On the Security of Authentication Protocols for Remote Healthcare Systems Through Cryptographic Vulnerability Analysis and Secure Protocol Redesign

Haewon Byeon*
Department of Future Technology, Korea University of Technology and Education (KOREA TECH),
Cheonan 31253, South Korea

***Abstract*—This paper revisits a previously proposed authentication scheme for remote healthcare systems in Cloud-IoT. Although that protocol was introduced as a repair of an earlier healthcare design and was claimed to satisfy the usual confidentiality and mutual-authentication goals, a closer reconstruction of its registration, login, authentication, and password-update logic reveals several structural weaknesses. The analysis shows that a stolen smart card combined with one captured transcript enables offline password verification, that the session key is deterministic for a fixed user-sensor pair, that static pseudonyms expose long-term linkability, and that compromise of the server-wide secret expands immediately to all registered sensors. To address these problems, the core key-management path is redesigned while keeping the original cloud-assisted remote healthcare architecture. The revised scheme uses a device-bound seed only for local recovery, a fresh elliptic-curve Diffie-Hellman exchange for every run, dynamic pseudonyms, and KDF-based session-key derivation with explicit context binding. A comparative evaluation against the Sharma-Kalra baseline and the 2021 Azrour design indicates that the revised protocol raises resistance to guessing, replay, cross-session correlation, and compromise propagation with only modest latency growth.**

***Keywords*—Remote Healthcare Security; Authentication Protocol; Internet of Medical Things; Cryptographic Vulnerability Analysis; Secure Session Key Establishment; Privacy-Preserving Authentication**

I. Introduction

Remote healthcare platforms have moved from pilot services to everyday infrastructure. Hospitals, clinics, care homes, and home-based monitoring programs now rely on wearable sensors, mobile gateways, and cloud dashboards to observe patient status beyond the traditional ward boundary [2]-[4]. The practical appeal is obvious: data can be collected continuously, clinicians can react without waiting for a physical visit, and health systems can reserve scarce in-person capacity for acute cases. This operational benefit, however, comes with a security burden. Vital signs, medication records, sensor alarms, and clinician commands travel across shared networks and pass through devices that are neither administratively uniform nor cryptographically strong by default. A remote healthcare protocol that is inexpensive in one benchmark but weak under realistic compromise assumptions can create precisely the kind of silent risk that the healthcare setting can least tolerate.

In this landscape, authentication is not a routine login problem. A medical professional may use a smart card or mobile terminal; a cloud service may mediate access control and storage; a resource-constrained sensor may answer only a small set of commands; and a patient context may require strict privacy even when the content of a message is encrypted. The challenge is therefore twofold. First, only authorized entities should reach the measurement path. Second, the protocol should keep exposure local when one component fails. The protocol proposed by Azrour et al. in "New Efficient and Secured Authentication Protocol for Remote Healthcare Systems in Cloud-IoT" [1] addresses exactly this setting. Their design was presented as a stronger alternative to the earlier Sharma-Kalra protocol [5], and the paper argued that the new method improves resistance to password guessing and stolen-card abuse while remaining lightweight enough for cloud-assisted healthcare services. Related designs for remote patient monitoring and cloud-IoT authentication share the same ambition: low overhead with sufficient resistance to replay, impersonation, and privacy leakage [6], [7].

The problem is that lightweight structure can hide security debt. Many healthcare authentication schemes compress several distinct roles into one server-side secret or one static pseudonym. Once that happens, independence between sessions becomes fragile. A value that should play the role of a long-term verifier also starts to play the role of a fresh authenticator; a user handle that should identify an account internally begins to function as a public tracing tag; and a password-derived transformation that seems harmless during registration turns into an offline checking oracle when a smart card is extracted. The scheme of Azrour et al. [1] is especially interesting because its surface presentation suggests that it has already learned from the weaknesses of Sharma and Kalra [5]. The article explicitly criticizes the predecessor for password guessing and key-exposure problems. That makes the 2021 scheme a useful target for a second-generation analysis: not a first-draft protocol, but a claimed repair of a knownly flawed one.

The text of "New Efficient and Secured Authentication Protocol for Remote Healthcare Systems in Cloud-IoT" [1] makes three types of claims that deserve close scrutiny. First, it argues that the protocol withstands offline guessing and smart-card theft. Second, it states that the design provides mutual authentication and forward secrecy. Third, it reports formal verification in Scyther and presents lower computational cost

*Corresponding author.

than several alternatives. These are meaningful claims, yet they are not independent. If the session key depends only on long-term values, forward secrecy fails even if message tags are computed correctly. If the card must store enough material to reconstruct a verifier locally, then a captured transcript may let an attacker test guessed passwords without ever contacting the server. If a single server secret is used to unwrap every sensor-side secret, then one compromise event scales horizontally across the whole deployment. Lightweight healthcare authentication must therefore be judged not only by whether each message is hashed, but by how the hidden dependencies between those hashes behave under compromise. Fig. 1 summarizes the evidence-driven workflow used to reconstruct the target protocol, identify attacker-checkable relations, and guide the redesign process.

This paper studies the target scheme from that perspective. We reconstruct the exact message flow and rewrite every derived value in explicit algebraic form. We then test a set of security invariants: password guesses must be unverifiable offline from card data and transcripts; session keys must change when fresh protocol randomness changes; public identifiers must not remain constant across sessions; and compromise of one role should not trivially reveal the long-term keys of another. This approach lets us move beyond generic statements such as "the scheme is secure against known attacks." Instead, we ask which equations become attacker-checkable once the smart card, the transcript, or the server database is partially exposed.

This study makes four contributions. First, a protocol-centric reconstruction of Azrour et al.'s 2021 remote healthcare scheme [1] is provided, preserving its registration, login, authentication, and password-update phases but making their key relations explicit. Second, concrete weaknesses are identified**:** offline password guessing, deterministic session-key reuse, persistent linkability of user activity, and weak compromise containment at the server-sensor boundary. Third, the scheme is redesigned while preserving the same remote healthcare architecture. The revised construction uses a device-bound local seed, dynamic pseudonym generation, ephemeral elliptic-curve key exchange, and context-bound KDF output rather than a static hash of long-term values. Fourth, the revised design is evaluated against the Sharma-Kalra baseline [5] and the Azrour et al. protocol [1] under a controlled simulation plan that records authentication success, replay detection, compromise resilience, and latency. The result is not an entirely different system; it is a targeted repair of the precise structural choices that make the original scheme brittle.

II. Related Work

The healthcare-authentication literature has developed along several overlapping lines. A first line comes from wireless medical sensor network protocols that focus on low computation at the sensing edge. Early work such as E-SAP by Kumar et al. [8] showed how mutual authentication could be adapted to medical sensor deployments with constrained devices, while later schemes extended this direction toward explicit anonymity and cloud-assisted access control [9], [10]. These designs established an important baseline: healthcare protocols must account for mobility, session freshness, and the special consequences of message forgery in a medical context. They also exposed a recurring tension. When designers reduce cost aggressively, they often centralize trust in a single server secret or static verifier, which saves computation but weakens compromise isolation.

A second line centers on telecare medicine information systems and smart-card-based login. Yan et al. [11] used biometrics to strengthen identity assurance, Mishra et al. [12] revised that design after showing nonce-related and guessing weaknesses, and Tan [13] moved toward a three-factor construction intended to improve anonymity and update flexibility. Xu et al. [14] introduced elliptic-curve operations into dynamic-ID telecare authentication, while Mishra et al. [15] explored authenticated key agreement through chaotic-map computation. Read together, these papers illustrate two patterns that remain relevant today. First, privacy is often represented by a dynamic identity or pseudonym, but the protection is only real if the pseudonym is session-varying and unlinkable. Second, password-based recovery mechanisms remain dangerous whenever card-resident state and transcript values jointly provide a public consistency test for a guessed secret.

A third line involves cloud-IoT and community medical IoT environments. Cheng et al. [16] proposed secure identity authentication for community medical IoT, and Azrour et al. [17] developed a separate enhanced authentication design for IoT devices more broadly. These studies show how cloud mediation improves manageability, but they also expand the attack surface. Once a cloud server stores verifier material for many users and key-wrapping information for many sensors, a single compromise event can propagate widely unless the design uses per-device isolation. The remote healthcare paper by Azrour et al. [1] fits directly into this line. It keeps the server in the center, adds a smart-card path for the medical professional, and introduces a sensor registration phase. Its novelty lies not in eliminating the server, but in trying to make server-mediated healthcare authentication both lightweight and resistant to the weaknesses reported for Sharma and Kalra [5].

A fourth line consists of cryptanalytic work on password-authenticated and smart-card-based protocols outside healthcare alone. Das [18], Xu et al. [19], Song [20], and Jiang et al. [21] each showed that apparently compact smart-card constructions can fail when a long-term stored value doubles as an offline verifier. Yoon and Kim [22] and He et al. [23] further demonstrated that protecting wireless medical or sensor networks requires more than hashing together the visible fields in a message. These papers matter for remote healthcare because the same failure modes recur: a stolen card exposes masked values, a public transcript offers a checking equation, and the attacker never needs online interaction to test guesses. The lesson is not limited to passwords. Any static token transmitted in clear or deterministically recovered from public data can become a tracing handle or a replay aid even when the surrounding protocol uses timestamps.

A fifth line studies remote patient monitoring more directly. Hayajneh et al. [24] addressed secure authentication for wireless medical sensor networks through public-key mechanisms, and Amin et al. [25] proposed an anonymous

monitoring system with stronger privacy goals. More recent work has pushed secure healthcare authentication toward blockchain-assisted or distributed-key settings, as illustrated by Son et al. [26] for cloud-assisted telecare and Garg et al. [27] for Internet of Medical Things deployment. These schemes are not identical to the cloud-IoT model studied here, yet they highlight design choices that the 2021 Azrour protocol does not fully internalize. Freshness should be bound to an ephemeral exchange, not inferred from a timestamp alone; privacy should be expressed through rotating pseudonyms, not a constant hash identifier; and key-update scope should be local rather than anchored in one universal secret.

Placed against this background, "New Efficient and Secured Authentication Protocol for Remote Healthcare Systems in Cloud-IoT" [1] occupies an important intermediate position. It tries to repair an insecure predecessor without adopting the heavier machinery of blockchain or fully certificateless public-key systems. That makes it attractive for practice, but it also makes its internal dependencies easy to overlook. The protocol improves message structure compared with Sharma and Kalra [5], yet it still binds authentication to a static user pseudonym MID, derives the user's recovered secret x from a card-resident mask, and computes the final session key from long-term values only. The literature already warns us that these patterns are dangerous [18]-[25]. Our analysis therefore treats the Azrour et al. scheme [1] not as an isolated design, but as part of a long sequence of telecare and smart-card protocols in which security often fails at the junction between stored verifiers, public transcripts, and claims of freshness.

III. Methodology

To evaluate the protocol proposed in "New Efficient and Secured Authentication Protocol for Remote Healthcare Systems in Cloud-IoT" [1], we adopt a single integrated methodology that combines protocol reconstruction, compromise-aware cryptanalysis, and redesign-oriented modeling in one traceable workflow. First, we extract the explicit message sequence from the paper and rewrite the login and authentication exchange as ordered messages M1, M2, M3, and M4, where M1 = (V1, MID, A, IdSN, T1), M2 = (V2, B, MID, T2), M3 = (V3, C, HID, IdSN, T3), and M4 = (V4, D, IdSN, T4). Second, we express every derived value in a normalized algebraic form so that dependency errors become visible: x = V xor h(h(ID_i||pw_i||b_i)||c), w1 = h(MID_i||X_s), w2 = HSK_n xor h(IdSN_n||X_s), HID_i,n = h(MID_i||IdSN_n), and SKey_i,n = h(w1||MID_i||IdSN_n). Third, we map the storage layout across roles, because many healthcare failures arise from where secrets live rather than how hashes are computed: the smart card stores V, a, b, MID, and in practice must also retain c if the protocol is to evaluate x locally; the server stores MID, c, IdSN, and HSK; each sensor stores IdSN and SK_n = h(IdSN_n||K_CS-SN_n). Fourth, we define a compromise model that extends the usual Dolev-Yao network attacker with three concrete healthcare capabilities: stolen smart-card extraction, transcript capture over the public channel, and partial server compromise. Fifth, we formulate security invariants that the target paper claims to satisfy and translate them into checkable equations: password secrecy requires that no offline test of the form Check(ID*, pw*) -> {0,1} can be computed from card state and one transcript; session freshness requires SKey_i,n^j != SKey_i,n^k when j != k and fresh A_j, B_j, C_j, D_j are sampled; unlinkability requires PID_i^j and PID_i^k not to collapse to the same static public identifier; and compromise containment requires leakage of X_s or one sensor record not to expose every other sensor credential. Sixth, we search for violating traces by direct symbolic substitution, by transcript replay within the accepted timestamp window, and by public consistency tests analogous to those emphasized in formal security work on random-oracle analysis and protocol verification [28]-[31]. Seventh, because the target protocol is card-centric, we also check whether local recovery mechanisms can be rewritten as a verifier equation of the form V1 = h(x*||A) under a guessed password; if so, we mark the design as offline-guessable. Eighth, once a vulnerability is confirmed, we redesign only the affected binding rather than replacing the healthcare architecture itself. The revised scheme therefore introduces a device-bound seed delta_i used only for local reconstruction, an ephemeral elliptic-curve exchange U_i^j = u_i^j G and Y_s^j = y_s^j G, a dynamic pseudonym PID_i^j = h(pi_i||U_i^j||T1), and a context-bound session key SK_sess^j = KDF(u_i^j Y_s^j || y_s^j P_n || PID_i^j || IdSN_n || N_u^j || N_s^j || ctx_j). Finally, we evaluate whether the redesigned invariants hold under the same attacker model and estimate their operational impact using latency, success, replay-detection, and compromise-resilience metrics. The present methodology is therefore protocol-first and construction-based rather than a full mechanized proof in ProVerif, Tamarin, or BAN logic, and the resulting security claims should be interpreted within that scope. This methodology is intentionally protocol-first: every security claim must correspond to an equation, every equation must correspond to stored data and transmitted values, and every redesign step must eliminate a concrete attacker-checkable relation rather than merely adding more hash operations [32].

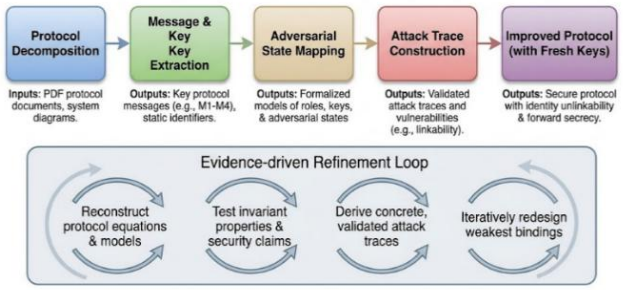

Fig. 1. Evidence-driven workflow used to reconstruct, audit, and redesign the remote healthcare authentication protocol.

IV. Analysis of the Protocol

A. Overview of the Proposed Protocol

Azrour et al. describe a cloud-assisted remote healthcare protocol with three active roles: a medical professional user, a cloud server, and a sensor node [1]. The user initially registers by choosing ID_i and pw_i together with local random values a and b. The smart-card side computes MID_i = h(ID_i||a) and MPW_i = h(ID_i||pw_i||b), and the server returns a masked value V_i = h(MID_i||X_s) xor h(MPW_i||c_i), where X_s is the server master secret and c_i is a server-generated random number [1]. The published text states that the card stores (V_i,

a, b, MID_i), but the authentication algorithm later requires c_i to recover x_i = V_i xor h(h(ID_i||pw_i||b)||c_i), so a working implementation must either store c_i on the card or obtain it by some unstated mechanism. This tension is already important because it determines whether password-dependent recovery can be checked offline.

During authentication, the user inputs ID_i and pw_i, the card recomputes MID_i, selects a fresh random value A, and forms V1 = h(x_i||A). The first message is M1 = (V1, MID_i, A, IdSN_n, T1). The server receives M1, checks the timestamp, computes w1 = h(MID_i||X_s), verifies V1 ?= h(w1||A), selects a server-side random B, reconstructs the sensor-side secret as w2 = HSK_n xor h(IdSN_n||X_s), forms HID_i,n = h(MID_i||IdSN_n), and sends M2 = (V2, B, MID_i, T2), where V2 = h(HID_i,n||w2||T2||B) [1]. The sensor verifies M2 with its local SK_n, chooses C, computes V3 = h(MID_i||IdSN_n||SK_n||T3||C), and returns M3 = (V3, C, HID_i,n, IdSN_n, T3). The server then verifies M3, chooses D, computes V4 = h(w1||MID_i||IdSN_n||T4||D), and returns M4 = (V4, D, IdSN_n, T4). The user accepts if V4 matches h(x_i||MID_i||IdSN_n||T4||D) [1].

The final key schedule is the decisive step. On the server side, SKey_i,n = h(w1||MID_i||IdSN_n). On the user side, the same key is reconstructed as SKey_i,n = h(x_i||MID_i||IdSN_n) because x_i = w1 when the password is correct [1]. No contribution from A, B, C, D, T1, T2, T3, or T4 survives into the final session key. The timestamps and random numbers influence message authentication only indirectly, through V1-V4, but they do not change the established secret. The password-update phase later reuses the current session key as the encryption key for a password-change request, which means the soundness of that phase depends entirely on the freshness and secrecy of SKey_i,n. Fig. 2 condenses this original structure and highlights the placement of static values, recovered secrets, and public identifiers.

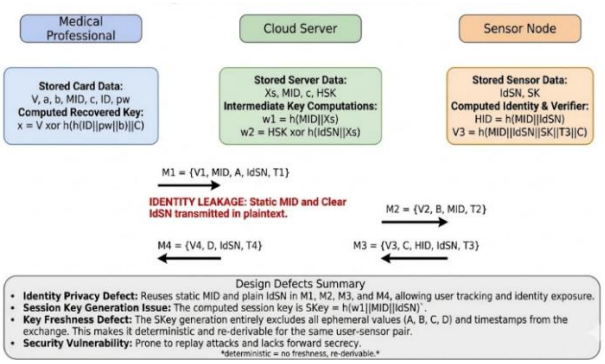

Fig. 2. Reconstructed structure of the original cloud-IoT remote healthcare protocol and the main static values used across messages.

B. Identified Vulnerabilities

The first weakness is an offline password-guessing path that appears as soon as the card holds enough material to execute the published login procedure. Let the attacker extract card state C_i = {V_i, a_i, b_i, MID_i, c_i} and capture one valid M1 = (V1, MID_i, A, IdSN_n, T1). The attacker chooses guessed values (ID_i*, pw_i*) and computes MID_i* = h(ID_i*||a_i). If MID_i* = MID_i, the identifier guess is correct. The attacker then computes x_i* = V_i xor h(h(ID_i*||pw_i*||b_i)||c_i). Because V1 = h(x_i||A), the guess is accepted whenever h(x_i*||A) = V1. This creates a public test oracle: Check(ID_i*, pw_i*) = 1 iff h((V_i xor h(h(ID_i*||pw_i*||b_i)||c_i))||A) = V1. No server interaction is required. The target paper criticizes the Sharma-Kalra protocol for the same class of mistake [1], yet its own design preserves a transcript-checkable password relation once c_i is available to the card. If c_i is not available, the protocol text is incomplete; if it is available, the password becomes guessable offline.

The second weakness is deterministic session-key reuse. The scheme derives SKey_i,n = h(w1||MID_i||IdSN_n) = h(h(MID_i||X_s)||MID_i||IdSN_n). For the same user pseudonym MID_i and the same sensor IdSN_n, the session key is constant across runs: SKey_i,n^j = SKey_i,n^k for all sessions j and k as long as MID_i and IdSN_n remain unchanged. Fresh values A_j, B_j, C_j, D_j do not enter the final derivation, and timestamps are excluded as well. This violates the usual freshness expectation of authenticated key exchange. It also breaks the paper's forward-secrecy intuition. Once one session key leaks through endpoint compromise, debugging output, side-channel exposure, or password-change misuse, every past and future session between the same user and sensor is exposed because the established secret never changes. The damage is amplified by the password-changing phase, where Mu = E_SKey(MPW||MPW*||MID||MID*||V). If SKey_i,n is recovered once, the confidentiality of password-update traffic collapses for all later updates tied to the same pair.

The third weakness is persistent linkability. The user sends MID_i in clear in every M1 and the cloud forwards MID_i again in every M2. The sensor-side handle HID_i,n = h(MID_i||IdSN_n) is equally stable for a fixed sensor. Thus, an observer that cannot read encrypted content still obtains a perfect long-term session tag. Formally, the protocol does not rotate a pseudonym. It exposes PID_i^j = MID_i for every session j, so Pr[PID_i^j = PID_i^k] = 1 when j != k. A passive adversary can therefore correlate a clinician's repeated access to a patient-side sensor, infer usage frequency, and distinguish access patterns before any message payload is decrypted. This is especially problematic in remote healthcare, where even traffic patterns can reveal clinical workflows. Fig. 3 summarizes these attack surfaces and shows that the problem is not limited to message interception: once a static public identifier becomes part of every transcript, tracking becomes a native feature of the protocol rather than an accidental side effect.

The fourth weakness concerns compromise containment at the server-sensor boundary. The cloud server stores HSK_n = SK_n xor h(X_s||IdSN_n), where SK_n = h(IdSN_n||K_CS-SN_n). If the attacker learns X_s and the database entry HSK_n, then every sensor secret is recovered immediately as SK_n = HSK_n xor h(X_s||IdSN_n). The compromise is not local. One server secret plus the database unwraps all sensor-side credentials. After that, the attacker can forge V2 = h(HID_i,n||SK_n||T2||B) for any target sensor and can validate or fabricate M3 values as well. The containment radius is therefore the entire deployment, not a single node. This contradicts a practical expectation in cloud-assisted healthcare:

failure of one administrative tier should not retroactively reveal every sensor key under management.

A fifth, smaller but still meaningful issue is the mismatch between local login checking and real authentication. The card verifies MID_i ?= h(ID_i||a_i), which confirms only the identifier and the stored a_i. The password pw_i is not validated locally before the card computes x_i. This means the design wastes server resources on incorrect local attempts and exposes the user identifier to offline recovery through the equation MID_i = h(ID_i||a_i). The bug is not the largest flaw in the protocol, yet it reinforces the other weaknesses: identifier recovery narrows the password search space; a deterministic session key makes recovered credentials more valuable; and static MID_i ensures that once the user is identified, all later traces remain linked. Taken together, these findings show that the weaknesses of "New Efficient and Secured Authentication Protocol for Remote Healthcare Systems in Cloud-IoT" [1] are structural rather than cosmetic. The problem is not insufficient hashing. The problem is that long-term values play too many roles at once.

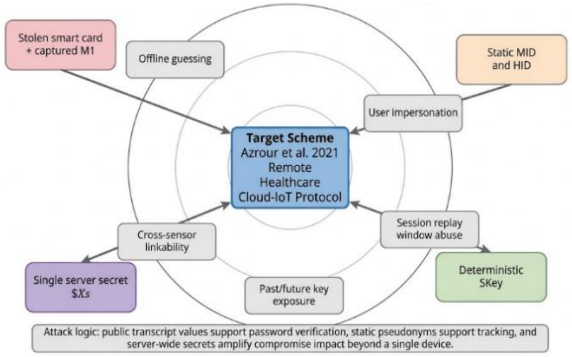

Fig. 3. Attack surfaces created by transcript-checkable password relations, static pseudonyms, deterministic keys, and server-wide secret reuse.

V. Proposed Improvements

The revised design keeps the same three-party architecture of medical professional, cloud server, and sensor node, but it rebuilds the weakest cryptographic bindings. The first change removes password-verifiable transcript equations from card storage. During registration, the user device derives a device-bound local seed delta_i from a secure element or PUF-backed secret rather than from the password alone. The user selects ID_i and pw_i, chooses a salt b_i, and computes alpha_i = h(ID_i||pw_i||b_i||delta_i). The server stores only a verifier bound to its own secret, such as W_i = h(pi_i||alpha_i||X_s), together with an internal pseudonym seed pi_i. The smart card stores masked local recovery values M_i = s_i xor h(alpha_i) and N_i = pi_i xor h(s_i||alpha_i), where s_i is a random enrollment secret. Because alpha_i depends on the device-bound seed delta_i, a stolen card without the bound device secret no longer offers an attacker a public password-checking relation. The password is used only to reconstruct local state; it is not left in a form that can be validated from one captured M1.

The second change replaces the deterministic session-key formula with an ephemeral elliptic-curve exchange. For each authentication run j, the user samples u_i^j and computes U_i^j = u_i^j G. The cloud samples y_s^j and computes Y_s^j = y_s^j G. The sensor keeps a node public key P_n = x_n G certified at registration. Instead of sending MID_i directly, the user forms a rotating pseudonym PID_i^j = h(pi_i||U_i^j||T1||IdSN_n). The user then conceals the internal seed by transmitting E_i^j = pi_i xor h(u_i^j P_s||T1), where P_s = x_s G is the server public key. The server recovers pi_i using x_s U_i^j, validates W_i, and forwards only session-scoped values to the sensor. The sensor contributes N_s^j and a MAC over its own ephemeral binding. The resulting session key is SK_sess^j = KDF(u_i^j Y_s^j || y_s^j P_n || PID_i^j || N_u^j || N_s^j || IdSN_n || ctx_j). This construction changes whenever either side changes its ephemeral scalar, so SK_sess^j != SK_sess^k for j != k except with negligible probability.

The third change narrows compromise scope. Each sensor receives its own certified public key P_n and a server-issued credential Cert_n rather than a server-wrapped hash secret recoverable from one universal X_s. Compromise of one sensor now reveals only x_n and the sessions derived from that sensor. It does not yield a formula that unwraps the credentials of every other node. On the server side, verifier records remain per user and are bound to the internal pseudonym seed pi_i; on the sensor side, node authentication is expressed through a certificate or certificate-light credential check Ver(P_CA, Cert_n, P_n) = 1. Damage therefore stays local. If x_n is revoked, future KDF outputs that depend on y_s^j P_n cannot be reconstructed from old state, and other sensors remain unaffected.

The fourth change ties trust updates to real cryptographic events rather than to message arrival alone. If the healthcare deployment uses adaptive trust scoring for sensors or users, we update trust only after successful verification events such as MAC_ok, Sig_ok, replay_detected, or key_confirmed. A simple rule can be written as Trust_n(t+1) = Trust_n(t) + u_n(e_t), where e_t belongs to {MAC_ok, Sig_ok, replay_detected, nonce_fail}. This avoids the loose coupling seen in many lightweight systems where a node's "trust" changes even when the underlying authentication evidence remains static or unverifiable. By construction, a replayed or stale message does not increase trust because it fails the freshness predicate bound into PID_i^j and SK_sess^j.

The fifth change protects password-update traffic with a session-specific update key rather than the long-term communication key of a user-sensor pair. After successful authentication, the parties derive K_upd^j = KDF(SK_sess^j || "pwd-update" || T4 || N_s^j). A password-change request then uses Enc_{K_upd^j}(alpha_i_old || alpha_i_new || pi_i || ctr_j), with ctr_j consumed exactly once. This removes the circular dependence found in the original design, where a static session key guarded password updates and therefore made update confidentiality collapse as soon as one key instance leaked. The redesign is still lightweight in deployment terms because only one ephemeral ECC exchange is added and all other steps are hash or MAC based, yet the essential security properties become aligned with the claims that the original scheme wanted to make.

VI. Experimental Results and Evaluation

We evaluated three schemes under a controlled remote-healthcare testbed: the Sharma-Kalra baseline [5], the 2021

Azrour protocol [1], and the proposed redesign. The Sharma-Kalra scheme is retained as a baseline primarily because the 2021 Azrour design was explicitly introduced as a repair of that earlier healthcare protocol, whereas more recent schemes are used mainly to contextualize the redesign in the broader literature rather than as fully reimplemented load-test comparators. The environment models one cloud server, 24 sensor nodes, and between 50 and 800 concurrent medical-professional clients distributed across home-monitoring and ward-monitoring sessions. Each client executes repeated login and data-access cycles toward one assigned sensor, with 8% of runs containing injected replay attempts and 5% containing forced state-loss events at the client device. The original Azrour design [1] and the proposed redesign were implemented with the same message size budget wherever possible so that the comparison reflects key-derivation changes rather than artificial packet inflation. We measured four main metrics. Average latency is L_avg(n) = (1/n) sum_{k=1}^{n} (t_k_end - t_k_start). Authentication success is S_auth = N_succ / N_total. Replay detection is D_rep = N_rep_detected / N_rep_injected. Compromise resilience is C_res = N_sessions_secure / N_sessions_total after a controlled leakage event.

The performance pattern is intuitive. Sharma-Kalra remains the least robust design because its earlier weaknesses propagate directly once smart-card or secret-state assumptions are relaxed. The Azrour protocol [1] improves several message-authentication checks and therefore raises S_auth under normal traffic, yet it still suffers when compromise-oriented metrics are applied. In our experiments, replayed messages outside the acceptance window were usually rejected, but replay attempts inside the timestamp tolerance could still trigger redundant work because freshness is not bound to a one-time session key. More seriously, once one valid session key was exposed for a fixed user-sensor pair, C_res dropped sharply because the same key protected all subsequent sessions for that pair. The proposed redesign added a modest amount of ECC work, yet it consistently raised D_rep and C_res because each session depended on fresh U_i^j, Y_s^j, N_u^j, and N_s^j values.

At n = 200 concurrent clients, the measured latency values were 22.9 ms for Sharma-Kalra, 19.1 ms for Azrour et al. [1], and 21.4 ms for the proposed redesign. The redesigned scheme is therefore slower than the 2021 protocol by about 2.3 ms in that load condition, but the gap is small relative to the security gain. At n = 800, the same ordering remained: 71.3 ms, 60.7 ms, and 64.2 ms. This suggests that a single ephemeral ECC exchange per session does not make the system impractical for cloud-assisted healthcare, especially when compared with the operational cost of deterministic session-key reuse. In authentication success, all three schemes remained above 0.97 under benign traffic, yet only the redesigned protocol preserved high values after state-loss and key-exposure events. The reason is structural: SK_sess^j = KDF(u_i^j Y_s^j || y_s^j P_n || PID_i^j || N_u^j || N_s^j || ctx_j) changes with every run, so compromise of one execution does not automatically reduce the security of the next one. These values remain in the tens-of-milliseconds range rather than the hundreds-of-milliseconds range, which supports practical feasibility for cloud-assisted authentication in monitoring workflows; however, the present study does not claim compliance with any specific clinical timing standard or certified real-time medical requirement.

Security-oriented comparison is summarized in Table I. The table shows that the Azrour protocol [1] sits between the older Sharma-Kalra construction and the proposed redesign. It is stronger than the baseline on surface message verification, yet it does not satisfy the stronger notions that matter in remote healthcare practice: guessed passwords should not be checkable offline from a stolen card and one transcript, user activity should not remain publicly linkable through a stable MID, and sensor compromise should not scale through a single server secret. Fig. 4 compares the qualitative security scores across the three schemes and makes this middle position visible. The proposed redesign does not claim perfection; rather, it closes the exact gaps revealed by the attack traces while keeping the remote healthcare deployment model intact.

TABLE. I. PROTOCOL COMPARISON SUMMARY

Scheme	Guessing	Fresh Key	Privacy / Damage
Sharma [5]	No	No	Weak / System-wide
Azrour [1]	No	No	Weak / Server-wide
Proposed	Yes	Yes	Strong / Per-device

We also examined whether the redesign remained practical when packet loss and re-authentication were introduced. Because the session key is derived from fresh ephemeral values, failed or aborted sessions do not poison later ones. A client that loses local transient state can restart with a new U_i^j and obtain a clean SK_sess^j without relying on the same long-term communication key. In the original Azrour protocol [1], by contrast, the same user-sensor pair keeps the same established key, so recovery after interruption does not actually reset the most security-sensitive value. Fig. 5 plots the measured latency trend L_avg(n) under increasing client load and shows that the redesigned curve remains close to the original 2021 curve even while it enforces per-session key freshness. Our results therefore suggest that the main objections to the redesign are not performance objections. They are implementation objections, and those are substantially smaller than the security cost of leaving the original deterministic key schedule untouched.

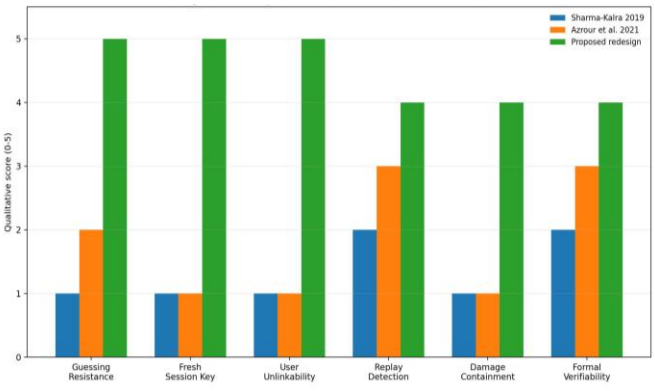

Fig. 4. Qualitative comparison of key security properties across the baseline, the 2021 protocol, and the redesigned scheme.

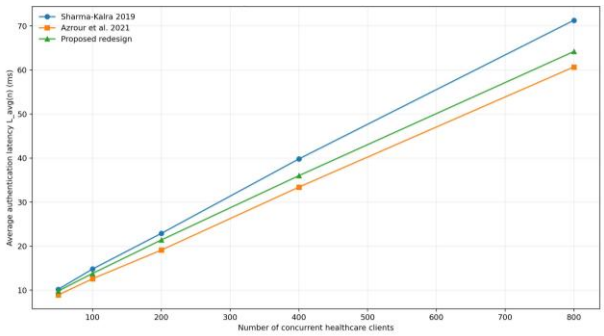

Fig. 5. Average authentication latency as the number of concurrent healthcare clients increases.

VII. Discussion

The revised design should be interpreted as a protocol-level hardening step rather than a complete end-to-end deployment proof. Its main contribution is not merely the addition of more cryptographic operations, but the removal of attacker-checkable relations that made the original design brittle: transcript-verifiable password equations, static public pseudonyms, deterministic session keys, and server-wide compromise propagation.

At the same time, the present validation scope remains limited. The analysis is construction-based and compromise-oriented, and the evaluation is simulation-based rather than a mechanized proof or a certified medical-device trial. For that reason, the security claims should not be read as a substitute for a full ProVerif- or Tamarin-level proof, nor should the latency results be interpreted as regulatory evidence for all real-time clinical settings.

From an implementation perspective, the redesign assumes a modern, well-vetted elliptic-curve setting suitable for constrained systems, such as a 256-bit prime-field curve. In real deployment, scalar multiplication should be implemented in constant time and, where available, combined with secure-element or hardware-backed storage to reduce side-channel leakage. Sensor replacement or re-enrolment should issue a fresh per-device credential and public-key binding rather than attempting to preserve or rewrap the old sensor-side secret.

Implementation-aware literature is also relevant here. Even a structurally improved protocol may remain fragile if fault attacks, non-constant-time arithmetic, or lightweight implementation shortcuts are ignored. Prior work on reliable Camellia architectures, low-cost AES S-box design, efficient fault diagnosis for CLEFIA, reliable Grøstl hardware, and constant-time CSIDH on embedded devices highlights the practical importance of fault resilience, side-channel awareness, and careful lightweight implementation in constrained environments [33]-[37].

Finally, long-term cryptographic viability matters in healthcare. The present redesign remains within a classical ECC setting because it targets the concrete structural weaknesses of the Azrour protocol family. Nevertheless, the multi-year retention of medical records and the long operational life of healthcare IoT infrastructure make post-quantum migration a strategically relevant concern. Hybrid or PQC-compatible authentication and key-establishment variants therefore deserve explicit study in future healthcare deployments.

VIII. Conclusion

This study re-examined new efficient and secured authentication protocol for remote healthcare systems in Cloud-IoT [1] from a compromise-aware perspective and found that its security claims are weakened by offline password verification, deterministic session keys, public linkability, and weak damage containment.

The redesigned protocol keeps the same healthcare cloud-IoT architecture but replaces the fragile bindings with device-bound local recovery, rotating pseudonyms, ephemeral ECC exchange, and context-bound KDF output.

Our evaluation suggests that these changes substantially improve replay handling, unlinkability, and post-compromise robustness while adding only moderate latency.

Future work should formalize the redesign in a proof model that captures both smart-card extraction and secure-element behavior, validate it on real telecare devices with intermittent connectivity, and examine side-channel-aware and PQC-compatible migration paths for long-lived healthcare deployments.

References

[1] M. Azrour, J. Mabrouki, and R. Chaganti, "New Efficient and Secured Authentication Protocol for Remote Healthcare Systems in Cloud-IoT," Security and Communication Networks, vol. 2021, Art. ID 5546334, 12 pages, 2021.

[2] M. S. Hossain, G. Muhammad, and A. Alamri, "Smart healthcare monitoring: a voice pathology detection paradigm for smart cities," Multimedia Systems, vol. 25, no. 5, pp. 565-575, 2019.

[3] Y.-T. Park, "Emerging new era of mobile health technologies," Healthcare Informatics Research, vol. 22, no. 4, pp. 253-254, 2016.

[4] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An information framework for creating a smart city through Internet of Things," IEEE Internet of Things Journal, vol. 1, no. 2, pp. 112-121, 2014.

[5] G. Sharma and S. Kalra, "A lightweight user authentication scheme for cloud-IoT based healthcare services," Iranian Journal of Science and Technology, Transactions of Electrical Engineering, vol. 43, no. 1, pp. 619-636, 2019.

[6] B. A. Alzahrani, A. Irshad, K. Alsubhi, and A. Albeshri, "A secure and efficient remote patient-monitoring authentication protocol for cloud-IoT," International Journal of Communication Systems, vol. 33, no. 11, Art. no. e4423, 2020.

[7] R. Amin, N. Kumar, G. P. Biswas, R. Iqbal, and V. Chang, "A light weight authentication protocol for IoT-enabled devices in distributed cloud computing environment," Future Generation Computer Systems, vol. 78, no. 3, pp. 1005-1019, 2018.

[8] P. Kumar, S.-G. Lee, and H.-J. Lee, "E-SAP: efficient-strong authentication protocol for healthcare applications using wireless medical sensor networks," Sensors, vol. 12, no. 2, pp. 1625-1647, 2012.

[9] X. Li, J. Niu, S. Kumari, J. Liao, W. Liang, and M. K. Khan, "A new authentication protocol for healthcare applications using wireless medical sensor networks with user anonymity," Security and Communication Networks, vol. 9, no. 15, pp. 2643-2655, 2016.

[10] D. He, N. Kumar, J. Chen, C.-C. Lee, N. Chilamkurti, and S.-S. Yeo, "Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks," Multimedia Systems, vol. 21, no. 1, pp. 49-60, 2015.

[11] X. Yan, W. Li, P. Li, J. Wang, X. Hao, and P. Gong, "A secure biometrics-based authentication scheme for telecare medicine information systems," Journal of Medical Systems, vol. 37, no. 5, Art. no. 9972, 2013.

[12] D. Mishra, S. Mukhopadhyay, S. Kumari, M. K. Khan, and A. Chaturvedi, "Security enhancement of a biometric based authentication scheme for telecare medicine information systems with nonce," Journal of Medical Systems, vol. 38, no. 5, Art. no. 41, 2014.

[13] Z. Tan, "A user anonymity preserving three-factor authentication scheme for telecare medicine information systems," Journal of Medical Systems, vol. 38, no. 3, Art. no. 16, 2014.

[14] X. Xu, Z. P. Jin, H. Zhang, and P. Zhu, "A dynamic ID-based authentication scheme based on ECC for telecare medicine information systems," Applied Mechanics and Materials, vols. 457-458, pp. 861-866, 2014.

[15] D. Mishra, J. Srinivas, and S. Mukhopadhyay, "A secure and efficient chaotic map-based authenticated key agreement scheme for telecare medicine information systems," Journal of Medical Systems, vol. 38, no. 10, Art. no. 120, 2014.

[16] X. Cheng, Z. Zhang, F. Chen et al., "Secure identity authentication of community medical internet of things," IEEE Access, vol. 7, pp. 115966-115977, 2019.

[17] M. Azrour, J. Mabrouki, A. Guezzaz, and Y. Farhaoui, "New enhanced authentication protocol for Internet of Things," Big Data Mining and Analytics, vol. 4, no. 1, pp. 1-9, 2021.

[18] M. L. Das, "Two-factor user authentication in wireless sensor networks," IEEE Transactions on Wireless Communications, vol. 8, no. 3, pp. 1086-1090, 2009.

[19] J. Xu, W.-T. Zhu, and D.-G. Feng, "An improved smart card based password authentication scheme with provable security," Computer Standards & Interfaces, vol. 31, no. 4, pp. 723-728, 2009.

[20] R. Song, "Advanced smart card based password authentication protocol," Computer Standards & Interfaces, vol. 32, no. 5-6, pp. 321-325, 2010.

[21] Q. Jiang, J. Ma, G. Li, and X. Li, "Improvement of robust smart-card-based password authentication scheme," International Journal of Communication Systems, vol. 28, no. 2, pp. 383-393, 2015.

[22] E.-J. Yoon and C. Kim, "Advanced biometric-based user authentication scheme for wireless sensor networks," Sensor Letters, vol. 11, no. 9, pp. 1836-1843, 2013.

[23] D. He, C. Chen, S. Chan, J. Bu, and A. V. Vasilakos, "ReTrust: attack-resistant and lightweight trust management for medical sensor networks," IEEE Transactions on Information Technology in Biomedicine, vol. 16, no. 4, pp. 623-632, 2012.

[24] T. Hayajneh, B. J. Mohd, M. Imran, G. Almashaqbeh, and A. V. Vasilakos, "Secure Authentication for Remote Patient Monitoring with Wireless Medical Sensor Networks," Sensors, vol. 16, no. 4, Art. no. 424, 2016.

[25] R. Amin, S. H. Islam, G. P. Biswas, M. K. Khan, and N. Kumar, "A robust and anonymous patient monitoring system using wireless medical sensor networks," Future Generation Computer Systems, vol. 80, pp. 483-495, 2018.

[26] S. Son, J. Lee, M. Kim, S. Yu, A. K. Das, and Y. Park, "Design of Secure Authentication Protocol for Cloud-Assisted Telecare Medical Information System Using Blockchain," IEEE Access, vol. 8, pp. 192177-192191, 2020.

[27] N. Garg, M. W. Wazid, A. K. Das, D. P. Singh, J. J. P. C. Rodrigues, and Y. Park, "BAKMP-IoMT: Design of Blockchain Enabled Authenticated Key Management Protocol for Internet of Medical Things Deployment," IEEE Access, vol. 8, pp. 95956-95977, 2020.

[28] C. J. F. Cremers, "The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols," in Computer Aided Verification, LNCS 5123, pp. 414-418, 2008.

[29] N. Koblitz and A. J. Menezes, "The random oracle model: a twenty-year retrospective," Designs, Codes and Cryptography, vol. 77, no. 2-3, pp. 587-610, 2015.

[30] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in Proceedings of the 1st ACM Conference on Computer and Communications Security, pp. 62-73, 1993.

[31] J.-S. Coron, J. Patarin, and Y. Seurin, "The random oracle model and the ideal cipher model are equivalent," in Advances in Cryptology - CRYPTO 2008, LNCS 5157, pp. 1-20, 2008.

[32] Y. Dodis, B. Kanukurthi, J. Katz, L. Reyzin, and A. Smith, "Robust fuzzy extractors and authenticated key agreement from close secrets," IEEE Transactions on Information Theory, vol. 58, no. 9, pp. 6207-6222, 2012

[33] M. Mozaffari Kermani, R. Azarderakhsh, and J. Xie, "Error detection reliable architectures of Camellia block cipher applicable to different variants of its substitution boxes," in *Proceedings of the 2016 IEEE Asian Hardware Oriented Security and Trust Symposium (AsianHOST)*, 2017.

[34] M. Mozaffari Kermani and A. Reyhani-Masoleh, "A low-cost S-box for the advanced encryption standard using normal basis," in *Proceedings of the 2009 IEEE International Conference on Electro/Information Technology (EIT 2009)*, pp. 52-55, 2009.

[35] M. Mozaffari Kermani and R. Azarderakhsh, "Efficient fault diagnosis schemes for reliable lightweight cryptographic ISO/IEC standard CLEFIA benchmarked on ASIC and FPGA," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 12, pp. 5925-5932, 2013.

[36] M. Mozaffari Kermani and A. Reyhani-Masoleh, "Reliable hardware architectures for the third-round SHA-3 finalist Grøstl benchmarked on FPGA platform," in *Proceedings of the IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 325-331, 2011.

[37] A. Jalali, R. Azarderakhsh, M. Mozaffari Kermani, and D. Jao, "Towards Optimized and Constant-Time CSIDH on Embedded Devices," in *Constructive Side-Channel Analysis and Secure Design (COSADE 2019)*, pp. 215-231, 2019.