Secure and Efficient Anonymous Authentication for Distributed Healthcare Systems Handling Medical Big Data

National Institute of Technology, Warangal

October 2, 2024

Presented by:

Adya Anant	**21CSB0A02**
Tanu Priya	**21CSB0A58**
Natasha Jha	**21CSB0F06**

Under the Guidance of:

Balaprakash Rao Killi

Introduction

- Big data in smart healthcare comes from interconnected IoT devices.
- **Main challenges:** Unauthorized access, data manipulation, and cyberattacks.
- Ensuring **CIA Triad** (Confidentiality, Integrity, Availability) is crucial.
- Vulnerabilities exist due to insecure communication and lack of robust security.
- **Mutual authentication** is needed for authorized access.
- Existing solutions are often too heavy for IoT environments.
- **Aim:** Propose a lightweight and anonymous mutual authentication scheme to secure medical big data.

Problem Statement

- **Unauthorized access** to sensitive medical data.
- Risk of data **interception and tampering** in transmission.
- Resource-constrained IoT devices struggle with **heavy security protocols.**
- Need for user anonymity with **secure mutual authentication.**

Lightweight Authentication Scheme

- The system model includes four entities: **IoT devices, gateway, Central Administrator (CA), and users.**
- **Aim:** Provide a secure session between an authenticated user and IoT device.
- **Key operations:** XOR, hash functions, timestamp verification, and secure key exchange.

System Model and Design Goals

System Model:

- **IoT Devices:** Collect and transmit patient data via a gateway.
- **Gateway:** Intermediary between IoT devices and users.
- **Central Administrator (CA):** Manages registration and authentication.
- **Users:** Access data, such as doctors or healthcare workers.

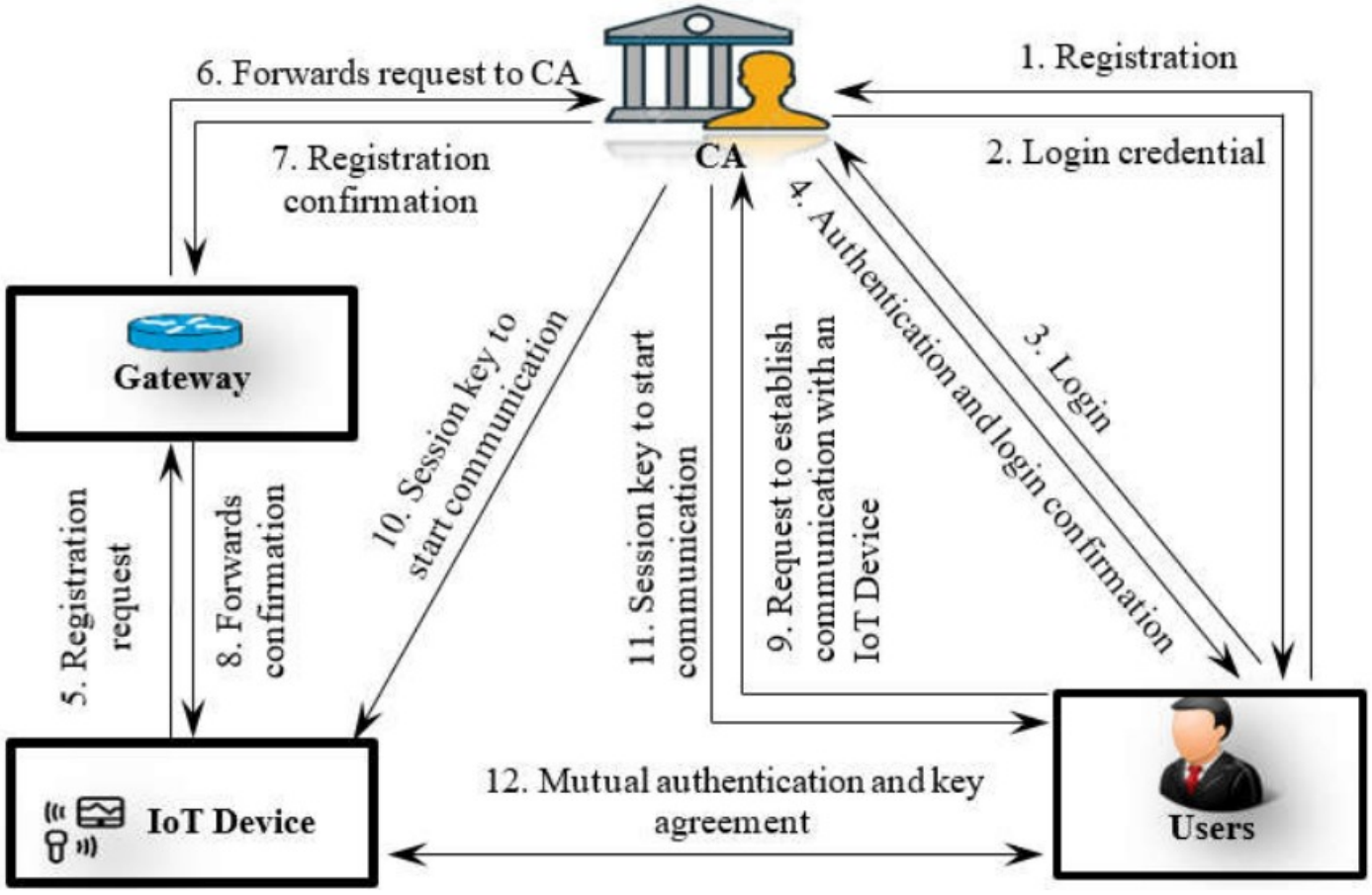
Design Goals:

- Mutual authentication
- Message integrity
- Identity anonymity
- Lightweight design

Key Algorithms in the Proposed Scheme

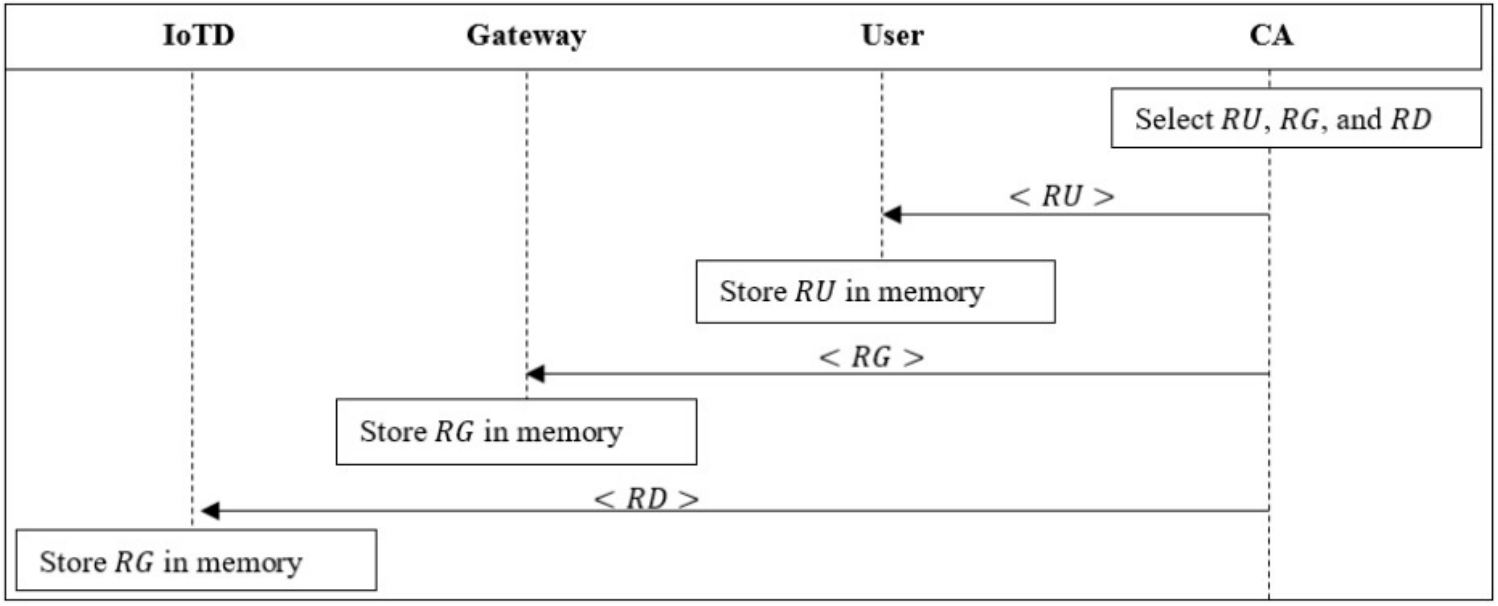
1. **System Initialization**: Central Administrator generates and distributes secret keys to users and devices.
2. **User Registration**: Users authenticate with CA using a unique ID, password, and temporary identity.
3. **Device Registration**: IoT devices register with the CA through the gateway and receive a secure identity.
4. **Login and Authentication**: Secure mutual authentication between the user and IoT device before any data exchange.

Workflow of the Proposed Scheme

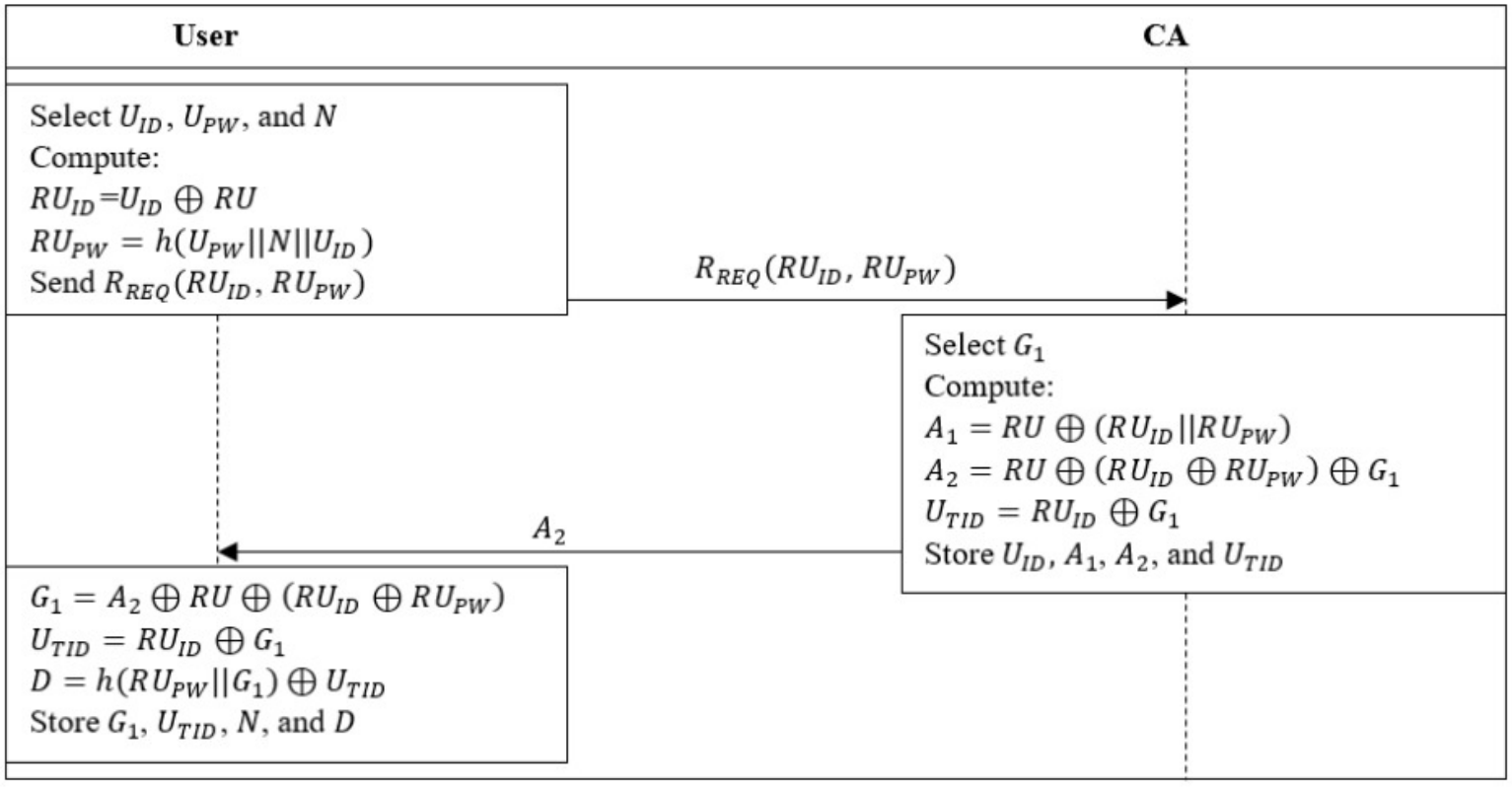


Proposed Work

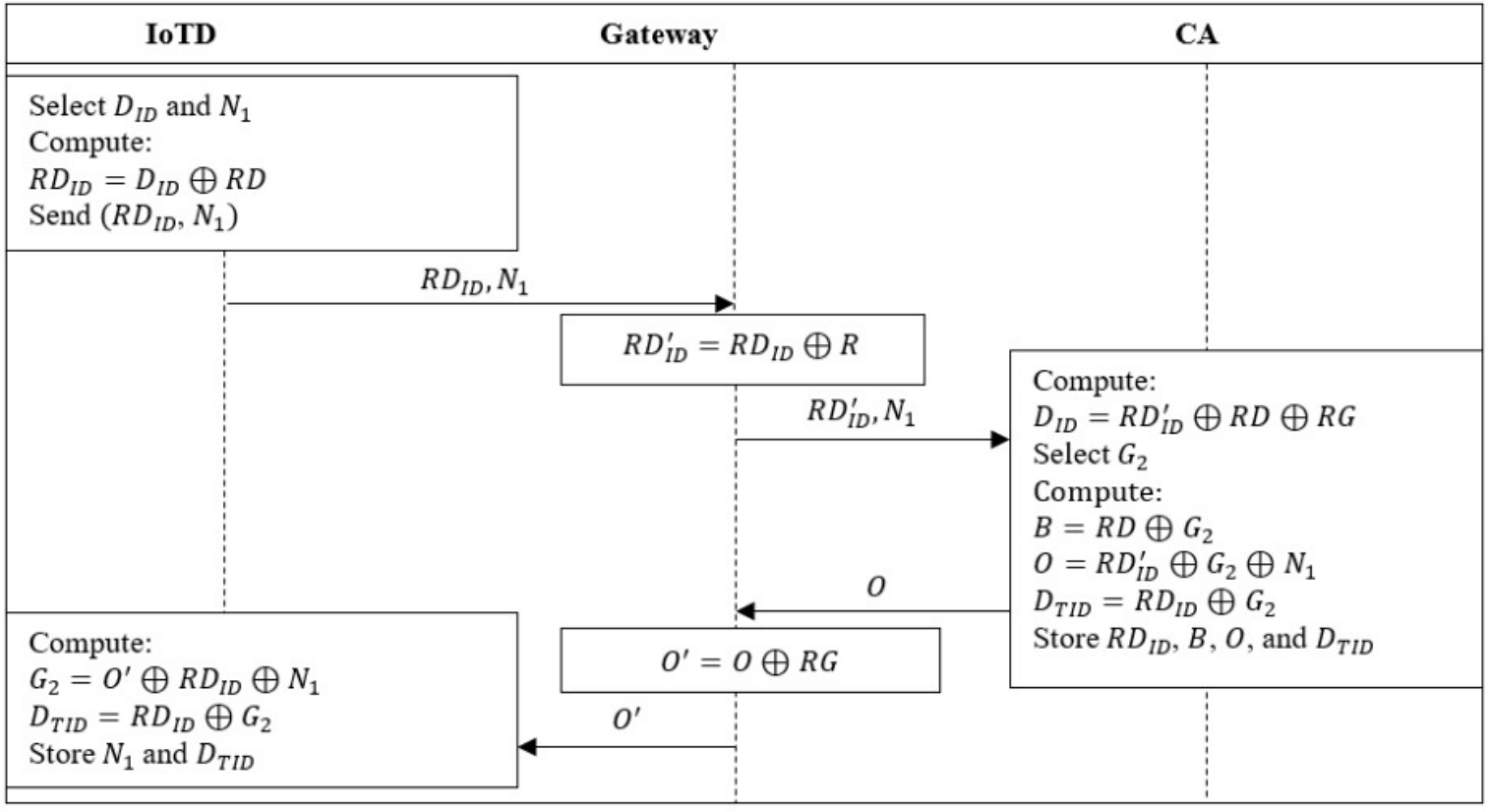
- **System Initialization**



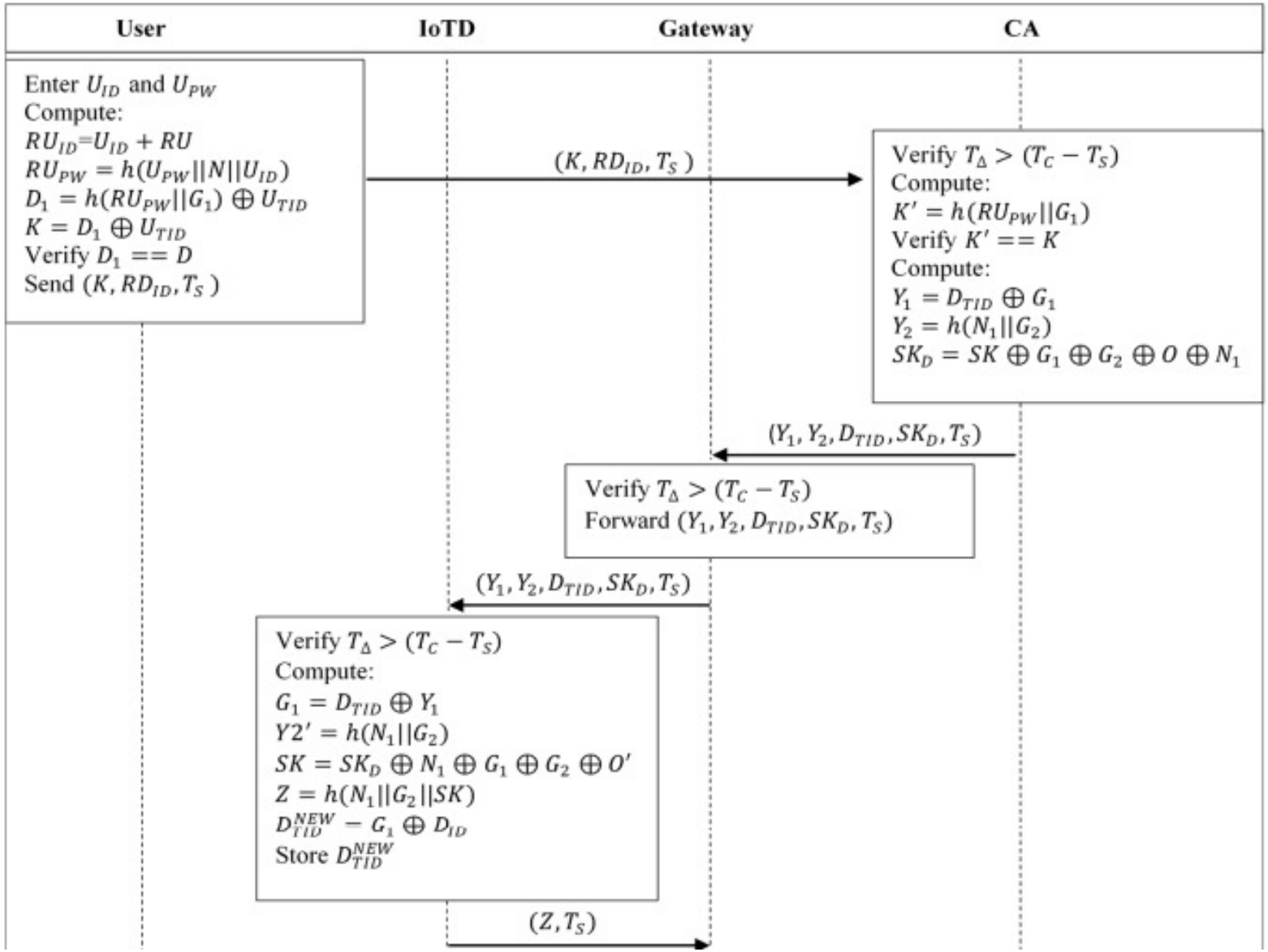
User Registration

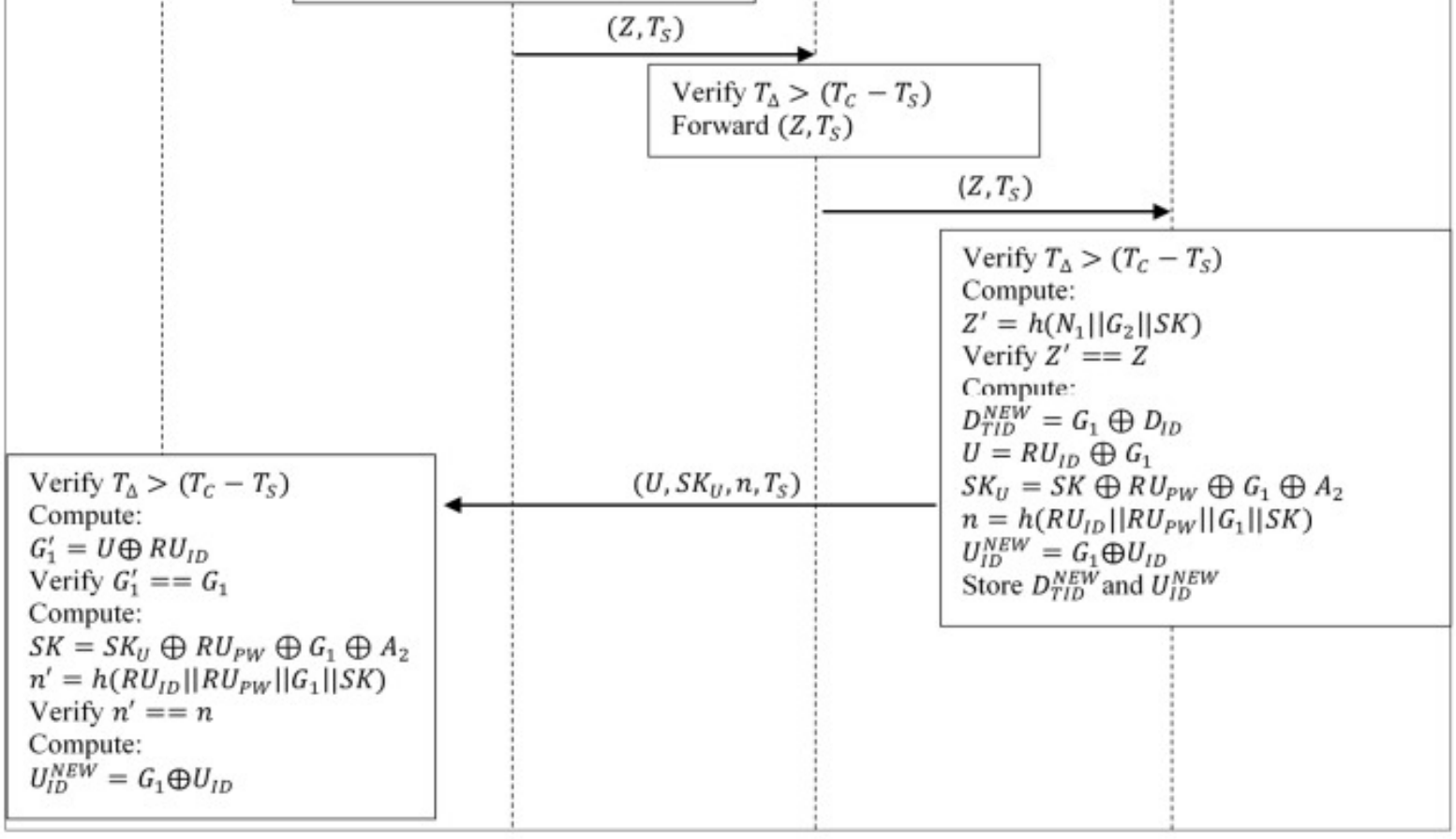


Device Registration



Login and Authentication




(Z, T_S)
Verify $T_\Delta > (T_C - T_S)$
Forward (Z, T_S)
(Z, T_S)
Verify $T_\Delta > (T_C - T_S)$
Compute:
$Z' = h(N_1||G_2||SK)$
Verify $Z' == Z$
Compute:
$D_{TID}^{NEW} = G_1 \oplus D_{ID}$
$U = RU_{ID} \oplus G_1$
$SK_U = SK \oplus RU_{PW} \oplus G_1 \oplus A_2$
$n = h(RU_{ID}||RU_{PW}||G_1||SK)$
$U_{ID}^{NEW} = G_1 \oplus U_{ID}$
Store D_{TID}^{NEW} and U_{ID}^{NEW}
(U, SK_U, n, T_S)
Verify $T_\Delta > (T_C - T_S)$
Compute:
$G_1' = U \oplus RU_{ID}$
Verify $G_1' == G_1$
Compute:
$SK = SK_U \oplus RU_{PW} \oplus G_1 \oplus A_2$
$n' = h(RU_{ID}||RU_{PW}||G_1||SK)$
Verify $n' == n$
Compute:
$U_{ID}^{NEW} = G_1 \oplus U_{ID}$

Informal Security Analysis

- **Replay Attacks:** Messages include a timestamp to prevent attackers from replaying previous session data.
- **Eavesdropping:** Even if an adversary intercepts messages, the secret session key remains secure, ensuring that the attacker cannot obtain useful information.
- **Privileged Insider Attacks:** Attackers with access to credentials can't compute session keys as they lack secret parameters (RU and G_1).
- **Stolen Verification Attack:** Capturing verification parameters is futile without access to the secret key, which is required to generate valid session keys.
- **Anonymity and Untraceability:** The attacker cannot trace the user's identity or link activities, as the user's credentials are protected by secret keys.

Formal Proofs

The base formula for the adversary's advantage is given as:

$$\text{Adv}(n) = |2 \cdot \text{Prob}[\text{Suc}(n)] - 1|$$

In **GAME 0**, no adversary queries are executed, making the chance of A breaking the scheme negligible. Thus, the probability of success is akin to a random guess:

$$\text{Prob}[\text{Suc}(G_0)] = \frac{1}{2} + \text{neg}(n)$$

Therefore, substituting in the formula:

$$\text{Adv}(G_0) = \text{neg}(n)$$

In **GAME 1**, the adversary performs eavesdropping queries and intercepts communications between entities. However, A cannot access random secret values stored with the user and the gateway such as k_u, k_{IoTD}, R_1, and R_2. Thus, the probability of success remains close to Game 0:

$$\text{Prob}[\text{Suc}(G_1)] = \text{Prob}[\text{Suc}(G_0)] = \frac{1}{2} + \text{neg}(n)$$

Therefore, substituting in the formula:

$$\text{Adv}(G_1) = \text{neg}(n)$$

In **GAME 2**, the adversary can perform the send query send(A, M) to send manipulated messages and observe responses. However, the scheme resists these attacks.
Per **Zipf's Law**: Frequent responses may be predictable, but others are harder to guess. The adversary might exploit patterns, yet the scheme ensures no meaningful insights into secret values (e.g., session keys, random numbers) are obtained.

$$\text{Prob}[\text{Suc}(G_2)] - \text{Prob}[\text{Suc}(G_1)] \leq \frac{Q_{snd}}{2^f}$$

Therefore:

$$\text{Prob}[\text{Suc}(G_2)] = \frac{Q_{snd}}{2^f} + \frac{1}{2} + \text{neg}(n)$$

Substituting in the base formula, we get:

$$\mathbf{Adv}(G_2) = \left| 2 \cdot \left(\frac{Q_{snd}}{2^f} + \frac{1}{2} + \mathbf{neg}(n) \right) - 1 \right|$$

Simplifying:

$$\mathrm{Adv}(G_2) = \frac{Q_{snd}}{2^{f-1}}$$

In **GAME 3**, A performs the $\mathrm{Hash}(\_, M)$ query, deriving specific values. According to the birthday paradox, the following probabilities are obtained:

1. **Hash Outputs:** The hash function produces n-bit outputs, giving 2^n possible values.
2. **First Query:** The first query always results in a unique hash value:

$$P_1 = 1$$

3. **Second Query:** The probability of a distinct value for the second query is:

$$P_2 = \frac{(2^n) - 1}{2^n}$$

4. **Third Query:** The probability for the third query is:

$$P_3 = \frac{(2^n) - 2}{2^n}$$

5. q_H **Queries:** The probability that q_H hash queries produce distinct values is:

$$P(\text{no collision}) = 1 \cdot \frac{(2^n) - 1}{2^n} \cdot \frac{(2^n) - 2}{2^n} \cdot \ldots \cdot \frac{(2^n) - (q_H - 1)}{2^n}$$

Simplifying the Product: Using the approximation $1 - x \approx e^{-x}$ for small values of x, we can approximate each term as:

$$\frac{(2^n) - k}{2^n} \approx \exp\left(\frac{-k}{2^n}\right)$$

So, the entire product becomes:

$$P(\text{no collision}) \approx \exp\left(-\sum_{k=0}^{q_H - 1} \frac{k}{2^n}\right)$$

Sum of k: The sum of the first $q_H - 1$ integers is:

$$\sum_{k=0}^{q_H-1} k = \frac{q_H(q_H-1)}{2}$$

Therefore, the probability of no collision becomes:

$$P(\text{no collision}) \approx \exp\left(-\frac{q_H^2}{2 \cdot 2^n}\right)$$

Probability of a Collision: The probability of at least one collision (which is the complement of no collision) is:

$$P(\text{collision}) = 1 - P(\text{no collision})$$

Using the approximation for $P(\text{no collision})$:

$$P(\text{collision}) \approx \frac{q_H^2}{2 \cdot 2^n}$$

Thus:

$$\text{Prob}[\text{Suc}(G_3)] - \text{Prob}[\text{Suc}(G_2)] \leq \frac{Q_{hsh}^2}{2^{f+1}}$$

Therefore:

$$\text{Prob}[\text{Suc}(G_3)] \leq \frac{Q_{hsh}^2}{2^{f+1}} + \frac{Q_{snd}}{2^f} + \frac{1}{2} + \text{neg}(n)$$

Substituting in the base formula, we get:

$$\text{Adv}(G_3) = \left| 2 \cdot \left(\frac{Q_{hsh}^2}{2^{f+1}} + \frac{Q_{snd}}{2^f} + \frac{1}{2} + \text{neg}(n) \right) - 1 \right|$$

Simplifying:

$$\text{Adv}(G_3) = \frac{Q_{hsh}^2}{2^f} + \frac{Q_{snd}}{2^{f-1}}$$

In **GAME 4**, the adversary can perform a capture query, accessing secret parameters from entities in the system. This tests the system's resilience to memory exposure attacks.
Adversary Capabilities:

- Send queries to interact with the system.
- Capture stored secrets from IoTD, Gateway, or User, but the system limits exposure by decentralizing critical information.

$$\max \left(D' \cdot Q_{snd}, \frac{Q_{snd}^b}{2^f} \right)$$

This term describes the adversary's success, where D' is a system-specific constant, and Q_{snd} represents the number of adversary queries.

Attack Methods:

1. **Brute-force attack:** The adversary attempts to guess the f-bit secret key with a success probability of $\frac{1}{2^f}$.

2. **Node capture attack:** The adversary gains access to secret information by capturing nodes. The success probability is $D' \cdot Q_{snd}$.

   The adversary's total advantage comes from the maximum of these two methods:

$$\text{Prob}[\text{Suc}(G_4)] - \text{Prob}[\text{Suc}(G_3)] \leq \max\left(D' \cdot Q_{snd}, \frac{Q_{snd}^b}{2^f}\right)$$

   Thus:

$$\textbf{Prob}[\textbf{Suc}(G_4)] \leq \frac{Q_{hsh}^2}{2^{f+1}} + \frac{Q_{snd}}{2^f} + \frac{1}{2} + \textbf{neg}(n) + \max\left(D' \cdot Q_{snd}, \frac{Q_{snd}^b}{2^f}\right)$$

Substituting in the base formula, we get:

$$\text{Adv}(G_4) = \left| 2 \cdot \left(\frac{Q_{hsh}^2}{2^{f+1}} + \frac{Q_{snd}}{2^f} + \frac{1}{2} + \text{neg}(n) + \max\left(D' \cdot Q_{snd}, \frac{Q_{snd}^b}{2^f} \right) \right) - 1 \right|$$

Simplifying:

$$\text{Adv}(G_4) = \frac{Q_{hsh}^2}{2^f} + \frac{Q_{snd}}{2^{f-1}} + \max\left(D' \cdot Q_{snd}, \frac{Q_{snd}^b}{2^f} \right) + \frac{1}{2} + \text{neg}(n)$$

In **GAME 5**, the adversary simulates a legitimate user by executing hash queries and guessing passwords or keys. The success rate depends on hash queries (Q_{hsh}) and is limited by the bit-length f, leveraging the birthday paradox.
In a second preimage attack, the adversary, given x_1, seeks x_2 such that:

$$h(x_1) = h(x_2)$$

Context: This attack targets protocols where the adversary must find an alternative input that hashes to the same value.
Success Probability: The success is bounded by hash queries, following the birthday bound:

$$\frac{Q_{hsh}^2}{2^f}$$

Therefore:

$$\text{Prob}[\text{Suc}(G_5)] - \text{Prob}[\text{Suc}(G_4)] \leq \frac{Q_{hsh}^2}{2^f}$$

Thus:

$$\text{Prob}[\text{Suc}(G_5)] \leq \frac{Q_{hsh}^2}{2^{f+1}} + \frac{Q_{snd}}{2^f} + \frac{1}{2} + \text{neg}(n) + \max\left(D' \cdot Q_{snd}, \frac{Q_{snd}^b}{2^f}\right) + \frac{Q_{hsh}^2}{2^f}$$

Substituting in the base formula, we get:

$$\text{Adv}(G_5) = \left|2 \cdot \left(\frac{Q_{hsh}^2}{2^{f+1}} + \frac{Q_{snd}}{2^f} + \frac{1}{2} + \text{neg}(n) + \max\left(D' \cdot Q_{snd}, \frac{Q_{snd}^b}{2^f}\right) + \frac{Q_{hsh}^2}{2^f}\right)\right.$$

Simplifying:

$$\text{Adv}(G_5) = 2 \cdot \max\left(D' \cdot Q_{snd}, \frac{Q_{snd}^b}{2^f}\right) + \frac{Q_{snd}}{2^{f-1}} + \frac{3Q_{hsh}^2}{2^f} + \frac{1}{2} + \text{neg}(n)$$

ALTERNATIVE METHOD FOR FORMAL PROOF

The probability of success is:

$$\text{Prob}[\text{Suc}(G_6)] - \text{Prob}[\text{Suc}(G_5)] \leq \frac{Q_{hsh}^2}{2^{f-1}}$$

Finally:

$$\frac{1}{2}\text{Adv} = |\text{Prob}[\text{Suc}(G_0)] - \text{Prob}[\text{Suc}(G_6)]| \quad (1)$$

$$= \sum_{i=0}^{5} |\text{Prob}[\text{Suc}(G_{i+1})] - \text{Prob}[\text{Suc}(G_i)]| \quad (2)$$

Therefore:

$$\text{Adv}_A^{\xi} \leq 2 \max \left\{ D' \cdot q_{snd}^b, \frac{q_{snd}}{2^f} \right\} + \frac{q_{snd}}{2^{f-2}} + \frac{3q_{hsh}^2}{2^{f-1}}$$

Formal Security Analysis (ROR Model)

- **Session Key Verification:** Uses the Real-or-Random (ROR) model.
- **Forward Secrecy:** Ensures secrecy of past session keys.
- **Negligible Compromise Probability:** Extremely low risk of session key compromise.
- **Offline Password Guessing:** Prevents offline guessing attacks.
- **Replay Attacks:** Protects against replay attacks.

Formal Verification with Verifpal

- **Result:** Scheme is **SAFE** against attacks such as MITM, replay, and eavesdropping.

```
Result • authentication? Iotd -> Ca: sk — When:
         a1 → HASH(CONCAT(CONCAT(uid, ru), HASH(CONCAT(upw, n, uid))))
         a2 → CONCAT(HASH(CONCAT(CONCAT(uid, ru), HASH(CONCAT(upw, n, uid)))), g1)
         utid → CONCAT(CONCAT(uid, ru), g1)
         xi → CONCAT(ru, CONCAT(uid, ru))
         yi → HASH(CONCAT(HASH(CONCAT(upw, n, uid)), g1))
         d → CONCAT(HASH(CONCAT(HASH(CONCAT(upw, n, uid)), g1)), CONCAT(CONCAT(uid, ru), g1))
         rd_prime → CONCAT(CONCAT(did, rd), rg)
         o → CONCAT(CONCAT(CONCAT(did, rd), rg), CONCAT(g2, n1))
         dtid → CONCAT(CONCAT(did, rd), g2)
         o_prime → CONCAT(CONCAT(CONCAT(CONCAT(did, rd), rg), CONCAT(g2, n1)), rg)
         ci → CONCAT(CONCAT(did, rd), n1)
         ei → HASH(CONCAT(HASH(CONCAT(upw, n, uid)), g1))
         d1 → CONCAT(HASH(CONCAT(HASH(CONCAT(upw, n, uid)), g1)), CONCAT(CONCAT(uid, ru), g1))
         k → CONCAT(CONCAT(HASH(CONCAT(HASH(CONCAT(upw, n, uid)), g1)), CONCAT(CONCAT(uid, ru), g1)), CONCAT(CONCAT(uid, ru), g1))
         k_prime → HASH(CONCAT(HASH(CONCAT(upw, n, uid)), g1))
         y1 → CONCAT(CONCAT(CONCAT(did, rd), g2), g1)
         skd → HASH(CONCAT(sk, g1, g2, CONCAT(CONCAT(CONCAT(did, rd), rg), CONCAT(g2, n1)), n1)) ← obtained by Attacker
         u → CONCAT(CONCAT(uid, ru), g1)
         sku → HASH(CONCAT(sk, HASH(CONCAT(upw, n, uid)), g1, CONCAT(HASH(CONCAT(CONCAT(uid, ru), HASH(CONCAT(upw, n, uid)))), g1)))
         g1_prime → CONCAT(CONCAT(CONCAT(uid, ru), g1), CONCAT(uid, ru))
         fi → HASH(CONCAT(HASH(CONCAT(upw, n, uid)), g1, CONCAT(HASH(CONCAT(CONCAT(uid, ru), HASH(CONCAT(upw, n, uid)))), g1)))
         n_prime → HASH(CONCAT(CONCAT(uid, ru), HASH(CONCAT(upw, n, uid)), g1, sk))
        sk (sk), sent by Ca and not by Iotd, is successfully used in HASH(CONCAT(sk, g1, g2, CONCAT(CONCAT(CONCAT(did, rd), rg), CONCAT(g2, n1)), n1)) within Ca's state.


Verifpal • Thank you for using Verifpal.
```

Storage, Computation Costs, and Execution Time Analysis

- **Storage:** Proposed scheme requires 2432 bits, which is lower than other schemes.
- **Computation Cost:** Requires only 9 hash operations, reducing processing time.
- **Efficiency:** Well-suited for IoT environments with limited resources.
- **Execution Time:**
  - Registration and authentication phases are optimized for time efficiency.
  - The proposed scheme takes 0.0036 ms for login and authentication.
  - It outperforms existing schemes that take up to 0.0144 ms.

Conclusion

- The proposed lightweight mutual authentication scheme ensures the security of medical big data.
- Suitable for distributed IoT-enabled healthcare systems.
- Provides improved performance, security, and efficiency compared to existing solutions.
- Future work: Explore multi-factor authentication and scalability improvements.