STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **x**, Month 202x, pp 0–15.
Published online in International Academic Press (www.IAPress.org)

FogTrust: A Lightweight Mutual Authentication and Key Agreement Protocol for Fog–IoT-Based E-Healthcare Environments

Meriam Fariss [1,*], Ahmed Toumanari [2]

[1] *LabSIV, Computer Science Department, Faculty of Science Agadir, Ibn Zohr University, Morocco*
[2] *MAISI, ENSA Agadir, Ibn Zohr University, Morocco*

Abstract Internet of Things (IoT) is nowadays an important component of the E-healthcare systems, enabling real-time patient status tracking in addition to highly effective medical services. Nonetheless, this integration of IoT technologies represent important security and performance challenges. We develop FogTrust, an optimized authentication and cryptographic key agreement protocol for fog–based E-Healthcare environments, founded on Elliptic Curve algorithms, biometric validation and Physical Unclonable Functions (PUFs). Our scheme ensures users, fog nodes and IoT devices anonymity and untraceability while withstanding several security attacks. Additionally, FogTrust provides a fine-grained access control phase where only authorized users are allowed to access confidential medical data. Additionally, we employ EdDSA to enhance signature efficiency and performance compared to traditional ECC-based signature schemes. FogTrust's formal security analysis is realized through the AVISPA Tool and the Scyther Tool that both demonstrate its robustness against various attacks. Along with its strong security features, FogTrust achieves high performance than alternative solutions, making it applicable in IoT devices known for there limited resources feature.

Keywords Mutual Authentication, Access Control, ECC, Key Agreement, PUF, IoT, Fog Computing, Biometrics

DOI: 10.19139/soic-2310-5070-4129

1. Introduction

The wide deployment of Internet of Things (IoT) in healthcare is a great revolution in medical services by ensuring real-time monitoring and personal care. Modern IoT healthcare systems are structured into three interconnected layers [1]. Patient-facing devices continuously collect health data directly from patients [2]. Subsequently, this data is processed locally by an intermediate layer employing fog computing to ensure fast, secure and immediate response. Finally, a cloud-based layer stores and validates the data. Despite its advantages, this multi-layer IoT healthcare architecture is vulnerable to various challenges related to security concerns due to the highly sensitive medical records, the diversity of IoT devices and the distributed communication network [3, 4]. All these challenges increase exposure to significant security weaknesses such as unauthorized access, data tampering and anonymity and traceability issues. Developing robust security mechanisms are therefore essential to establish trust among patients, physicians, IoT devices and gateways [5, 6]. Particularly, mutual authentication protocol, including multi-factor authentication, have been widely analyzed to avoid impersonation attacks and achieve high security levels. Moreover, fine-grained access control schemes are also developed to provide strict authorization policies, therefore, only legitimate users can access to remote health data [7, 8, 9]. As a result, it is mandatory to realize a compromise between security strength and efficiency in IoT-based healthcare systems given the limited capabilities of wearable and sensor devices [10]. Traditional cryptographic techniques often require significant

*Correspondence to: Meriam Fariss (Email: m.fariss@uiz.ac.ma). Department of Computer Science, Faculty of Science Agadir, Ibn Zohr University, Morocco.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

computational, memory and energy overhead, making them infeasible in large-scale medical IoT deployments [11, 12]. As a result, Elliptic Curve Cryptography (ECC) serves as a resource-optimized mechanism that maintains high security robustness with smaller key parameters and lower complexity [13, 14].

Motivation: Although these advances exist for a higher security, designing an authentication protocol for IoT-healthcare remains very challenging due to strong security requirements that must be satisfied without demanding excessive computational and communication burden on resource-limited devices used in medical field. Particularly, in healthcare systems, secure user authentication is no more enough, but also a high protection of user private credentials, mutual authentication between different communicating parties, secure session-key computation and exchange, preserving patient and user privacy, and fine-grained authorization to highly sensitive medical records. Meanwhile, executing cryptographic operations that are computationally intensive directly on IoT devices can considerably affect their efficiency. Hence, integrating fog computing technology as an intermediate security layer makes the security mechanism able to support authentication and access-control operations near to medical devices while minimizing their computational cost. Taking all these challenges into consideration, we propose FogTrust, a lightweight fog-assisted authentication and key agreement protocol that combines cryptographic and hardware-assisted techniques to achieve secure and efficient access to healthcare IoT networks.

Contributions: FogTrust protocol ensures many major contributions that we can summarize in five points:

- **Three-factor authentication:** FogTrust proposes a secure mechanism to securely store and manage private credentials while combining three factors: password, biometrics and device specific secrets.
- **Use of lightweight cryptographic primitives:** FogTrust combines ECC, EdDSA, PUFs, fuzzy extractors and MAC through different protocol phases (setup, registration, login, mutual authentication and access control), ensuring both secure authentication, data integrity and confidentiality, along with high efficiency. All this makes FogTrust feasible in resource-constrained IoT E-healthcare systems.
- **Fog Computing-based security architecture:** FogTrust architecture leverages fog nodes that have a significant impact on improving the mutual authentication and key exchange procedure. This fog layer represents an intermediate trusted entity between users and IoT devices and significantly reduces communication and computation costs.
- **Security verification:** The robustness of FogTrust protocol against several security threats is proved through a detailed informal analysis. Moreover, we complement it with automated security validation with both AVISPA and Scyther security analysis tools.
- **Performance analysis:** The efficiency of FogTrust is evaluated through a comprehensive analytical assessment of its computational and communication costs and is benchmarked against related authentication protocols.

Organization: We structure our paper as outlined below: second Section presents a discussion of existing literature related to our work. Third Section presents an in-depth overview of the system model and provides more details about the different phases of FogTrust. In Fourth section, the security of FogTrust is analyzed combining informal and formal verification by the AVISPA Tool and the Scyther Tool. The performance of the protocol is evaluated in Fifth Section, taking into consideration computation and communication efficiency, and benchmarking it against similar protocols proposed previously. The final Section draws the conclusion of our work, where we briefly summarize our results and present the future work based on our findings.

2. Related Work

There is a wide range of mutual authentication mechanisms to preserve a high level of integrity and confidentiality while exchanging sensitive medical data. To address security flaws in the heterogeneous E-healthcare IoT environments, [15] proposed a security protocol using symmetric encryption and hash functions. Although effective for small-scale networks, the scheme is inefficient when additional sensor nodes are involved in the network. Authors in [16] reviewed the protocol in [17] and shows that user anonymity and untraceability are not guaranteed.

To correct the security weakness, they proposed an improved scheme that ensures secure session key and mutual authentication. [18] introduced a lightweight cryptographic key exchange protocol for the Internet of Medical Things, using PUFs to facilitate network devices in authenticating doctors and sensors before establishing shared session key. Authors in [19] reviewed the [20]'s lightweight authentication scheme for wearable devices and found that it is exposed to insider, device compromise and desynchronization attacks. To correct these flaws, authors in [19] designed the SHAPARAK protocol offering scalable and anonymous authentication with secure device registration and password/biometric updates. However, their solution is susceptible to men-in-the-middle and impersonation attacks [21]. Authors in [22] developed the ESEAP scheme, that can withstand offline password guessing attacks. [23] introduced a formally proven scheme enabling secure fine-grained access control mechanism for healthcare applications in mobile clouds. However, it is vulnerable to inadequate repairability and delayed typo detection. Authors of [24] proposed an ECC-based biometric authentication and key agreement scheme for multi-server TMIS architecture. However, the protocol suffers from practical limitations related to the management and distribution of server public keys [25]. Authors in [25] introduced a protocol for authentication in IoT healthcare environments based on 5G networks preserving user anonymity.

Blockchain technology is also integrated in recent research studies especially for healthcare systems. Authors in [26] integrated blockchain, fog computing and IoT to securely manage electronic health records via decentralized storage and access control. Wang et al. [27] based their study on blockchain technology and developed a zero-trust authentication protocol aimed for Medical IoT (MIoT) environments. Their approach eliminates single points of failure and allows node revocation.

In addition to blockchain technology, recent research efforts have also focused on post-quantum cryptographic protocols and their use especially in healthcare environments. Authors in [28] introduced a quantum-resistant protocol to achieve mutual authentication in the Internet of Health Things (IoHT). Their work is based on blockchain architecture in order to provide decentralized authentication while being resistant against quantum attacks. Another research direction has also explored advanced PUF-based authentication schemes. For instance, BLAKE is a protocol proposed by authors in [29] that integrates PUFs with fog devices to address resistance against impersonation, man-in-the-middle, and session-key disclosure attacks with minimal computation costs. In 2025, Tawfik et al. [30] designed a protocol that integrates blockchain technology with zero-knowledge proofs (ZKPs) to preserve privacy while resisting to quantum threats. In 2026, a lightweight elliptic curve signature protocol was proposed by El Ouafi et al. [31]. Their proposed scheme is not only resistant to key-recovery attacks, but also achieves low computation costs which makes it suitable for resource-constrained IoT devices.

3. Proposed Protocol

FogTrust system consists of four main entities that participate in the communication process as illustrated in Figure 1. Users ($User_i$) include medical personnel such as doctors, nurses and other authorized staff who need to access, manage and/or share patient medical data and Personal Health Records (PHRs). Their access rights are determined by assigned roles and managed by the gateway node (GWN), that plays the role of a trusted central authority, equivalent to the Hospital Registration Center. It registers users, fog nodes and IoT devices, manages their permissions and enforces access control policies through an Access Control List(ACL)-based mechanism [32]. Fog Nodes (FN_j) are trusted entities between resource-constrained IoT devices and the GWN. Each fog node acts as a cluster head and manages a cluster of IoT devices while performing local data processing, offloading computation and communication tasks. The main purpose of clustering in fog-IoT environments is to organize smart devices into easily and effectively manageable groups. The assignment of every device to the nearest cluster head is based on multiple criteria such as proximity, communication latency and signal strength. Finally, IoT devices (dev_j) including wearable sensors, actuators, implantable devices,... which collect physiological data and securely transmit it to the corresponding fog node. Table 1 lists the main notations and functions involved in FogTrust.

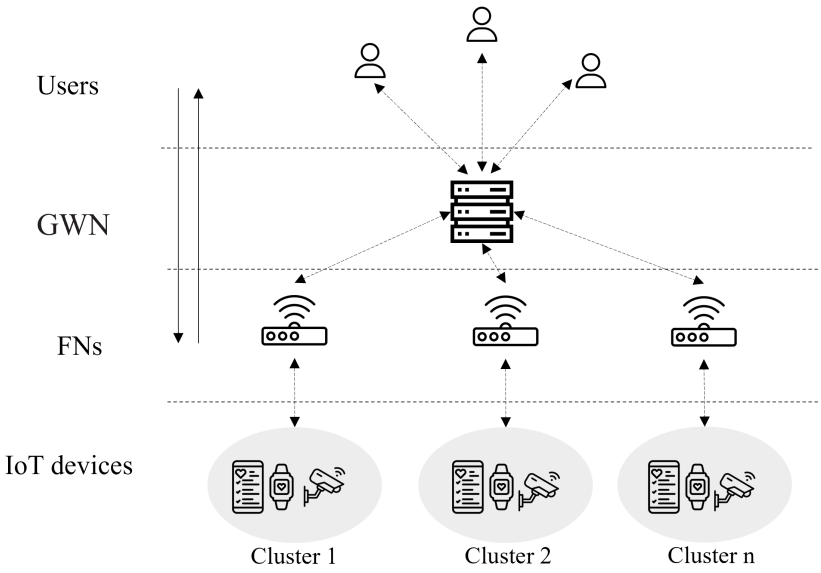

Figure 1. Proposed System Model

Table 1. Notation and description of parameters

Parameter	Description
ID_i	user$_i$'s identity
SID_j	IoT device$_j$'s identity
FID_j	Fog node$_j$'s identity
$PUF(\cdot)$	Physical Unclonable Function
$MAC_k(\cdot)$	Message authentication code using key k
(t_i, T_i)	user$_i$'s private/public key pair
(x_{GWN}, X_{GWN})	GWN's private/public key pair
$E_{AES}(M)_k / D_{AES}(M)_k$	AES encryption/decryption of message M using symmetric key k
$Sign_s(M)$	EdDSA signature generation using secret key s
$Verify_K(M)$	EdDSA signature verification using public key K
TS	Timestamp
$status_i$	Status of $user_i$ (active or revoked)
$status_{dev_j}$	Status of IoT device dev_j (active or revoked)

3.1. Threat Model

FogTrust follows the Dolev-Yao threat model [33], where all communications are performed over the unsecure public channel under the control of the attacker. This latter is able to eavesdrop, intercept, modify, replay, inject, delay, or block the exchanged messages among protocol entities. Nonetheless, the attacker cannot break the cryptographic primitives used in FogTrust protocol which are assumed to be computationally secure. The attacker is also assumed to capture user smart cards and IoT devices. Moreover, the Gateway Node represents a trusted third party to perform registration, authentication, access control, and revocation phases.

3.2. FogTrust Initialization Phase

Initially, the GWN performs system setup. GWN selects an EC $E(\mathbb{F}_p)$ over $\mathbb{F}_p$,a finite field, and defines a generator point P of large prime order n. After that, GWN produces its primary key x_{GWN} and derives $X_{\text{GWN}} = x_{\text{GWN}}P$ its

public key. Finally, the system public parameters $PP = \{E(\mathbb{F}_p), G, P, h(), X_{\text{GWN}}\}$ are published to enable secure operations within the network.

3.3. User Registration Phase

New users register with GWN, which verifies their identity (ID_i) and assigns access rights to specific IoT devices ($listIoTdev_i$) and permissions ($permission\_list_i$) via an ACL (Figure 2).

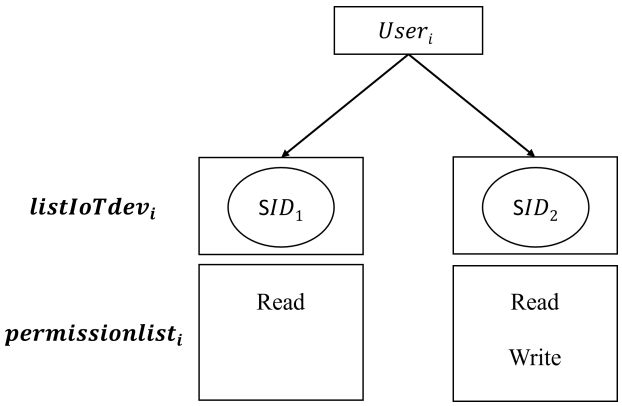

Figure 2. Example of a user's Access Control List

Figure 3 describes the execution of this phase.

To avoid the permanent storage of the challenge $B_i = h(HPW_i \| c_i)$ in the smart card memory, B_i is deterministically computed at every login phase based on the user password input and the biometric derived value. B_i represents the challenge to the embedded PUF. In FogTrust, we assume that the registration phase is performed over a secure channel.

$User_i(t_i, T_i)$	$GWN(x_{GWN}, X_{GWN})$
Chooses ID_i and PW_i generates u_i randomly calculates $HPW_i = h(PW_i \| u_i)$ imprints b_i	
$\{ID_i, b_i, HPW_i\}$ → *secure channel*	checks the length of ID_i checks if ID_i exists in its database, chooses a codeword $c_i \in C$ calculates $F(c_i, b_i) = (\alpha, \delta)$ where $\alpha = h(c_i)$ and $\delta = c_i \oplus b_i$, calculates $A_i = h(ID_i \| HPW_i \| c_i)$, $B_i = h(HPW_i \| c_i)$, calculates $t_i = PUF_i(B_i)$, $T_i = t_i P$ sets $status_i = active$ stores $\{ID_i, T_i, status_i, listIoTdev_i, permission\_list_i\}$ in its users database stores $\{\alpha, \delta, A_i, f()\}$ in smart card that is equipped with an embedded PUF_i module
Smart Card ← *Secure channel*	
stores u_i in smart card	

Figure 3. user registration phase

3.4. Fog Node Registration Phase

Within this process, GWN and FN_j is establish a shared secret key K_{GWN-Fj}. the GWN also computes the fog node public key F_j. Details of this phase are shown in Figure 4

$GWN(x_{GWN}, X_{GWN})$	$FN_j(f_J, F_j)$
selects a unique FID_j generates f_j randomly stores f_j in FN_j's memory calculates the secret shared key with FN_j: $K_{GWN-Fj} = PUF_j(f_j)$ and its corresponding public key $F_j = f_jP$ publishes the public key F_j stores $\{FID_j, K_{GWN-Fj}\}$ in its FNs database.	Each FN_j has its own pre-implemented $PUF_j()$

Figure 4. Fog node registration process

3.5. IoT device Registration Phase

During this phase, each dev_j is securely enrolled in the network by the GWN. this latter assigns a unique device identifier SID_j and generates and stores a private key d_j in the device memory. Then,the public key is derived $D_j = d_jP$. Finally, GWN stores parameters $\{SID_j, D_j, FID_j, status_{dev_j} = active\}$ in its devices database.

3.6. FogTrust Login and Mutual Authentication Phase

There are four steps in this phase detailed in Figure 5.

1. **Step 1:** This step involves several sub-steps performed by the user:
   - The user inputs a fresh biometric data b_i'. Then, c_i' is computed as $c_i' = f(\delta \oplus b_i')$ using the stored helper data δ. .Then, the user checks that $h(c_i') = \alpha$. If true, this means that the biometric variation is within the acceptable error-correction range. If the comparison fails, the login request is locally rejected and the process ends. The user can retry a new login process.
   - If the comparison holds, the user inserts ID_i and PW_i, and computes $A_i' = h(ID_i \,\|\, h(PW_i \,\|\, u_i) \,\|\, c_i)$ and checks if $A_i' = A_i$.
   - If the comparison holds, then the user login is completed successfully. Consequently, the user calculates $t_i = PUF_i(h(h(PW_i \,\|\, u_i) \,\|\, c_i))$.
   - e_i is randomly selects and the one-time public key $E_i = e_iP$ is computed accordingly, then the user generates the symmetric key $key_1 = t_iX_{GWN}$.
   - The user encrypts the message composed of its identity ID_i, the targeted IoT device public key D_j, and the one-time public key E_i using the symmetric key key_1 to obtain the ciphertext $Enc_1 = E_{AES}(ID_i \,\|\, D_j \,\|\, E_i)_{key_1}$.
   - The user computes the EdDSA signature $Sign_1 = Sign_{t_i}(Enc_1 \,\|\, TS_1)$ using its private key t_i.
   - Finally, the user sends $\{Enc_1, TS_1, Sign_1\}$ to the GWN.
2. **Step 2:** The GWN performs the following checks and calculations:
   - Checks TS_1'. If the difference is $\leq \Delta T$, the verification proceeds. In FogTrust, it is assumed that all involved entities maintain clock synchronization within an acceptable threshold ΔT as widely required by authentication mechanisms based on timestamps to prevent replay attacks.
   - Uses $Verify_{T_i}(Enc_1 \,\|\, TS_1, Sign_1)$.
   - If successful, the GWN calculates the symmetric key $key_1' = x_{GWN}T_i$ to decrypt the received message.

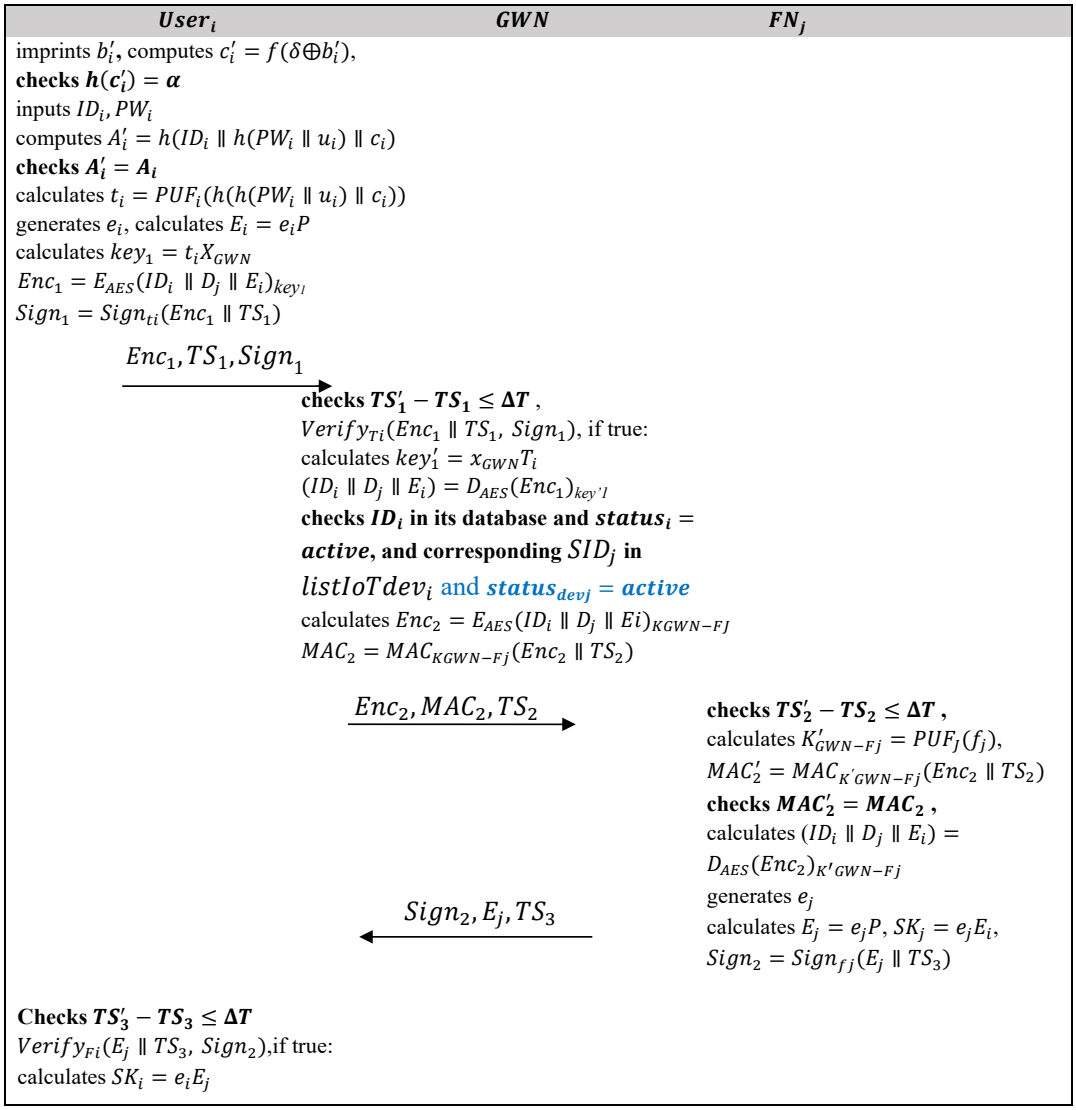

Figure 5. FogTrust Mutual authentication and key agreement

- Performs further checks to ensure that $status_i = active$ and the targeted device SID_j exists in $listIoTdev_i$ and $status_{dev_j} = active$.
- If verification passes, the GWN uses the symmetric key K_{GWN-Fj} to encrypt $Enc_2 = E_{AES}(ID_i \| D_j \| E_i)_{K_{GWN-Fj}}$ and authenticates it using the MAC function: $MAC_2 = MAC_{K_{GWN-Fj}}(Enc_2 \| TS_2)$, then sends Enc_2, MAC_2 and TS_2 to the fog node responsible for the targeted IoT device.

3. **Step 3:** The fog node performs the following:

- Checks whether $TS_2' - TS_2 \leq \Delta T$.
- Calculates the key $K'_{GWN-Fj} = PUF_j(f_j)$ using its private key f_j.
- Computes $MAC_2' = MAC_{K'_{GWN-Fj}}(Enc_2 \| TS_2)$ and checks if $MAC_2' = MAC_2$.
- If verification succeeds, decrypts Enc_2 using K'_{GWN-Fj}.

- Generates a random value e_j and computes the one-time public key $E_j = e_j P$.
- Calculates the shared secret key $SK_j = e_j E_i$.
- Generates an EdDSA signature $Sign_2 = Sign_{f_j}(E_j \parallel TS_3)$.
- Sends $\{Sign_2, E_j, TS_3\}$ to U_i.

4. **Step 4:** Once the user receives the message:

   - Checks TS_3.
   - Verifies E_j signature using F_j:$Verify_{F_j}(E_j)$ to ensure it is authentic.
   - Calculates the shared secret key $SK_i = e_i E_j$, which will be used for secure communication with the fog node during the session.

In FogTrust, the authentication process is a sequence of consecutive steps and progression depends on successful verifications of previous steps. In case of lost, invalid or delayed messages, the current session ends and temporary parameters are discarded. Since long-term credentials are not modified during the session, a new authentication process can be restarted without causing any desynchronization issues between communicating entities.

3.7. Access Control Phase

The access control phase involves the user requesting access to medical data sensed by the IoT device and the GWN that controls resource access. The access control phase is summarized in Figure 6. In our proposed protocol, we use Access Control List (ACL) to achieve a fine-grained authorization process, where the GWN performs the verification of the user and IoT device status. Moreover, it checks whether the requested operation over the targeted device exists in the user's permission list before granting access to the requested data. Instead of using role or attributed based access control mechanisms, FogTrust relies on ACL approach due to its suitability to resource-constrained IoT environments.

The access control phase is performed as follows:

Step 1: The user contacts FN_j, responsible for the targeted IoT device cluster, by sending a request to access the sensed data. Before sending the request, the user calculates $\text{Enc}_1 = E_{AES}\big(ID_i \parallel D_j \parallel \text{request}\big)_{SK}$. Then, the user calculates $\text{MAC}_1 = \text{MAC}_{SK}\big(\text{Enc}_i \parallel TS_1\big)$. This step ensures the integrity and authenticity of the message, preventing any unauthorized access or modification of the message during transmission. Finally, the user sends $\{\text{Enc}_1, \text{MAC}_1, TS_1\}$ to the fog node FN_j.

Step 2: The fog node checks if $(TS_1' - TS_1) \leq \Delta T$ and then calculates a new message authentication code $\text{MAC}_1' = \text{MAC}_{SK}(\text{Enc}_1 \parallel TS_1)$ using the shared session secret key SK and checks if MAC_1' matches the MAC_1 received with the access request. A matching pair of MACs confirms the integrity of the access request. Therefore, the fog node decrypts the access request using the same secret key SK to recover the identity of the requesting user and the targeted IoT device through its public key D_j, along with the *request*. The fog node then calculates $\text{Enc}_2 = E_{AES}(ID_i \parallel D_j \parallel \text{request})_{K_{GWN-Fj}}$ with the same identity and data source along with the request, but this time, it is encrypted using a different secret key K_{GWN-Fj}. Finally, the fog node calculates a new message authentication code $\text{MAC}_2 = \text{MAC}_{K_{GWN-Fj}}(\text{Enc}_2 \parallel TS_2)$ and sends $\{\text{Enc}_2, \text{MAC}_2, TS_2\}$ to the GWN.

Step 3: The GWN checks the timestamp to ensure that it falls within an acceptable range. The GWN calculates $\text{MAC}_2' = \text{MAC}_{K_{GWN-Fj}}(\text{Enc}_2 \parallel TS_2)$ using the symmetric key K_{GWN-Fj}. The GWN compares the calculated MAC_2' with MAC_2 received from the fog node to ensure message integrity. The GWN decrypts the message Enc_2 by computing $(ID_i \parallel D_j \parallel \text{"request"}) = D_{AES}(\text{Enc}_2)_{K_{GWN-Fj}}$ to obtain ID_i, D_j and the request. The GWN inspects its database to verify if ID_i is listed with an active $status_i$, and if the targeted IoT device with identity SID_j is in $listIoTdev_i$ inline with an active status and the request is in $permissionlist_i$. If the check is successful, the GWN sets the variable $access = 1$. This operation confirms that the user is permitted to obtain the requested data. If not, it sets $access = 0$. Then, the GWN calculates $\text{MAC}_a = \text{MAC}_{K_{GWN-Fj}}(access \parallel TS_3)$. Finally, the GWN sends $\{access, \text{MAC}_a, TS_3\}$ to the fog node.

During every new access request, the GWN verifies the user's current authorization data by checking $status_i$, $status_{dev_j}$, $listIoTdev_i$ and $permissionlist_i$. In case of user's privileges revocation, this update is applied from the next access request process.

Step 4: In this step, the fog node performs the following operations: If $access = 0$, access is denied, and the process ends. If $access = 1$, the fog node checks the timestamp difference $(TS_3' - TS_3) \leq \Delta T$ against the time threshold ΔT to ensure that the request is still valid. Then, the fog node calculates $\text{MAC}_a' = \text{MAC}_{K_{GWN-Fj}}(access \parallel TS_3)$ and compares it with the received MAC_a. The fog node then calculates $key_2 = f_j D_j$. Next, the fog node encrypts the message *"request"* using key_2 to get $\text{Enc}_3 = E_{AES}(\text{request})_{key_2}$. Finally, the fog node calculates $\text{MAC}_3 = \text{MAC}_{key_2}(\text{Enc}_3 \parallel TS_4)$, and then sends $\{\text{Enc}_3, \text{MAC}_3, TS_4\}$ to the targeted IoT device.

Step 5: After successful timestamp check,IoT device calculates the symmetric key $key_2' = d_j F_j$ and uses it to compute $\text{MAC}_3' = \text{MAC}_{key_2'}(\text{Enc}_3 \parallel TS_4)$. Then, the IoT device checks whether the calculated MAC_3' matches the received MAC_3 and decrypts the received encrypted *request* by calculating $\text{request} = D_{AES}(\text{Enc}_3)_{key_2'}$. It encrypts its corresponding response message by computing $\text{Enc}_4 = E_{AES}(\text{response})_{key_2'}$ and calculates $\text{MAC}_4 = \text{MAC}_{key_2'}(\text{Enc}_4 \parallel TS_5)$ and sends $\{\text{Enc}_4, \text{MAC}_4, TS_5\}$ to the fog node.

Step 6: The fog node first checks if $(TS_5' - TS_5) \leq \Delta T$. Then, FN_j calculates $\text{MAC}_4' = \text{MAC}_{key_2}(\text{Enc}_4 \parallel TS_5)$ and checks if $\text{MAC}_4' = \text{MAC}_4$. It then calculates $\text{response} = D_{AES}(\text{Enc}_4)_{key_2}$, which decrypts the encrypted response message using key_2 to obtain the response. Next, FN_j calculates $\text{Enc}_5 = E_{AES}(\text{response})_{SK}$, encrypting the response message using SK, and then computes $\text{MAC}_5 = \text{MAC}_{SK}(\text{Enc}_5 \parallel TS_6)$. Finally, it sends $\{\text{MAC}_5, \text{Enc}_5, TS_6\}$ to the user.

Step 7: the user first checks if the timestamp, then calculates $\text{MAC}_5' = \text{MAC}_{SK}(\text{Enc}_5 \parallel TS_6)$ and compares this MAC_5' with the MAC_5 received from the fog node to ensure that the response has not been tampered with. If the comparison is successful, the user decrypts the response by computing $\text{response} = D_{AES}(\text{Enc}_5)_{SK}$ to obtain the requested data.

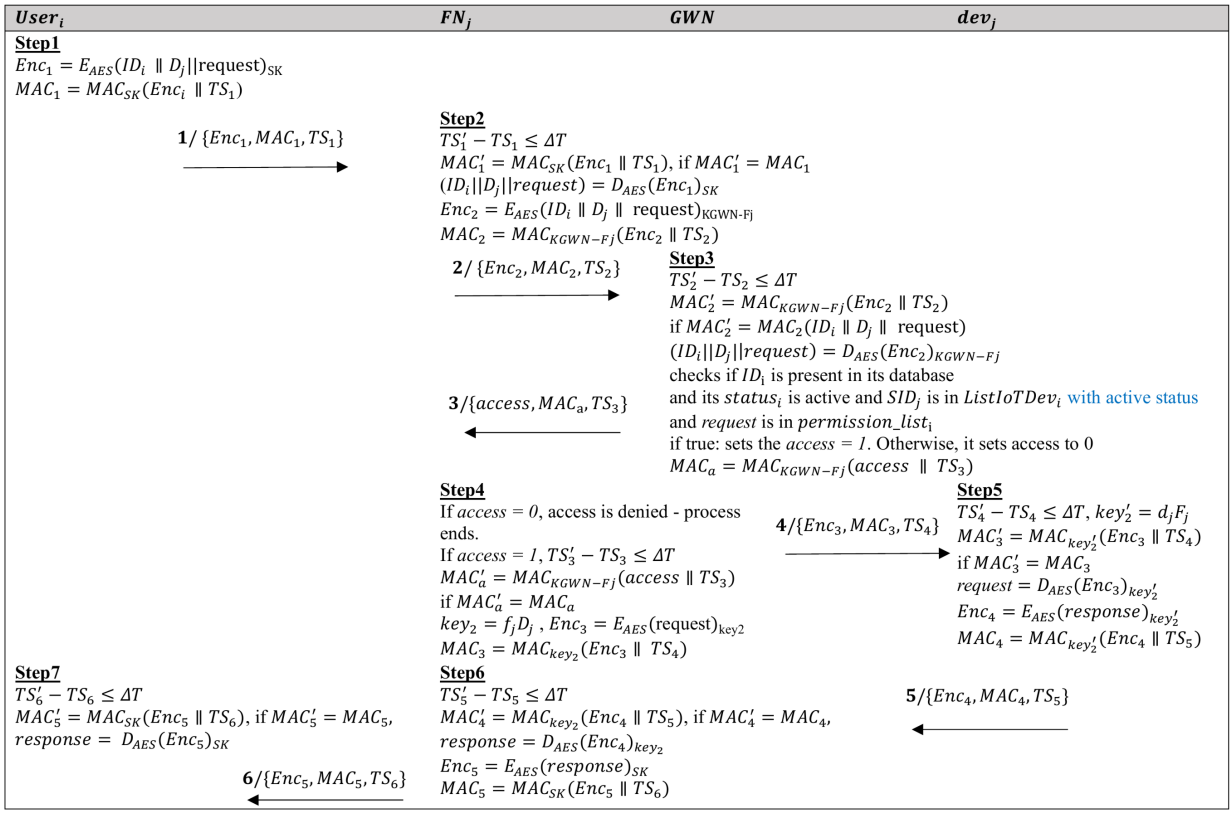

Figure 6. Access control phase

3.8. User and IoT Device Revocation

In case of user or device compromise, an authenticated administrator can perform a revocation request through a trusted channel. Thus, the GWN updates the status from $active$ to $revoked$. During subsequent mutual authentication and access control phases, the GWN grants access only if $status_i$ and $status_{dev_j}$ are both active. Hence, revoked users and devices can no longer participate in the FogTrust communication process.

4. Security Analysis of FogTrust

4.1. Informal Security Analysis

We informally discuss in the following subsection how FogTrust protocol can withstand several security flaws.

Resistant to replay attacks Let's analyze the exchanged messages :
MSG1:$\{Enc_1, TS_1, Sign_1\}$ from the user to the GWN
MSG2:$\{Enc_2, TS_2, MAC_2\}$ from the GWN to the FN
MSG3:$\{E_j, TS_3, Sign_2\}$ from the FN to the user
All these messages include fresh timestamps (TS_1, TS_2, TS_3) and session specific keys (E_i, E_j) which are randomly generated for each session. The timestamp verification performed by each entity ensures that only recently generated messages within the window ΔT are accepted. Moreover E_i and E_j bind each session to a unique session key which prevents any adversary from reusing old messages in upcoming sessions.

Resistant to user, fog node and IoT device impersonation attack For successful authentication, a user must provide three valid factors:smart card, password and biometric input. The private key is computed locally t_i. Without access to all these three factors, an adversary cannot compute the correct t_i. Therefore, the signature $Sign_1$ for the message $\{Enc_1, TS_1, Sign_1\}$ is invalid. The GWN verifies $Sign_1$ using the user's public key T_i, ensuring that only legitimate user can initiate the authentication.
Each fog node FN_j has a unique private key f_j generated from the GWN and the shared secret key $K_{GWN} = PUF_j(f_j)$ between FN_j and GWN, obtained from the device's unique PUF. Moreover, each fog node signs its ephemeral public key E_j using its private key f_j, which is verified by the remote user using the fog node's public key F_j.
An adversary cannot impersonate an IoT device dev_j since only dev_j knows its private key d_j. Without d_j an attacker cannot generate valid signatures or correctly encrypt/decrypt exchanged data.

No password verifier-table In the proposed scheme, GWN never stores user passwords in plaintext or hashed form. During user registration phase the password PW_i is combined with a random value u_i and hashed $HPW_i = H(PWi||u_i)$. PW_i is never transmitted to the GWN.

No smart card loss attack In the proposed protocol, to generate a valid authentication request, an adversary must compute the private key t_i, which is derived using the device-specific PUF function as $t_i = PUF_i(B_i)$. However, the value B_i is never stored in the smart card, instead $B_i = h(h(PW_i||u_i)||c_i)$. Since neither PW_i, nonce u_i nor c_i are stored in the smart card, reconstructing B_i is computationally infeasible.

Session Key agreement U_i and FN_j establish a secure session key using ECC. The user randomly chooses e_i and computes the relevant public key E_i. Similarly, FN_j generates its own random ephemeral private key e_j and computes $E_j = e_j P$. Both entities independently compute the shared session key $SK_i = e_i E_j$ (user side) and $SK_j = e_j E_i$ (fog node side) where $SK = SK_i = SK_j$. Let's consider an adversary who tries to intercept all exchanged messages including E_i and E_j. To derive the session key, the adversary must compute either e_i from $E_i = e_i P$ or $E_j = e_j P$. This operation requires solving the Elliptic Curve Discrete Logarithm Problem.

Early detection of incorrect login credentials Prior to initiating the authentication process, the smart card performs a local verification of login credentials input by the user by checking $h(c_i') = \alpha$ and $A_i' = A_i$. The authentication occurs only if both verification are correct and immediately ends if not. Hence, FogTrust can detect incorrect credentials before engaging extra communication and computation costs.

Anonymity and untraceability of users, fog nodes and IoT devices Our goal is to prevent an attacker from identifying or tracking system entities through different sessions. FogTrust achieves this goal via:
One-time public keys: users and fog nodes randomly generate temporary public keys E_i and E_j respectively for every session. Even if an adversary gains access to these keys, it's impossible to for them to link them to previous or future sessions.
Encrypted identities: Users, IoT devices and fog nodes identities are not explicitly communicated across the public channel. Instead, they are always encrypted. This approach prevents eavesdroppers from learning what entities are involved in the communication process.

Resilient to Cloning Attack Our protocol resists cloning attacks because both fog nodes and user's smart cards rely on PUFs to generate unique responses to specific challenges. These PUF responses are inherently tied to the physical characteristics of each device and cannot be duplicated, even if an adversary obtains the device.

Resistance to Physical Capture and Side-Channel Attacks FogTrust is designed to avoid physical capture and side-channel attacks against smart card and fog nodes. In fact, the only stored data in the smart card are $\{\alpha, \delta, A_i, f(), u_i\}$ in addition to the embedded PUF. Whereas sensitive data such as PW_i, b_i, t_i and B_i are not stored. Moreover, by design, the fuzzy extractor generates the helper data δ without revealing the secret value of the PUF response c_i. Additionally, FogTrust dynamically regenerates the private key $t_i = PUF_i(B_i)$ which prevents an attacker from retrieving t_i even if they possess the physical smart card. Similarly, at the fog node side, the shared secret key $K_{GWN-F_j} = PUF_j(f_j)$ is generated based on the the fog node embedded PUF.

Resistance to Key Compromise Impersonation: In FogTrust, if a permanent private key is compromised, an attacker is still unable to impersonate another legitimate entity. For example, if the user's private key t_i is known, the attacker cannot impersonate the fog node to the user without f_j which is necessary to generate $Sign_2 = Sign_{f_j}(E_j \,\|\, TS_3)$. In the same way, knowledge of f_j does not guarantee user impersonation because the GWN verifies $Sign_1 = Sign_{t_i}(Enc_1 \,\|\, TS_1)$ through $Verify_{T_i}(Enc_1 \,\|\, TS_1, Sign_1)$.

Resistance to Ephemeral Key Leakage: For every authentication session, FogTrust randomly generates new temporary secret values e_i and e_j at the user and the fog node sides respectively. These values are used in the computation of the one-time public keys $E_i = e_iP$ and $E_j = e_jP$ along with the shared session key $SK_i = SK_j = e_iE_j = e_je_iP$. In addition to the ephemeral characteristic of these values, they are also discarded at the end of every session, hence their compromise does not reveal neither the previous nor future authentication sessions.

Perfect Forward Secrecy: Perfect forward secrecy is provided by FogTrust since the session key is only generated from temporary secret values independently generated during each authentication session. The fog node calculates $SK_j = e_jE_i$, and the user computes $SK_i = e_iE_j$, resulting in the computation of the same shared session key $SK_i = SK_j = e_ie_jP$. The permanent private keys t_i, x_{GWN}, and f_j are not directly integrated in the generation of the session key which makes an adversary unable to recover previous session keys even if these values are compromised.

Security of User and Device Revocation FogTrust prevents revoked users and IoT devices from participating in the authentication process by checking their status $status_i$ and $status_{dev_j}$ before forwarding authentication requests. The revocation decision is exclusively managed by the GWN.

4.2. Formal Security Analysis

We carry out a formal security assessment of FogTrust using the AVISPA Tool [34]. For our analysis, we utilized the OFMC and CL-AtSe backends and followed the syntax provided by the HLPSL supported by AVISPA. The formal security analysis results are shown in Figure 7 to demonstrate that FogTrust is secure based on the AVISPA tool formal security analysis.

% OFMC % Version of 2006/02/13 SUMMARY SAFE DETAILS BOUNDED_NUMBER_OF_SESSIONS PROTOCOL /home/span/span/testsuite/results/IoTMAKA2.if GOAL as_specified BACKEND OFMC COMMENTS STATISTICS parseTime: 0.00s seaGWNhTime: 7.38s visitedNodes: 680 nodes depth: 16 plies	SUMMARY SAFE DETAILS BOUNDED_NUMBER_OF_SESSIONS TYPED_MODEL PROTOCOL /home/span/span/testsuite/results/ IoTMAKA2.if GOAL As Specified BACKEND CL-AtSe STATISTICS Analysed : 138109 states Reachable : 95061 states Translation: 0.07 seconds Computation: 71.11 seconds

Figure 7. FogTrust AVISPA tool-based analysis results

We complete this AVISPA-based analysis with the use of Scyther tool for further verification to strengthen the security proof of FogTrust protocol. Scyther is a powerful tool used for automatic protocols security analysis based on Dolev-Yao adversary model. The main properties being verified are secrecy of nonces and shared session keys, authentication and role liveliness. We started our analysis by modeling FogTrust protocol in SPDL, where we accurately defined the roles of the three participating entities: user, GWN and Fj. The exchanged messages and the security claims are also modeled. The Scyther security verification results shown in Figure 8 prove that FogTrust achieves all security claims including secret, alive, weakagree, niagree and nisynch with no attacks detected for all claims.

5. Performance Analysis

5.1. Computation costs analysis

Based on the computation times in [35] and [36], the approximate computation time for each cryptographic operation used in our study are summarized in Table 2.

We provide in this section, the comparative assessment of computation costs results involved in the mutual authentication phase of various protocols proposed in the literature, including [37], [38] , [35] , [39], [40], [41], [42] and our proposed protocol FogTrust as shown in Table 3. This performance comparison focuses on the three different components, namely user-side, GWN side, and sensor node-side. The total computation time for every method is also provided in milliseconds (ms) in Table 4.

From the Table 4, we can observe that our protocol has the lowest total computation time, with a value of 4.287ms, compared to [35],[39],[40], [41] and [42]. Moreover, the mutual authentication and key agreement phase in our protocol does not involve the sensor nodes. Instead, this phase is offloaded to the fog node with sufficient computational resources, which results in a computation cost of $0ms$ at the sensor node side during this phase. When outsourcing the computation operations to a higher-capacity computing resource, the overall system

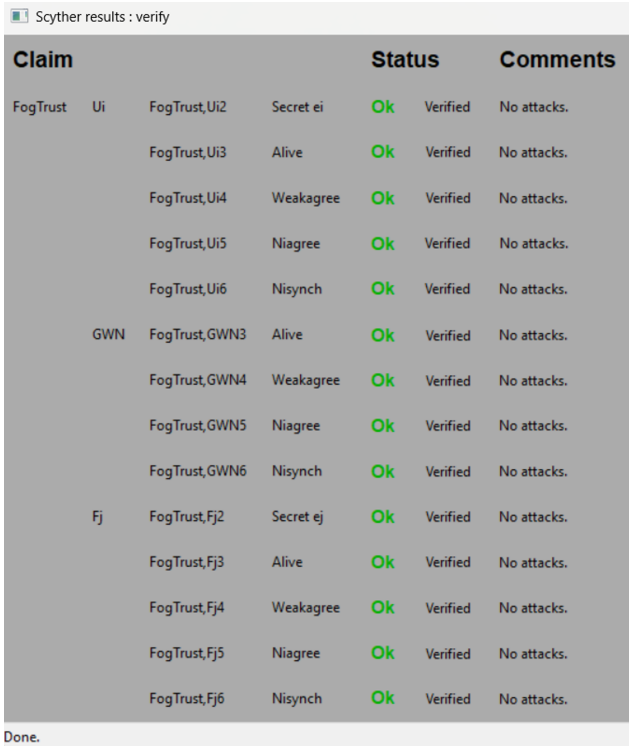

Figure 8. FogTrust formal security analysis results using Scyther tool

Table 2. Approximate computation time for each cryptographic operation

Notation	Description	Computation time (ms)
T_h	Hash function	0.069
T_{MAC}	Message Authentication Code	0.069
T_{AES}	AES-128 encryption/decryption	0.069
T_{ECM}	ECC point multiplication	0.508
T_{ECA}	ECC point addition	0.025
T_s	EdDSA signature generation	0.08
T_v	EdDSA signature verification	0.16

Table 3. Results of the analysis of the computation overhead

Protocol	User	GWN	Sensor Node	Total
[37]	$2T_{ECM} + 8T_h$	$T_{ECM} + 9T_h$	$4T_h$	$3T_{ECM} + 21T_h$
[38]	$3T_{ECM} + 7T_h$	$T_{ECM} + 4T_h$	$2T_{ECM} + 4T_h$	$6T_{ECM} + 15T_h$
[35]	$4T_{ECM} + 4T_h$	$4T_{ECM} + 5T_h$	$2T_{ECM} + 3T_h$	$10T_{ECM} + 12T_h$
[39]	$5T_h + 5T_{ECM}$	$4T_h + 5T_{ECM}$	$3T_h + 4T_{ECM}$	$12T_h + 14T_{ECM}$
[40]	$4T_h + 2T_{ECM}$	$2T_h + 2T_{ECA} + 5T_{ECM}$	$T_h + 2T_{ECA} + 3T_{ECM}$	$4T_{ECA} + 10T_{ECM} + 7T_h$
[41]	$10T_h + 2T_{ECM} + 3T_{AES} + 2T_{MAC}$	$11T_h + 2T_{ECM} + 4T_{AES} + 2T_{MAC}$	$T_h + T_{AES}$	$22T_h + 4T_{ECM} + 8T_{AES} + 4T_{MAC}$
[42]	$4T_{ECM} + 8T_h + T_{AES}$	$2T_{ECM} + 5T_h + T_{AES}$	$2T_{ECM} + 2T_h$	$8T_{ECM} + 15T_h + 2T_{AES}$
FogTrust	$3T_{ECM} + 4T_h + T_{AES} + T_s + T_v$	$T_{ECM} + 2T_{AES} + T_{MAC} + T_v$	$-^*$	$6T_{ECM} + 4T_h + 5T_{AES} + 2T_{MAC} + 2T_s + 2T_v$

* Fog node side ($T_{MAC} + T_{AES} + 2T_{ECM} + T_s$); the sensor node is not involved in the mutual authentication phase.

performance can be improved. Hence, our protocol can provide secure authentication without placing a significant computational burden on the resource-constrained IoT devices. By reducing the processing requirements on the IoT devices side, FogTrust protocol improves energy consumption and battery lifetime.

Table 4. Computation costs Analysis (ms)

Protocol	User side	GWN	Sensor Node side	Total
[37]	1.568	1.129	0.276	2.973
[38]	2.007	0.784	1.292	4.083
[35]	2.308	2.377	1.223	5.908
[39]	2.885	2.816	2.239	7.940
[40]	1.292	2.728	1.643	5.663
[41]	2.051	2.189	0.138	4.378
[42]	2.653	1.430	1.154	5.237
FogTrust	2.178	0.875	0^*	4.287

* Fog node side ($1.234ms$); the sensor node is not involved in the mutual authentication phase.

5.2. Communication costs comparison

The bit sizes involved in our study are listed in Table 5. Table 6 compares total number of bits transmitted for different security protocols and our proposed protocol.

Table 5. Protocol elements size (bits)

Protocol element	Size (bits)
Hash	160
Identities	160
MAC	128
Nonces	128
AES-encrypted data	128
Timestamp	64
EC Point	512
EdDSA Signature	512

Table 6. Communication costs Analysis (bits)

Protocol	Total of bits
[37]	2592
[38]	4512
[35]	4763
[39]	3552
[40]	2752
[41]	1536
[42]	2688
FogTrust	2144

For the compared protocols, the complete transmission size in bits ranges between 1563 bits and 4763 bits. FogTrust requires a total size of 2144 bits. The results presented in Table 6 reveal that the designed solution is the most communication-efficient compared to [37],[38],[35],[39],[40] and [42], as it transmits the smallest amount of data. When we analyze the comparison findings, we conclude that FogTrust maintains high performance costs outperforming existing solutions.

6. Conclusion

FogTrust is a resource-efficient authentication protocol enabling fine-grained access control for Fog-IoT-based E-healthcare systems. The protocol combines ECC, biometric authentication, PUFs and EdDSA signatures to provide robust security while ensuring high efficiency for IoT devices known for their limited resources. We employed both AVISPA and Scyther tools to demonstrate that FogTrust securely resists common security attacks. Performance evaluation shows that FogTrust is computationally and communication efficient, outperforming several existing schemes while enabling fog nodes to offload computationally intensive operations from resource-constrained IoT devices. As future work, we are working on an enhanced version of our protocol supporting dynamic device mobility in Fog-IoT environments where seamless re-authentication process is guaranteed when IoT devices move across different cluster heads. This enhancement will ensure high flexibility while being deployed in real-world E-healthcare networks. In our future research, we are exploring the integration of post-quantum cryptography in FogTrust protocol to address potential vulnerabilities of ECC and EdDSA to quantum attacks while preserving the protocol efficiency and performance. Another direction is to explore a distributed multi-gateway architecture to improve authentication management in large-scale deployments.

REFERENCES

1. P. Pal and S. Das, "Evaluation of security approaches towards vulnerable attacks in real-time monitoring and customised support for iot-healthcare system," *Procedia Computer Science*, vol. 258, pp. 669–678, 2025, doi :10.1016/j.procs.2025.04.300.
2. H. A. Khattak, M. Ruta, and E. E. Di Sciascio, "Coap-based healthcare sensor networks: A survey," in *Proceedings of 2014 11th International Bhurban Conference on Applied Sciences & Technology (IBCAST) Islamabad, Pakistan, 14th-18th January, 2014*. IEEE, 2014, pp. 499–503.
3. O. Popoola, M. Rodrigues, J. Marchang, A. Shenfield, A. Ikpehai, and J. Popoola, "A critical literature review of security and privacy in smart home healthcare schemes adopting iot blockchain: Problems, challenges and solutions," *Blockchain: Research and Applications*, vol. 5, no. 2, p. 100178, 2024, doi : 10.1016/j.bcra.2023.100178.
4. Z. Arabi, R. R. Oskouei, and M. Hosseinzadeh, "Enhancing security in iot networks: A multifaceted approach to vulnerability analysis and protection," *Array*, vol. 29, p. 100626, 2026, doi :10.1016/j.array.2025.100626.
5. R. Iqbal, M. Afzaal, and G. Rathee, "Hybrid and adaptive framework for secure and scalable authentication in healthcare iot," *Array*, vol. 28, p. 100527, 2025, doi : 10.1016/j.array.2025.100527.
6. R. Somasundaram and M. Thirugnanam, "Review of security challenges in healthcare internet of things," *Wireless Networks*, vol. 27, no. 8, pp. 5503–5509, 2021.
7. T. Xue, Y. Zhang, Y. Wang, W. Wang, S. Li, and H. Zhang, "Edge computing for iot: Novel insights from a comparative analysis of access control models," *Computer Networks*, vol. 270, p. 111468, 2025, doi : 10.1016/j.comnet.2025.111468.
8. S. Ravidas, A. Lekidis, F. Paci, and N. Zannone, "Access control in internet-of-things: A survey," *Journal of Network and Computer Applications*, vol. 144, pp. 79–101, 2019, doi :10.1016/j.jnca.2019.06.017.
9. A. Ouaddah, H. Mousannif, A. Abou Elkalam, and A. Ait Ouahman, "Access control in the internet of things: Big challenges and new opportunities," *Computer Networks*, vol. 112, pp. 237–262, 2017, doi :10.1016/j.comnet.2016.11.007.
10. V. A. Thakor, M. A. Razzaque, and M. R. Khandaker, "Lightweight cryptography algorithms for resource-constrained iot devices: A review, comparison and research opportunities," *IEEE access*, vol. 9, pp. 28 177–28 193, 2021.
11. S. Singh, P. K. Sharma, S. Y. Moon, and J. H. Park, "Advanced lightweight encryption algorithms for iot devices: survey, challenges and solutions," *Journal of Ambient Intelligence and Humanized Computing*, vol. 15, no. 2, pp. 1625–1642, 2024.
12. B. J. Mohd and T. Hayajneh, "Lightweight block ciphers for iot: Energy optimization and survivability techniques," *IEEE Access*, vol. 6, pp. 35 966–35 978, 2018.
13. S. S. Dhanda, B. Singh, and P. Jindal, "Lightweight cryptography: a solution to secure iot," *Wireless Personal Communications*, vol. 112, no. 3, pp. 1947–1980, 2020.
14. S. Ullah, J. Zheng, N. Din, M. T. Hussain, F. Ullah, and M. Yousaf, "Elliptic curve cryptography; applications, challenges, recent advances, and future trends: A comprehensive survey," *Computer Science Review*, vol. 47, p. 100530, 2023, doi : 10.1016/j.cosrev.2022.100530.
15. Z. A. Hussien, H. Jin, Z. A. Abduljabbar, M. A. Hussain, A. A. Yassin, S. H. Abbdal, M. A. Al Sibahee, and D. Zou, "Secure and efficient e-health scheme based on the internet of things," in *2016 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*. IEEE, 2016, pp. 1–6.
16. Q. Jiang, Z. Chen, B. Li, J. Shen, L. Yang, and J. Ma, "Security analysis and improvement of bio-hashing based three-factor authentication scheme for telecare medical information systems," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 4, pp. 1061–1073, 2018.
17. Y. Lu, L. Li, H. Peng, and Y. Yang, "An enhanced biometric-based authentication scheme for telecare medicine information systems using elliptic curve cryptosystem," *Journal of medical systems*, vol. 39, no. 3, p. 32, 2015.
18. M. Masud, G. S. Gaba, S. Alqahtani, G. Muhammad, B. B. Gupta, P. Kumar, and A. Ghoneim, "A lightweight and robust secure key establishment protocol for internet of medical things in covid-19 patients care," *IEEE Internet of Things Journal*, vol. 8, no. 21, pp. 15 694–15 703, 2020.

19. R. Hajian, S. ZakeriKia, S. H. Erfani, and M. Mirabi, "Shaparak: Scalable healthcare authentication protocol with attack-resilience and anonymous key-agreement," *Computer Networks*, vol. 183, p. 107567, 2020.
20. A. Gupta, M. Tripathi, T. J. Shaikh, and A. Sharma, "A lightweight anonymous user authentication and key establishment scheme for wearable devices," *Computer Networks*, vol. 149, pp. 29–42, 2019, doi :10.1016/j.comnet.2018.11.021.
21. O. Alruwaili, M. Tanveer, F. M. Alotaibi, W. Abdelfattah, A. Armghan, and F. M. Alserhani, "Securing the iot-enabled smart healthcare system: A puf-based resource-efficient authentication mechanism," *Heliyon*, vol. 10, no. 18, p. e37577, 2024, doi : 10.1016/j.heliyon.2024.e37577.
22. A. Kumari, S. Jangirala, M. Y. Abbasi, V. Kumar, and M. Alam, "Eseap: Ecc based secure and efficient mutual authentication protocol using smart card," *Journal of Information Security and Applications*, vol. 51, p. 102443, 2020, doi :10.1016/j.jisa.2019.102443.
23. S. Roy, A. K. Das, S. Chatterjee, N. Kumar, S. Chattopadhyay, and J. J. Rodrigues, "Provably secure fine-grained data access control over multiple cloud servers in mobile cloud computing based healthcare applications," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 457–468, 2018.
24. M. Qi, J. Chen, and Y. Chen, "A secure biometrics-based authentication key exchange protocol for multi-server tmis using ecc," *Computer Methods and Programs in Biomedicine*, vol. 164, pp. 101–109, 2018, doi : 10.1016/j.cmpb.2018.07.008.
25. Minahil, M. F. Ayub, K. Mahmood, S. Kumari, and A. K. Sangaiah, "Lightweight authentication protocol for e-health clouds in iot-based applications through 5g technology," *Digital Communications and Networks*, vol. 7, no. 2, pp. 235–244, 2021, doi : 10.1016/j.dcan.2020.06.003.
26. M. Nayak, M. S. Shirisha, S. Pattnayak, and S. Sharma, "Chapter 21 - securing healthcare records: Fortifying electronic health records through blockchain, fog computing, and internet of things (iot) integration," in *Fundamentals of Fog Computing and the Internet of Things for Smart Healthcare*, J. B. Awotunde, A. K. Bhoi, P. Barsocchi, and V. H. C. de Albuquerque, Eds. Elsevier, 2026, pp. 445–470.
27. Z. Wang, Y. Guo, and Y. Guo, "Blockchain-based zero-trust continuous authentication scheme for medical iot," *Journal of Parallel and Distributed Computing*, vol. 217, p. 105333, 2026.
28. A. Ahmad and J. Srirangan, "Quantum-safe mutual authentication scheme for ioht using blockchain," *Results in Engineering*, vol. 28, p. 106945, 2025.
29. C. Gupta and G. Varshney, "Blake: Ble-based lightweight authentication and key-exchange via puf," *Ad Hoc Networks*, p. 104370, 2026.
30. M. Tawfik, A. H. Abdelhaliem, and I. Fathi, "Quantum-resistant privacy-preserving iot authentication via zero-knowledge proofs and blockchain integration," *Statistics, Optimization & Information Computing*, vol. 14, no. 3, pp. 1374–1402, 2025.
31. K. L. Uahabi, A. Aslimani, A. Zannou *et al.*, "Optimized ambiguous-key ecc signatures for lightweight and secure iot systems," *Statistics, Optimization & Information Computing*, vol. 15, no. 4, pp. 2928–2947, 2026.
32. R. S. Sandhu and P. Samarati, "Access control: principle and practice," *IEEE Communications Magazine*, vol. 32, no. 9, pp. 40–48, Sep. 1994, doi :10.1109/35.312842.
33. D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.
34. L. Viganò, "Automated security protocol analysis with the avispa tool," *Electronic Notes in Theoretical Computer Science*, vol. 155, pp. 61–86, 2006, proceedings of the 21st Annual Conference on Mathematical Foundations of Programming Semantics (MFPS XXI); doi: 10.1016/j.entcs.2005.11.052.
35. K. Zhang, K. Xu, and F. Wei, "A provably secure anonymous authenticated key exchange protocol based on ecc for wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 2018, no. 1, p. 2484268, 2018, doi :10.1155/2018/2484268.
36. P. Gupta, A. Sinha, P. Kr. Srivastava, A. Perti, and A. K. Singh, "Security implementations in iot using digital signature," in *Innovations in Electrical and Electronic Engineering: Proceedings of ICEEE 2020*. Springer, 2021, pp. 523–535, doi : 10.1007/978-981-15-4692-$1_4$0.
37. X. Li, J. Niu, S. Kumari, F. Wu, A. K. Sangaiah, and K.-K. R. Choo, "A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments," *Journal of Network and Computer Applications*, vol. 103, pp. 194–204, 2018, doi :10.1016/j.jnca.2017.07.001.
38. Y. Choi, D. Lee, J. Kim, J. Jung, J. Nam, and D. Won, "Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography," *Sensors*, vol. 14, no. 6, pp. 10 081–10 106, 2014, dOI : 10.3390/s140610081.
39. S. Challa, M. Wazid, A. K. Das, N. Kumar, A. Reddy, E.-J. Yoon, and Y. Kee-Young, "Secure signature-based authenticated key establishment scheme for future iot applications," *IEEE Access*, vol. PP, pp. 1–1, 03 2017, doi : 10.1109/ACCESS.2017.2676119.
40. M. Saqib, B. Jasra, and A. H. Moon, "A lightweight three factor authentication framework for iot based critical applications," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 9, pp. 6925–6937, 2022, doi: 10.1016/j.jksuci.2021.07.023.
41. S. Naoui, M. H. Elhdhili, and L. A. Saidane, "Novel smart home authentication protocol lrp-shap," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, 2019, pp. 1–6, doi:10.1109/WCNC.2019.8885493.
42. M. Fariss, H. El Gafif, and A. Toumanari, "A lightweight ecc-based three-factor mutual authentication and key agreement protocol for wsns in iot," *International Journal of Advanced Computer Science and Applications*, vol. 13, pp. 491–501, 07 2022, doi : 10.14569/IJACSA.2022.0130660.