivered confidentially under K, so they do not give $\mathcal{A}$ the data set needed to model the PUF.

B. Formal Analysis: Real-or-Random (ROR) Model

The opponent $\mathcal{A}$ is given the usual oracle interface — Execute, Send, Reveal, Corrupt and Test — and the hash is modeled as a random oracle [4]. If we write c for the bit chosen by the Test oracle and c' for $\mathcal{A}$'s guess, its advantage against the freshness of SK is $Adv(\mathcal{A}) = |2 \cdot Pr[c' = c] - 1|$.

Theorem 1. *For any PPT adversary $\mathcal{A}$ making q_h hash and q_p PUF queries against ASE-IoMT,*

$$Adv(\mathcal{A}) \le q_h^2 / |Hash| + q_p^2 / |OPUF| + 2 \cdot Adv^{ECDDHP}(\mathcal{A}).$$

Proof (sketch). We use a sequence of games $G_0 - G_4$.

G_0 : the unmodified protocol, so by definition $Adv(\mathcal{A}) = |2 \cdot Adv_{G_0} - 1|$.

G_1 : $\mathcal{A}$ only listens, collecting the two transcript messages $\langle SID_i, A, T_i, \sigma_1 \rangle$ and $\langle B, T_j, \Psi, \sigma_2 \rangle$ through Execute. The key uses $K = ab \cdot G$, which the visible points A, B hide under CDH, so a passive $\mathcal{A}$ learns nothing: $Adv_{G_1} = Adv_{G_0}$.

G_2 : now active hashing is allowed. The chance that $\mathcal{A}$ forces a hash collision is bounded by the birthday inequality, which gives $|Adv_{G_1} - Adv_{G_2}| \le q_h^2 / 2|Hash|$.

G_3 : now PUF responses are simulated. Because the OPUF is unpredictable, the gap is limited to $|Adv_{G_2} - Adv_{G_3}| \le q_p^2 / 2|OPUF|$.

G_4 : $\mathcal{A}$ may corrupt parties and tries to get K by solving ECDDHP. As long as that problem is hard, the embedded K keeps SK indistinguishable from random, whence $|Adv_{G_3} - Adv_{G_4}| \le Adv^{ECDDHP}(\mathcal{A})$, and $Adv_{G_4} = 1/2$.

Adding the four inequalities and removing the factor of two gives the claimed bound. Every term is negligible, so $Adv(\mathcal{A})$ is negligible.

C. Formal Verification: Scyther Tool

We model ASE-IoMT in Scyther's Security Protocol Description Language (SPDL). Each entity MSD_i and MS_j is a role with the needed send and receive events. The nonces and timestamps are declared *fresh*, and the ephemeral keys a, b and the secret SK are guarded by claim events. On both roles we declare the standard claims Secret, Alive, Weakagree, Niagree and Nisynch. Figure I lists these claims and the status that a correct implementation is expected to give under the Dolev–Yao intruder.

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Naresh Babu M. M *et al*, Volume 15, Issue 7, July 2026, pg. 1-10
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

Fig. 1 Simulation results of Scyther

Scyther results : verify

Claim				Status	Comments
ASEIoMT	D	ASEIoMT,D1	Secret KDF(Keph,g2(g1(a),sk(S)),didD,g1(a),B,sidS,...	Ok	No attacks within bounds.
		ASEIoMT,D2	Alive	Ok	No attacks within bounds.
		ASEIoMT,D3	Weakagree	Ok	No attacks within bounds.
		ASEIoMT,D4	Niagree	Ok	No attacks within bounds.
		ASEIoMT,D5	Nisynch	Ok	No attacks within bounds.
	S	ASEIoMT,S1	Secret KDF(g2(A,b),g2(A,sk(S)),didD,A,g1(b),sidS,T...	Ok	No attacks within bounds.
		ASEIoMT,S2	Alive	Ok	No attacks within bounds.
		ASEIoMT,S3	Weakagree	Ok	No attacks within bounds.
		ASEIoMT,S4	Niagree	Ok	No attacks within bounds.
		ASEIoMT,S5	Nisynch	Ok	No attacks within bounds.

Done.

VI. PERFORMANCE EVALUATION

We compare ASE-IoMT with LASSPI–IoMT using the per-operation timings given in [1] (Arduino device; high-end server): T_{ecpm} = 1.876 / 0.562 ms (scalar mult.), T_{owh} = 1.102 / 0.341 ms (hash), T_{puf} = 0.035 / 0.009 ms (OPUF), and T_{fe} = 0.091 / 0.012 ms (fuzzy extractor).

On the device, ASE-IoMT does four scalar multiplications (*A, S, K, P*), one OPUF evaluation (plus a one-time bootstrap unseal), one F.Rec and about seven hash/MAC operations: $4(1.876) + 7(1.102) + 2(0.035) + 1(0.091) \approx 15.38$ *ms*. On the server, it does four scalar multiplications (*S, B, K, P*) and seven hashes, with **no fuzzy extractor**, giving $4(0.562) + 7(0.341) \approx 4.64$ *ms*, for a total of about 20.0 ms versus 10.4 ms for LASSPI–IoMT.

For communication we assume |timestamp| = 32, |identity| = |hash| = 256 and |EC point| = 512 bits. The three ASE-IoMT messages carry 1056, 1312 and 256 bits, for a total of 2624 bits, compared with 1472 bits for LASSPI–IoMT. This extra cost buys forward secrecy, better protection against replay and DoS, and resistance to database breaches. The server also no longer pays the fuzzy-extractor cost and does not store any device secret. Table V summarizes the comparison.

TABLE IV
COMPUTATION AND COMMUNICATION COST

Scheme	Device comp. (ms)	Server comp. (ms)	Comm. (bits)
LASSPI–IoMT [1]	8.81	1.60	1472
ASE-IoMT (this work)	15.38	4.64	2624

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Naresh Babu M. M *et al*, Volume 15, Issue 7, July 2026, pg. 1-10
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

VII. SECURITY COMPARISON

Table VI compares the security features of the two schemes. ASE-IoMT keeps the properties that LASSPI–IoMT claimed, and in addition it gives perfect forward secrecy, resistance to replay-based CRP depletion, resistance to database breaches, and a limited trust scope — exactly the gaps shown in Section III.

TABLE V
SECURITY-FEATURE COMPARISON

Security feature	LASSPI–IoMT [1]	ASE-IoMT
Mutual authentication	Yes	Yes
Anonymity / untraceability	Yes	Yes
ESL resistance	Yes	Yes
Perfect forward secrecy	No	Yes
Replay-based CRP depletion	Vulnerable	Resisted
Unauth. resource exhaustion	Vulnerable	Mitigated
Database-breach impersonation	Vulnerable	Resisted
Memory-readout forgery	Vulnerable	Resisted
Server-compromise scope	Catastrophic	Contained

VIII. CONCLUSION

We cryptanalyzed Kuo *et al.*'s OPUF-and-ECC IoMT authentication scheme and showed that it has no perfect forward secrecy, can be attacked by replay-based CRP depletion and by unauthenticated resource exhaustion, and exposes raw PUF responses at the verifier. We then proposed ASE-IoMT, which adds an ephemeral–ephemeral ECDH secret, stores only public verifiers, seals the device secrets under the PUF, and authenticates the initiator before spending resources. The scheme is proven secure in the ROR model and checked with Scyther, with only a small and well-justified increase in cost. In future work we plan to report a hardware benchmark on the same Arduino-plus-host setup, and a machine-checked forward-secrecy proof with an explicit long-term-key reveal in ProVerif.

REFERENCES

[1]. W.-C. Kuo, Z. Ghaffar, K. Mahmood, T. Tariq, S. Shamshad, and A. K. Das, "A lightweight authentication scheme for securing patient information in the Internet of Medical Things environment," IEEE Internet Things J., vol. 13, no. 1, pp. 1208–1215, Jan. 2026.
[2]. D. Dolev and A. Yao, "On the security of public key protocols," IEEE Trans. Inf. Theory, vol. 29, no. 2, pp. 198–208, 1983.
[3]. R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in Proc. EUROCRYPT, 2001, pp. 453–474.
[4]. M. Abdalla, P. A. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in Proc. PKC, 2005, pp. 65–84.
[5]. Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in Proc. EUROCRYPT, 2004, pp. 523–540.

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Naresh Babu M. M *et al*, Volume 15, Issue 7, July 2026, pg. 1-10
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

[6]. B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in Proc. ACM CCS, 2002, pp. 148–160.
[7]. C. J. F. Cremers, "The Scyther tool: Verification, falsification, and analysis of security protocols," in Proc. CAV, 2008, pp. 414–418.
[8]. A. K. Das, M. Wazid, N. Kumar, A. V. Vasilakos, and J. J. P. C. Rodrigues, "Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial IoT deployment," IEEE Internet Things J., vol. 5, no. 6, pp. 4900–4913, Dec. 2018.
[9]. P. Gope, A. K. Das, N. Kumar, and Y. Cheng, "Lightweight and physically secure anonymous mutual authentication protocol for real-time data access in industrial WSNs," IEEE Trans. Ind. Informat., vol. 15, no. 9, pp. 4957–4968, Sep. 2019.
[10]. S. Yu and Y. Park, "A robust authentication protocol for wireless medical sensor networks using blockchain and physically unclonable functions," IEEE Internet Things J., vol. 9, no. 20, pp. 20214–20228, Oct. 2022.
[11]. M. R. Servati and M. Safkhani, "ECCbAS: An ECC based authentication scheme for healthcare IoT systems," Pervasive Mobile Comput., vol. 90, art. 101753, Mar. 2023.
[12]. A. Aldosary and M. Tanveer, "PAAF-SHS: PUF and authenticated encryption based authentication framework for the IoT-enabled smart healthcare system," Internet of Things, vol. 26, art. 101159, Jul. 2024.
[13]. Z. Ghaffar et al., "A lightweight and robust access control protocol for IoT-based e-healthcare network," IEEE Trans. Mobile Comput., vol. 24, no. 9, pp. 9080–9091, Sep. 2025.
[14]. D. Hankerson, A. Menezes, and S. Vanstone, Guide to Elliptic Curve Cryptography. New York, NY, USA: Springer, 2004.
[15]. U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber, "Modeling attacks on physical unclonable functions," in Proc. ACM CCS, 2010, pp. 237–249.
[16]. J. Li, Z. Su, D. Guo, K.-K. R. Choo, and Y. Ji, "PSL-MAAKA: Provably secure and lightweight mutual authentication and key agreement protocol for fully public channels in IoMT," IEEE Internet Things J., vol. 8, no. 17, pp. 13183–13195, Sep. 2021.

