A user-centric privacy-preserving authentication protocol for IoT-AmI environments

Mehedi Masud a,1, Gurjot Singh Gaba b,1, Pardeep Kumar c, Andrei Gurtov b

a Department of Computer Science, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia
b Department of Computer and Information Science (IDA), Linköping University (LIU), Linköping 58183, Sweden
c Department of Computer Science, Swansea University, Swansea SA1 8EN, UK

ARTICLE INFO

Keywords:
Ambient intelligence
Blockchain
Fog computing
Healthcare
Internet of Things (IoT)

ABSTRACT

Ambient Intelligence (AmI) in Internet of Things (IoT) has empowered healthcare professionals to monitor, diagnose, and treat patients remotely. Besides, the AmI-IoT has improved patient engagement and gratification as doctors’ interactions have become more comfortable and efficient. However, the benefits of the AmI-IoT-based healthcare applications are not availed entirely due to the adversarial threats. IoT networks are prone to cyber attacks due to vulnerable wireless mediums and the absence of lightweight and robust security protocols. This paper introduces computationally-inexpensive privacy-assuring authentication protocol for AmI-IoT healthcare applications. The use of blockchain & fog computing in the protocol guarantees unforgeability, non-repudiation, transparency, low latency, and efficient bandwidth utilization. The protocol uses physically unclonable functions (PUF), biometrics, and Ethereum powered smart contracts to prevent replay, impersonation, and cloning attacks. Results prove the resource efficiency of the protocol as the smart contract incurs very minimal gas and transaction fees. The Scyther results validate the robustness of the proposed protocol against cyber-attacks. The protocol applies lightweight cryptography primitives (Hash, PUF) instead of conventional public-key cryptography and scalar multiplications. Consequently, the proposed protocol is better than centralized infrastructure-based authentication approaches.

1. Introduction

The Healthcare industry is being revolutionized with the tremendous progress in digital technologies, along with the IoT [1,2]. The healthcare industry is being transformed to the advanced level when people, apps, sensors, and medical devices communicate when delivering healthcare solutions [3,4]. IoT-driven drones, smart wearable devices, and health monitoring systems are considered to advance the healthcare industry’s development [5]. The requirement to acquire, preserve, and study patient information has advanced the healthcare industry to consider numerous trending digital technologies. The increasing acceptance of AmI-IoT in healthcare and health domains has involved the latest systems, i.e., the Internet of Medical Things (IoMT) [6]. It decreases human mistakes and eradicates many decision-making delays. Some of the significant benefits of accepting IoT in the ambient Healthcare environments are real-time monitoring, improved patient experience, and cost minimization. Leveraging IoT-based systems and medical devices permits the clinician to monitor the patient remotely with real-time data that expedites the diagnosis and treatment and provides advantages, such as persistent communication, travel expenses, and diminishes hospital resources.

In the healthcare industry, the AmI-IoT is considered for interrelated healthcare entities/devices like patients’ observing systems, sensor-based equipment, and sensor devices that acquire real-time health data. In healthcare 1.0, the doctors diagnosed the patients in person and maintained records on papers, whereas in healthcare 2.0, the doctors processed the patient’s information as electronic health records (EHR). Healthcare 3.0 transformed the paradigm of medical diagnosis by introducing internet-enabled wearable devices, telemedicine, etc. Healthcare 4.0 uses a range of technologies like the IoT, UAVs, augmented reality (AR), artificial intelligence (AI), deep learning (DL), and natural language processing (NLP) to optimize and automate medical procedures [7,8].

IoT is a world of connected objects [9]. It is predicted that by the year 2030, the total number of connected devices would be around 24 billion [10]. IoT nodes are tiny in size and have limited storage, less power, and limited computation capabilities. Fig. 1 illustrates the smart
healthcare system using IoT, where objects are controlled and monitored in real-time through the internet, e.g., automatic gate opening for ambulances, UAVs delivering human organs, and remote ordering of medicines and meals from pharmacy and pantry, respectively. Besides, the IoMT helps doctors and other authorized caretakers control the pressure of the ventilator, rate, and rhythm of the pacemaker remotely and generate the faulty medical equipment’s alerts. In the recent COVID-19 situation, lockdowns and social distancing are the prime factors behind the rapid adoption of IoT by healthcare institutions.

1.1. Literature review

This section elaborates on the various authentication schemes developed to protect the AmI-IoMT networks. A mutual authentication approach for sensor networks was developed by Deebak [11], which was later found vulnerable [12]. Inspired by Deebak, Chen et al. [13] extended Deebak’s approach [11]; however, Xu et al. discovered that the scheme was prone to replay and impersonation attacks [14]. Wang [15] constructed elliptic curve cryptography (ECC) based mutual authentication approach to counter password guessing and verifier attacks. Odelu et al. did cryptanalysis on Wang’s approach and found it susceptible to cyber-threats [16]. Similarly, Turkanovic et al. [17] designed a user authentication and key establishment scheme for a resource-constrained ecosystem. However, both Farash [18], and Chang [19] found Turkanovic et al.’s scheme as insecure. Chang et al. [19] also attempted to devise privacy-preserving authentication and key agreement protocol, but Gope et al. [20] found it resource expensive and susceptible to traceability. Das et al. suggested a fuzzy extractor and smart card-based user authentication approach, whereas Li et al. introduced a biometric and password-based user legitimacy verification method; however, both the schemes have never been tested in hostile circumstances and are computationally expensive.

1.2. Research gap and motivation

The integration of AmI-IoMT has significantly transformed healthcare. The new paradigm enables the patient to communicate remotely with the doctor, hence saving resources and reducing strain on healthcare facilities. Despite the fact that IoMT is advantageous, cyber analysts believe that it could put patients’ and medical specialists’ lives in danger [21]. Based on the literature, IoMT networks are susceptible to attacks due to: (i) use of an insecure wireless medium, (ii) absence of strong cryptography solutions due to limited power, memory, and processing capability [22], and (iii) the lack of cyber knowledge to end-users (medical practitioners), hence falling prey to attackers. Health-related information is sensitive, thus requires privacy [23]. Besides, integrity is also a very important aspect because a minor alteration by the attacker in the diagnosis report could result in a patient’s different medication. As many medical technologies support automation, any malicious activity could trigger a detrimental action resulting in unprecedented outcomes.

Cybersecurity specialists presented a variety of methods to mitigate vulnerabilities and threats, including public key and lattice-based encryption, digital signatures, and so on. However, most recommended approaches have been found vulnerable to attacks in addition to being resource intensive. Despite the existence of conventional approaches, medical institutions and stakeholders got affected severely; 41.2 million records were compromised, whereas 2013 breaches were recorded from 86 countries in 2019. Moreover, the existing security solutions are more or less centralized that may not work well for a geographically vast and large IoT ecosystem [24]. Furthermore, centralized solutions have a single point of failure, decreased efficiency for larger networks, and notable delay [25,26]. Blockchain is coined as a solution by security practitioners because it works on decentralized and geographically distributed technology with attributes such as immutability, transparency, and fault tolerance [27]. In summary, security and privacy in AmI-IoT networks can be ensured by implementing robust authentication and key exchange mechanisms using blockchain [28,29].

1.3. Research contributions

The following are our contributions:

- We identify vulnerabilities and research gaps in the AmI-IoT healthcare ecosystem.
- We propose a decentralized and lightweight authentication framework based on Ethereum smart contracts, fog computing, PUF, and biometrics.
- We investigate the robustness of the proposed framework in hostile situations to demonstrate its applicability to sensitive medical applications.
- We calculate the transaction costs of smart contracts to evaluate the framework’s appropriateness for resource-constrained environments.

1.4. Paper organization

The remaining structure of the paper is as follows: Section 2 discusses the preliminaries required for the proposed protocol. Section 3 explains the working scenario of the proposed protocol. Section 4
justifies the robustness of the protocol through formal security analysis. Section 5 shows the results and discusses comparison analysis. Section 6 concludes the paper and highlights the future scope.

2. Preliminaries

2.1. System model

Fig. 2 illustrates the scenario of a smart healthcare institution where stakeholders (doctor, administrator, etc.) use digital gadgets (laptop, etc.) to access the internet-enabled wireless sensor nodes embedded on the medical appliances and patients’ body (e.g., pacemaker) [30]. The administrator has the responsibility to register the legitimate staff members and the IoT sensor nodes (e.g., drones) in the blockchain network. A user can be any staff member of the healthcare institution with interest to access the medical reports and appliances [31]. On the contrary, the IoT sensor nodes (e.g., Zigbee-IEEE 802.15.4) are tiny in size and have limited computation abilities but powerful enough to sense the physical environment and relay the information to the user through the gateway [32]. The healthcare institution has deployed a resource-abundant network gateway (supports IEEE 802.3 and IEEE 802.11) with the prime responsibility to facilitate communications between user, administrator, IoT sensor node, and blockchain network. To reduce the computation burden, blockchain fog nodes (BFN) are deployed to provide a decentralized authenticity verification framework [33].

2.2. Adversary model

The adversary model aka attacker model describes the various possible threats and the resulting risks that arise due to cyber attacks [34]. According to the Dolev–Yao (DY) model, a cyber adversary is capable of eavesdropping, replaying, cloning, intercepting, injecting, phishing, modifications, malware, impersonation, privilege escalation, and man-in-the-middle attacks. The implications of cyber-threats in the IoMT environment depends on the attack duration and application sensitivity. Most often, the organization and affected parties incur financial losses, reputational harm, legal ramifications, and intellectual property theft. These threats could result in benign (temporary shutdown of medical services) to severe (endangering patients lives) impact on medical IoT networks.

2.3. Goals

This section discusses the goals of the proposed security protocol. The protocol must prohibit unauthorized access and prevents cloning. Besides, the protocol must be robust to resist notable cyberattacks, e.g., man-in-the-middle, replay, impersonation, etc. [35]. The protocol should only permit legitimate entities to initiate the session, and establish session keys to attain confidentiality. Most importantly, the authentication framework should not rely on a single server. Instead, it should be decentralized to prevent physical and denial of service attacks. These security goals must be tied to an efficiency goal, i.e., computations and communications required to achieve the security goals must not be enormous [36].

2.4. Physical unclonable function (puf)

The traditional authentication protocols are computationally expensive due to the use of public-key cryptography. Besides, these protocols also demand storage space in tiny user and IoT devices [19]. As the IoT nodes and user devices are subjected to physical capturing, it is necessary to protect them from cloning attacks. PUF provides a robust and resource economical solution to resist hardware threats. PUF enables the devices to prove their legitimacy without complex computations and storage requirements. A nanoscale variation during manufacturing makes every PUF of the integrated circuit (IC) unique. Mathematically, PUF can be defined as, \( R = P(C) \). It is apparent from the expression that the output response of the PUF depends upon the input challenge and the device executing it. It is noteworthy that any physical tampering with the PUF would destroy its original attributes [37].

2.5. Blockchain

It is a technology proposed by Satoshi Nakamoto (pseudonym) for enabling peer-to-peer (decentralized) secure transactions [38,39]. The blockchain information is stored in the form of transactions that are further contained in blocks [40]. The block consists of various elements, to name a few, timestamp, transaction details, gas consumed, current hash, parent hash, and nonce, etc. whereas, each transaction comprises transaction hash, timestamp, transaction fee, nonce, and input data, etc. The nodes of the network follow the consensus (e.g., proof of work)
(PoW), proof of stake (PoS) to decide the acceptance or rejection of the transactions. Each new transaction includes the hash of the previous block, establishing a relationship between transactions (chain). Few properties that make the blockchain robust and the most reliable are unforgeability, non-repudiation, resilience, and transparency [41].

2.6. Smart contract

The smart contract is a concept similar to physical contracts but in digital form. Smart contracts establish a binding between untrusted and unknown parties; smart contracts are scripted and stored in the blockchain network as transactions [42,43]. Unlike centralized approaches, smart contracts do not require a mediator for binding and execution, which eliminates third-party expenses and facilitates dispute-free transactions. It has several benefits over conventional physical contracts, like immutability, speedy execution, real-time access, inexpensive, immense precision, etc. This concept is originally introduced by 

\[ \text{ether} \] 

(ETH) to use the decentralized characteristics of blockchain properties that make the blockchain robust and the most reliable are [41].

3.2. User registration phase

A user (doctor, nurse, etc.) interested in accessing the IoT medical network has to register himself at the healthcare organization. The entire registration method is described in Fig. 3 and disclosed as follows:

**Step 1:** The user device (UD) prepares the message \( M_1^1 = (U_{R_{req}} || U_{ide}) \) comprising of registration request \( U_{R_{req}} \) and institute provided unique identity \( U_{ide} \) and delivers it to admin. It is worth noting that the user and admin communicate through a secure channel [6] during the registration process.

**Step 2:** The admin receives the request and store the \( U_{ide} \) into its device memory (DM). Subsequently, the admin prompts a challenge \( \mathbf{C} \) credentials into the blockchain network (BN). Admin also prepares a message digest (MD) of \( \mathbf{C} \) to the PUF of the UD.

**Step 3:** UD generates the response, \( R_U = P_I(C_U) \) for a given challenge. Subsequently, the biometric credentials of the user \( (\text{Gen} (= (S_1, S_2)) \) are generated and stored. Further, UD computes \( a = h(S_1 || U_{ide}) \) and send \( a \parallel R_U \) to the admin.

**Step 4:** Upon receipt of \( M_2^{1A} \), admin stores \( a, C_U, R_U \) into its DM and develop a smart contract (SC) as shown in Fig. 4, \( S_C^1 \) to register users on the blockchain network (BN). Admin also prepares a message digest (MD) of \( R_U \) for later use. Admin applies its digital signature, \( Z = (B_{U_{ide}}^\text{bio}, S_C^1) \), to counter forgery while deploying SC into the BN. Remarkably, the admin and BN exchange information via a public (insecure) channel [6] during the registration process.

**Step 5:** Miners in the BN decrypts \( B_{U_{ide}}^\text{bio} \) and \( S_C^1 \) from BN and \( S_C^1 \) to its DM. After successful deployment (transaction), miner reverts to the admin with these details \( T_H, T_2, T_3, B_C, C_A, T_1, T_2, M_A \).

**Step 6:** Admin verifies \( M_{AB} \) and stores the transaction hash (\( T_H \)), and contract address \( (C_1) \) into its DM. Admin constructs a message \( E(B_{U_{ide}}^\text{bio}, a||\parallel C_1^1) \) to add credentials of user and its device into the SC.

**Step 7:** Miners decrypt the message \( D(B_{U_{ide}}^\text{bio}, V) \) and adds the user \( (a) \) and device (\( i \)) credentials into the \( S_C^1 \) with address \( C_A \). Blockchain nodes verify the identity of user from the existing legitimate list in \( S_C^1 \), in case of coherence, blockchain nodes return an error, otherwise it is stored as transaction. Upon successful transaction, blockchain nodes send these details \( D(T_H, T_2, T_3, B_C, C_A, T_1, T_2, C_1^1) \) to the admin.

**Step 8:** Admin retrieves the transaction details from \( M_{AB} \) and stores \( T_H \) into its DM. Finally, the admin generates the pseudo-identity \( (P_{ij}) \) for anonymity and gateway secret \( (G_1) \) for mutual authentication and send it to the user as \( M_2^{1A} \).

**Step 9:** User stores the \( T_H, C_A^1, P_{ij}, G_1^1 \) for future use.

3.3. IoT sensor node registration phase

The admin enrolls the IoT sensor nodes (ISN) to declare them authentic. The registration helps the gateway to allow only authorized ISN to interact with the user. The whole manner is depicted in Fig. 5 and demonstrated as follows:
Step 1: Admin prompts the challenge $C_{SN}$ to ISN. The ISN and admin communicate through a secure channel during the registration process.

Step 2: The ISN inputs the $C_{SN}$ to generate PUF response, $R_{SN} = P_{SN}(C_{SN})$, and delivers $S_{SN}ID\parallel R_{SN}$ to admin.

Step 3: Upon $M^{2}_{AS}$ arrival, admin stores $C_{SN}$, $R_{SN}$, and $S_{SN}ID$ into its DM. Subsequently, admin develops a smart contract $SC^{1}_{SN}$ to register ISN on the BN. Due to space limitations and relative similarity with $SC^{1}_{U}$, $SC^{1}_{SN}$ is not presented. Afterwards, the admin computes message digest, $g = h(S_{SN}ID\parallel R_{SN})$ and digital signature, $\delta = E^{pr_i}(B_{adm}, SC^{1}_{SN})$ to preserve integrity and prevent non-repudiation. Remarkably, the admin and BN exchange information via a public (insecure) channel during the registration process.

Step 4: The BN miners decrypts $D(B_{pub}^{adm}, \delta)$ and deploys the $SC^{1}_{SN}$ into the BN. Upon successful deployment, the miner returns $T^{3}_{H}, T^{3}_{S}, B^{3}_{H}, B^{3}_{S}, C^{3}_{H}, C^{3}_{S}, T^{3}_{F}, T^{3}_{G}, M^{3}_{A}$ to the admin.

Step 5: Admin analyzes the $M^{3}_{AB}$ and stores the transaction hash ($T^{3}_{H}$), and contract address ($C^{3}_{H}$) into its DM. Besides, admin computes $\Phi (= E(B_{adm}^{pr_i}, g))$ and corresponds to BN for the addition of ISN ($\Phi$) into the list of authorized nodes.

Step 6: Miner decrypts $D(B_{pub}^{adm}, \Phi)$ and retrieves $h(S_{SN}ID\parallel R_{SN})$. Blockchain nodes verify the identity of ISN from the existing legitimate list in $SC^{1}_{SN}$. In case of coherence, blockchain nodes returns an error, otherwise it is stored in $SC^{1}_{SN}$. Upon successful transaction, blockchain nodes send these details $T^{4}_{H}, T^{4}_{S}, B^{4}_{H}, B^{4}_{S}, C^{4}_{H}, C^{4}_{S}, T^{4}_{F}, T^{4}_{G}, M^{4}_{AS}, M^{4}_{A}$ to the admin.

Step 7: The admin verifies $M^{4}_{AB}$ and stores the transaction hash $T^{4}_{H}$ into its DM. Finally, admin generates pseudo-identity of ISN ($P^{1}_{SN}$) for anonymity and gateway secret ($G^{2}_{SN}$) for mutual authentication and send it to ISN as $M^{3}_{AS}$.


### 3.4. Mutual authentication and key agreement phase

It is imperative to investigate users’ and ISN’s legitimacy before permitting them to converse with each other. The introduced protocol

```
pragma solidity ^0.5.1;
contract IoTHealthcare {
    uint public alpha; //message digest of biometric and identity {h(S1, UIde)}
    uint public t; //message digest of PUF response {h(RU)}
    string status = "duplicate entry"; //UNH: User and device
    function addUnD(uint _alpha, uint _t) public returns(string memory result) {
        if (_alpha == alpha && _t == t) {
            return status; //returns an error if the user and device are already registered
        } else {alpha = _alpha; //otherwise, successful registration
            t = _t;
        }
    }
    function inspectUnD(uint _alpha, uint _t) public view returns(uint32 result) {
        if (_alpha == alpha && _t == t) {
            return 1; //returns 1 if user and device are legitimate else returns 2
        } else {return 2;
    }
```
guarantees mutual authentication and secure key establishment. The complete approach is illustrated in Fig. 6 and explained as follows:

Step 1: The UD generates $N_U^1$ and transmits $N_U^1 \parallel P_{1U}^1$ towards gateway $(G_W)$. Communication between all entities occurs over insecure public channels.

Step 2: The $G_W$ inspects $N_U^1$ and begins to locate $P_{1U}^1$. Upon success, the $G_W$ chooses the corresponding $C_U$, $R_U$ pair and generates nonce, $N_U^1$. To ensure safe reyling of the challenge $(C_U)$, $G_W$ computes $\beta = C_U \oplus G_1^\Delta$. Additionally, $G_W$ computes $j = h(G_U^1 || R_U)$ to help UD in verifying the authenticity of $G_W$. Successfully, $N_U^1, \beta, j$ are sent to UD.

Step 3: The UD confirms the freshness of $N_U^1$ and derives $C_U = G_1^\Delta \oplus \beta$. Afterwards, UD inputs the challenge to PUF and retrieves the response, $R_U = P_{R}(C_U)$. Besides, the UD computes $\omega = h(G_U^1 || R_U)$ and finds the equivalency between $\omega \equiv j$. The identiciality ($\omega \equiv j$) indicates successful authentication of $G_W$ at UD, whereas non-equivalency ($\omega \not\equiv j$) results in connection termination. Further, UD prompts the user to enter biometrics, Rep $(Bio', S_2) = S_1$. UD generates $N_U^2$ and computes $\Delta = h(S_1 || U_{ide}) \oplus G_2^\Delta, \mu = G_2^\Delta \oplus R_U^2$ to prove its authenticity at $G_W$. At last, UD constructs $M_{U2}^G = \{N_U^2, \Delta, \mu, PI_{SN}, T_{U2}^H, C_A^1\}$ and send it to $G_W$.

Step 4: The $G_W$ verifies $N_U^2$ and derives $h(S_1 || U_{ide}) = \Delta \oplus G_2^\Delta$. $G_W$ retrieves $R_U = \mu \oplus G_2^\Delta$ and prepares $G_1 = h(S_1 || U_{ide}) \parallel h(R_U)$. Finally, $G_W$ send $T_{U2}^H, C_A^1$ and $G_1$ to BFN.

Step 5: BFN identifies the smart contract with $C_A^1$ address, and locates the transaction hash, $T_{U2}^H$. Subsequently, BFN examines the authenticity of user and device by checking the received information $h(S_1 || U_{ide}) || h(R_U)$. Hence, the already transacted information, $a \equiv 1$.

Step 6: The $G_W$ evaluates $Re_{SC}$, wherein $Re_{SC} = 1$ approves the genuineness of user and its device, and $Re_{SC} = 2$ indicates an adversarial attempt. $G_W$ terminates the connection in case of a malicious attempt. Otherwise, it continues. Upon authentication, $G_W$ initiates the process to connect UD with the $P_{1U}^{ISN}$. $G_W$ fetches the $C_{SN}, R_{SN}$ pair from its DM. Further, $G_W$ generates the nonce, $N_{ISN}^2$ and computes $\gamma = G_2^\Delta \oplus C_{SN}$ to send the challenge $(C_{SN})$ safely to the $P_{1U}^{ISN}$. Besides, $G_W$ calculates $\eta = h(G_2^\Delta || R_{SN})$ to help ISN in verifying the authenticity of $G_W$. Successfully, $M_{GS}^2 = N_{ISN}^2 || \eta \parallel \eta$ is composed and sent to $P_{1U}^{ISN}$ (a pseudonym of $SN_{ID}^{IOT}$).

Step 7: ISN verifies $N_{ISN}^2$ and derives the challenge, $C_{SN} = \gamma \oplus G_2^\Delta$. Afterwards, ISN calculates the response, $R_{SN} = P_{SN}(C_{SN})$ and use it to compute $\omega = h(G_2^\Delta || R_{SN})$. ISN evaluates the identiciality, $\omega \equiv \eta$; equivalency proves the authenticity of $G_W$ at ISN, whereas non-equivalency results in connection termination. Upon authentication of $G_W$, ISN generates $N_{ISN}^2$ and computes $\omega = G_2^\Delta \oplus R_{SN}, \omega = G_2^\Delta \oplus SN_{ID}^{IOT}$ to prove its legitimacy at $G_W$. Finally, the ISN drafts $M_{IG}^2 = N_{ISN}^2 || \omega, \omega || \omega || T_{ISN}^H, C_A^2$ and send it to $G_W$.

Step 8: $G_W$ examines the freshness of $N_{ISN}^2$ and retrieves the response, $R_{SN} = \omega || G_2^\Delta$ and real identity of ISN, $SN_{ID}^{IOT} || R_{SN}$ and delivers $M_{IG}^2 = T_{ISN}^H || C_A^2 || G_2$ to BFN for authenticity verification of ISN.

Step 9: BFN identifies the smart contract with $C_A^2$ address, and locates the transaction hash, $T_{ISN}^H$. BFN examines the authenticity of ISN by comparing the received information $h(SN_{ID}^{IOT} || R_{SN})$ with the already transacted information, $\omega$. BFN prepares the response code, $Re_{SC}$ and delivers it to $G_W$ as $M_{BG}^2, Re_{SC}$ discloses the success/failure of authentication.

Step 10: $G_W$ evaluates $Re_{SC}$, wherein $Re_{SC} = 1$ approves the genuineness of ISN, and $Re_{SC} = 2$ indicates an adversarial attempt. $G_W$ terminates the connection in case of adversarial attempt, otherwise continues. Furthermore, $G_W$ generates the $N_{ISN}^3$ and $N_{ISN}^3$ calculates the new pseudo-identities of UD $(P_{1ISN}^1 = G_3^\Delta \oplus P_{1U}^1)$ and ISN $(P_{1ISN}^2 = G_3^\Delta \oplus P_{1SN}^2)$. Finally, $G_W$ computes the secret session key for user $(SK^u = G_3^\Delta \oplus SK)$ and ISN $(SK^s = G_3^\Delta \oplus SK)$. The $G_W$ composes $M_{IG}^3 = N_{ISN}^3 || SK^u || P_{1ISN}^1 || P_{1ISN}^2$ and $M_{IG}^3 = N_{ISN}^3 || SK^s || P_{1SN}^2$ and send $M_{IG}^3$ to UD and ISN, respectively. It is noteworthy that neither secret key nor pseudo-identity is shared in interpretable form.

Step 11: UD verifies the freshness of $N_{ISN}^3$, and derives the $SK$ $(= SK^u \oplus G_3^\Delta)$ and $P_{1ISN}^1 = P_{1SN}^1 \oplus G_3^\Delta)$. The SK is used to secure the current session whereas $P_{1ISN}^1$ is stored in UD memory for ensuring anonymity in the subsequent session.
Step 12: Upon receipt of $M_{UG}$, ISN verifies $N_{4}$ and calculates the SK ($= SK^k \oplus G_2^k$) and $P_{ISN}^2$ ($= P_{ISN}^2 \oplus G_2^2$). The SK is used to secure the current session whereas $P_{ISN}^2$ is stored for ensuring anonymity in the subsequent session.

4. Security analysis

Scyther facilitates the security protocol developers to test the strength of their devised protocol against attacks. It offers simplicity in modeling cryptosystems, and also supports the DY model [45]. To operate Scyther, we have installed Scyther 1.1.3, Graphviz 2.46, Python 2.7, and wxPython 2.8, in the computing system that has Ubuntu OS (Linux, 64-bit). We have used the default settings of Scyther; typed matching and pruning method is used with a maximum no. of runs as 5 and maximum number of patterns per claim as 10, respectively. The protocol is scripted in Security Protocol Definition Language and begins with global constants and functions declaration succeeded by roles of individual entities that comprise computations, communications, and claims. The results obtained from the Scyther are presented in Fig. 7; it proves the robustness of the proposed protocol against MITM, and replay attacks, etc. The claims ‘secret’ verified the confidentiality of the message elements, whereas ‘Nisynch’, ‘Niagree’, ‘Alive’, and
M. Masud, G.S. Gaba, P. Kumar et al. Computer Communications 196 (2022) 45–54

is used to execute smart contracts while etherscan for consistency check. Metamask supports two distinct platforms (JavaScript VM (JVM), Injected Web3 (IW3)) programming language [46]. The SC is deployed and tested on Solidity Remix Integrated Development Environment (IDE) using

5. Performance and comparative analysis

The smart contract (SC) employed by the proposed protocol is realized in the Remix Integrated Development Environment (IDE) using Solidity programming language [46]. The SC is deployed and tested on two distinct platforms (JavaScript VM (JVM), Injected Web3 (IW3)) for consistency check. Metamask supported IW3 Ropsten Test Network is used to execute smart contracts while etherscan is used to access transaction logs [47]. Table 2 presents the gas and the transaction fee spent for deploying and registering the nodes on the decentralized blockchain network. It is worth noting that SC does not levy any transaction costs while verifying the authenticity of nodes. Based on the calculations, the deployment, registration, and duplicacy prevention in the JVM and IW3 environment would cost around $1.15, $0.11, $0.081, and $0.16, and $0.0681, respectively. The costs may vary because the crypto-currencies are very volatile. The aforementioned calculations are based on this relationship, $1 = $1765.88.

It is evident from Table 3 that the proposed protocol attained the security properties such as data privacy, message integrity, freshness, anonymity, untraceability, and biometric security. The accomplishment of security properties strengthened the proposed security protocol to withstand attacks like replay, impersonation, modification, DoS, MITM, and cloning, etc. Besides, the proposed protocol uses decentralized blockchain-powered Ethereum SC to overcome the demerits of centralized infrastructure. It is apparent from Table 3 that existing schemes [19,20,36,48–52] are not able to resist all prominent attacks. Further, it is exposed that none of the traditional schemes [19,20,36,48–52] adopted a decentralized approach, thus resulting in decreased efficiency.
The computation cost calculations for the registration phase is omitted because it incurs only once during initialization. As evident from Table 4, the proposed protocol is computationally economical because it uses lightweight cryptography primitives (hash, XOR, and PUF) instead of computing expensive cryptography primitives (public-key cryptography, scalar multiplications). Fig. 8 demonstrates that the entities in the proposed scheme (user device, gateway, IoT sensor node) executes the cryptographic operations fewer times than the entities in the conventional schemes [19,20,36,48–52], indicating that it is computationally inexpensive. As apparent from Table 4 and Fig. 8, a few traditional schemes [19,50] have the reasonable computation, but those schemes [19,50] do not guarantee the complete security of the IoT networks. Therefore, the proposed protocol can be the best alternative to the existing compute expensive protocols.

### 6. Conclusions and future scope

Healthcare 4.0 in AmI environments is a technology-driven and patient-centric paradigm where IoT sensor nodes automatically operate, collect the information from medical equipment, and export it to the cloud. To protect sensitive health information from adversarial threats, authentication approaches were developed in the past. However, those schemes were infrastructure-centric, computationally expensive, and prone to adversarial threats. Therefore, we have employed PUF, blockchain-powered SC, and fog nodes in our proposed authentication protocol to circumvent SPOF, prevent cyber-attacks, and enhance efficiency. The Ethereum SC is developed in Remix IDE using solidity, executed on metamask RTN, and used to verify the authenticity of network entities at minimal cost. We have verified the resiliency of the protocol against attacks using Scyther. Due to nominal computation cost, the proposed scheme finds its applicability in resource-constrained IoT based healthcare networks. Zero-Knowledge Proofs (ZKPs) and homomorphic encryption will be applied to perform confidentiality-preserving authentication and processing of information, hence extending the security and privacy of the IoT-enabled healthcare applications.

### CRediT authorship contribution statement

**Mehedi Masud:** Conceptualization, Methodology, Software, Validation, Formal analysis, Data curation, Writing – original draft, Writing – review & editing, Project administration.

**Gurjot Singh Gaba:** Conceptualization, Methodology, Software, Validation, Formal analysis, Data curation, Writing – original draft, Writing – review & editing, Project administration.

**Andrei Gurtov:** Methodology, Resources, Writing – original draft, Writing – review & editing, Supervision.

**Supervision:** Methodology, Resources, Writing – original draft, Writing – review & editing, Supervision.

**Funding acquisition.**

### Acknowledgment

The research is supported by Taif University Researchers Supporting Project Number (TURSP-2020/10), Taif University, Taif, Saudi Arabia, CENIIT Project 17.01, and the Excellence Center at Linköping-Lund in IT (ELLIIT) Project A4.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Data availability

No data was used for the research described in the article.

### References


