BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 72(5), 2024, Article number: e151049
DOI: 10.24425/bpasts.2024.151049

CONTROL, INFORMATICS AND ROBOTICS

Secure system with security protocol for interactions in healthcare Internet of Things

Sabina SZYMONIAK*

Department of Computer Science, Częstochowa University of Technology, Poland

Abstract. The Internet of Things is a network of connected devices that can communicate and share data over the Internet. These devices often have sensors that collect data for various purposes, such as usage statistics, data processing, or performing specific actions based on the collected data. Also, medical Internet of Things devices are crucial in monitoring critical functions, measuring blood glucose levels, indicating when patients require medicine, and ensuring timely medication delivery. Communication in the Internet of Things is demanding, requiring diverse protocols that address communication security concerns. These protocols must be robust and secure, considering technical factors such as the network objective, energy requirements, and the nature of the communication because they can be exploited. This paper proposes an innovative system with a security protocol that supports and improves communication security in modern Internet of Things networks. The protocol aims to enhance communication safety between interconnected devices for information exchange in medicine or healthcare, ensuring the confidentiality and integrity of sent data and devices. The proposed protocol, tested through formal and automated verification, meets all security goals, including identity verification, anonymity protection, and access revokement. It also protects against man-in-the-middle, modification, replay, and impersonation attacks.

Keywords: security protocol; security; Internet of Things; healthcare.

1. INTRODUCTION

The Internet of Things (IoT) is a network of connected devices that can communicate with each other and share data with users over the Internet. IoT devices can connect to the Internet and are often equipped with sensors that enable them to collect data. Depending on the use of IoT devices, data may be collected for various purposes, such as usage statistics, data processing or performing specific actions based on the collected data. The Smart Home is one of the areas where IoT solutions are widely used. So, we can turn on an intelligent vacuum cleaner at work. Also, the vacuum cleaner can learn our habits and daily routine and turn on itself when we perform official duties [1].

Also, we can meet the IoT solutions in medicine or healthcare. Medical IoT devices play a crucial role in monitoring the critical functions of individuals with chronic illnesses, measuring blood glucose levels in individuals with diabetes, indicating when patients require medicine, and ensuring timely delivery of medication to the patients [2–4]. In the case of healthcare, the athletes utilize these solutions to regulate essential bodily functions and optimize performance while mitigating the risk of potentially fatal circumstances [5–7].

Communication in IoT is demanding. Primarily it is facilitated by the following standards: IEEE 802.15.4 [8], NFC [9], 6LoWPAN [10], MQTT [11], and Bluetooth Low Energy [12]. On the other hand, communication among IoT devices necessitates the utilization of diverse protocols that delineate the objective of the communication, the sequential execution of procedures involved, and the cryptographic methods employed to safeguard the sent data. The primary objective of the protocol is to facilitate device-to-device communication while addressing communication security concerns. The protocol aims to achieve mutual authentication of the parties involved and establish agreement on the session key. Typically, these protocols are referred to as security protocols. Ensuring communication security is imperative in light of the potential for diverse cyberattacks. Malignant individuals may attempt to intercept and manipulate transmitted messages, as well as pilfer sensitive information [13–16]. Furthermore, the deployment of protocols in networks of interconnected IoT devices must consider technical factors such as the network objective, the energy requirements of the devices, and the nature of the communication that will occur inside it [17].

The mentioned protocols are susceptible to exploitation by malevolent users who actively seek security weaknesses to intercept and exploit the data. The user identification and key agreement stages warrant particular emphasis, as the security of the following communication phases relies on their secure execution. We can indicate some typical cyberattacks in IoT solutions. For example, denial-of-service attacks are designed to overload IoT devices, rendering them inoperable or unavailable [18]. Brute-force attacks aimed at guessing passwords for IoT devices [19]. Attacks on protocols used in IoT networks aimed to intercept data violate the integrity of transmitted information or other forms of sabotage (for example, Key Compromise Impersonation attack [20]). To protect IoT networks against these and other attacks, it is essential to employ appro-

*e-mail: sabina.szymoniak@icis.pcz.pl

Manuscript submitted 2024-05-06, revised 2024-06-24, initially accepted for publication 2024-07-08, published in September 2024.

priate security practices, including regular software updates and robust authentication mechanisms.

Considering the obstacles and requirements associated with IoT solutions for medicine and healthcare, we proposed a new communication method. So, this paper introduces a novel system supported by security protocol that ensures safe communication in modern IoT networks. The suggested protocol aims to enhance communication safety between interconnected devices to exchange information in medicine or healthcare. Nevertheless, this protocol is equally applicable to diverse Internet of Things devices linked to a network that facilitates the transmission of certain data packets. The protocol functions inside a network of diverse devices responsible for data collection, transmission, and reception. We aimed to improve the security of contact and data exchange between the doctor, the patient, the coach, and his protégé. The suggested protocol guarantees the confidentiality and integrity of the sent data and devices. We validated this by formal verification with the BAN logic [21], informal analysis and automated verification utilizing the tool outlined in [14]. The suggested protocol guarantees the verification of both parties' identities, the protection of their anonymity, and the ability to revoke access if necessary. Additionally, it protects against man-in-the-middle, modification, replay, and impersonation attacks. The protocol efficiency is contingent upon the computational capabilities of the devices and the nature of the communicated data.

This paper's primary contributions are as follows:

- the proposition of the secure system for interactions in healthcare Internet of Things,
- the proposition of security protocol for communication in medicine or healthcare,
- formal, informal and automatic security analysis.

The subsequent sections of this work are structured in the following manner. Section 2 provides an account of the works closely connected to the topic. Section 3 outlines our assumptions of the network, the proposed protocol operational environment, and the proposed system security policy. Section 4 explains the organization of our protocol. In Section 5, we provide a comprehensive study of the security of our protocol, both in formal and informal terms. The final section will summarize the entire article, provide the research results, and outline our future goals.

2. RELATED WORKS

We can use many different security protocols in IoT environments. We can divide them according to the environment in which they operate and the security goals they pursue. For example, we can indicate the following solutions regarding security protocols for medical or healthcare solutions.

Attir *et al.* [22] researched security measures specifically designed for wireless body area networks. The authors introduced a streamlined protocol for devices with limited resources in wireless body area networks. The protocol employs XOR operations and lightweight hashes. Furthermore, supplementary network nodes possessing greater resources engage in computations and data processing derived from sensors. The authors validated the safety of the suggested method by employing the BAN logic [21] and the AVISPA tool [23].

Nyangaresi [24] introduced a three-factor authentication protocol for securing the medical Internet of Things in their publication. This procedure encompasses the patient's biometric data, smart card, and password. The authors verified the safety of their solution using the Real-Or-Random model [25] and BAN logic. Ija *et al.* in [26] addressed the concerns surrounding the security of medical IoT. The authors employed blockchain technology, elliptic curve cryptography, and physical unclonable functions.

Wang *et al.* [27] introduced a protocol designed for medical Internet of Things systems. This protocol safeguards Patient data from unauthorized servers, preventing illegal access. The authors devised an encryption technique for this protocol utilizing cyclic shift and XOR operation. The protocol ensures user safety without imposing excessive demands on devices. The authors validated the security of the proposed protocol by employing the BAN logic. Furthermore, they have demonstrated the protocol resilience against common assaults in IoT contexts. The authors also conducted a comparative analysis of their protocol with similar solutions and achieved satisfying outcomes in ensuring safety qualities and optimizing energy consumption during communication and calculations.

Rasslan *et al.* [28] have introduced identity-based robust designated verifier signature authentication procedures for the medical Internet of Things systems. The proposed protocols can facilitate the authentication process of the IoT device network, encompassing both conventional devices responsible for monitoring patients' vital functions and autonomous cars and drones. Both methods have the trait of having a compact signature size. Furthermore, the authors demonstrated that both strategies exhibit minimal communication and computing expenses compared to comparable solutions. The authors have verified that the suggested protocols adhere to the assumptions of the Random Oracle Model (ROM) [29], safeguard patient's privacy, and guarantee data integrity and authenticity.

Masud *et al.* have introduced an authentication protocol for medical Internet of Things systems in [30]. The proposed protocol utilizes blockchain [31], fog computing, Ethereum-powered smart contracts [32], physical unclonable functions (PUF), and biometrics. Blockchain and fog technology guarantee nonrepudiation, transparency, minimal delay, and optimal bandwidth utilization. Additional technologies are employed to mitigate the risks of replay, spoofing, and cloning attacks. The authors verified and validated the protocol security using the Scyther tool. Moreover, they compared their protocol with similar computing costs and performance options. The authors demonstrated the efficacy of the suggested procedure in healthcare networks employing resource-constrained devices.

Chen *et al.* in [33] presented the LAP-IoHT protocol in their publication, a three-factor authentication protocol specifically developed for Internet of Things solutions in the healthcare domain. Authentication relies on using smart cards, passwords, and biometric characteristics. The authors performed a safety study of the suggested regimen using the ROR model. Studies have demonstrated that the protocol exhibits resistance against replay attacks, user impersonation attacks, server impersonation attacks, privileged insider attacks, KSSTI attacks, and stolen smart card attacks. Furthermore, the protocol guarantees flawless forward secrecy. The authors demonstrated that the LAP-IoHT protocol exhibits superior computational efficiency compared to similar alternatives and incurs little communication expenses.

3. SYSTEM ASSUMPTIONS

This section will outline our assumptions about the network and environment in which the proposed protocol will function. Every protocol must guarantee superior security in transmitting information among distinct entities. Consequently, it should possess data secrecy, integrity, authenticity, attack resilience, and scalability. Data confidentiality pertains to the assurance of preventing any unauthorized access to the sent information. Data integrity refers to guaranteeing that data remains unchanged or unaltered during transmission. Authenticity pertains to verifying the identification of persons or systems engaged in communication. Attack resistance guarantees the protection of users and transmitted data from various network-based attacks. Scalability facilitates secure communication among many users or systems while accommodating new ones without requiring a complete protocol overhaul. Furthermore, the protocol must incorporate AAA (authentication, authorization, accounting) logic, which encompasses the tasks of user authentication inside the network, enforcement of user rules, and recording of session statistics [16, 34].

Considering the multiple facets and challenges of ensuring secure communication in IoT systems, we present a new approach to securing communication between Patients and doctors or coaches. We assume that in the devices network, we have included four roles assigned to specific types of devices: Patients' Healthcare IoT Devices (PHIoTD), patients' smartphones (PS), guards' smartphones (GS) and the trusted server (TS). A network architecture is shown in Fig. 1.

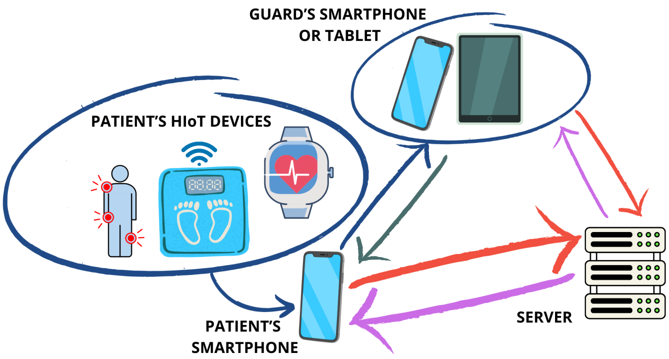

Fig. 1. The architecture of the proposed system

The proposed system is dedicated to exchanging health information between the Patient and a Guard, who may be a doctor or a trainer. The system and communication process work as follows. The Patient, i.e. who wants to exchange health information with the Guard, has a PHIoTD kit. PHIoTD is an IoT device that enables control of Patient health parameters, e.g. smartwatch, smart band, smart scale, or other sensors. The Patient then connects the selected PHIoTDs and the system support application on his PS. The type of connected PHIoTDs depends on the health parameters that the patient wants to control, e.g. weight, heart rate, blood pressure, and blood sugar level. The assistive application collects data from the sensors and forwards it to the assistive application installed on the GS. After learning about the Patient's health parameters, the Guard can provide him with guidelines regarding further actions, e.g. changing the training plan or guidelines regarding the nutrition plan.

The last device in the proposed system is the trusted server (TS). Its tasks include generating anonymous identifiers, password generation, key generation, storage, and handling of account data. The server has a hardware random number generator [35].

The security protocol is responsible for the security and course of communication. It consists of the following phases: creating an account, establishing a symmetric key and authentication, communication, password and key reset, and account deletion. All phases will be described in detail in the next section.

All operations related to protocol execution are supported by the client and server applications. So, the initial state of the protocol is as follows. The Patient or Guard must install the client application that enables the operation and execution of each protocol phases, generating private and public keys and sharing the public key for the server. The private and public key pair is automatically generated according to RSA algorithm [36] when the user installs the client application. The server public key K_S^+ is downloaded with the application and registered on the device. Also, they must know the second user's unique user number identifier to submit a request to the server to establish a symmetric key. The server must have the application installed that enables the operation and execution of each protocol phase, generating private and public keys and sharing the public key with the users. Also, the server must have the appropriate computing power to handle all requests and memory resources to store all information necessary for communication, including devices public keys related to users' accounts. Biometry ensures the security of all information stored in the device database. Launching the application requires confirming the user's identity using a fingerprint or iris scan. So, after correctly logging into the application, the user has access to all symmetric keys, and he or she can contact the selected user.

One of the elements generated when executing the protocol is the user's password. The password is generated in two situations: when creating and resetting an account. Storing passwords in a database on a server is a problematic issue due to the potential for cyber attacks that could result in hackers stealing the application database (e.g. SQL Injection [37] or Cross-Site Request Forgery [38]). Therefore, we are introducing the following policy for transmitting and storing passwords in the system. Firstly, the supporting application running on the server will generate a random value, the so-called pepper, which will be placed in the configuration files. Each generated password will be hashed using a one-way hash function of the administrator's choice. We suggest using the bcrypt algorithm [39]. The hashed password is sent to the user. However, before writing to the database, the pepper value is added to the hashed password and the whole thing is hashed again. The string of characters processed in this way is saved in the database. This solution protects the user's password against cracking in case of a database leak. When the server wants to check whether the user has sent the correct password, it retrieves the pepper value from the configuration files, adds it to the received hashed password, hashes the resulting string of characters and compares it with the value in the database.

During the proposed system operation, situations related to key loss due to physical loss of the device (loss, destruction, theft, replacement) or restoring the device to factory settings may be problematic. The symmetric key shared between the Patient and the Guard is established during the establishing a symmetric key and authentication phase of the proposed protocol. The generated keys are automatically saved in the applications on the users' devices so the users can communicate.

The server application supports the mentioned problematic situations. When the user reinstalled the client application and logged in, the user must select the "New device" option because the user does not have a symmetric key for communication. Selecting this option will inform the server that the user does not have the previously used device (from a physical or configuration point of view). Therefore, the server will perform the part of establishing a symmetric key and authentication phase by generating a new key for the Guard and Patient and automatically resetting the password and key for both users. Similarly, the server will enable resetting the key shared between the server and the user with the new device to enhance security. Also, new device registration causes the generation of new asymmetric key pair. Thus, the server updates its database information about the device public key.

Moreover, in a situation where the Guard handles many Patients or the Patient communicates with several Guards, the client application supports choosing a Patient or Guard for communication. So, the application is responsible for using the appropriate symmetric key during communication. Also, in a situation where multiple users communicate with a server using individual symmetric keys, the server must have a way to identify which symmetric key to use to decrypt each incoming message. There are several methods to identify the appropriate decryption key. In our approach, we used the method of dedicated identifiers. When creating an account, each device generates its identifier, which is explicitly attached to the transport header of the network protocol. The server uses this identifier to find the appropriate key in its database. The unique device identifier is not used in any way when executing the proposed protocol to strengthen communication security. Moreover, in both situations, all necessary symmetric keys are stored in a database on the user's (Patient or Guard) device. Biometry ensures the security of the database. So, first, the Patient or Guard must launch a client application, confirm the user's identity using a fingerprint or iris scan, and then contact the other user.

4. SECURITY PROTOCOL

In this section, we present the proposed security protocol. Considering the previously described password generation and storage policy, we will consider the term *password* sent in messages as a hashed password to simplify the description.

4.1. Notation used in this protocol

Firstly, we will present a notation used for the proposed protocol description. In our notation:

- $\{M\}_K$ means the message M encrypted by key K,
- S means the trusted server,
- P means the Patient,
- G means the Guard,
- U means the system user (both the Patient and Guard in creating an account),
- $i(U)$ means a device text identifier that is the user's e-mail; for example, $i(P)$ means Patient's identifier (e-mail),
- UID_U means unique user number identifier, for example, UID_P means Patient's number identifier,
- T_U means a timestamp generated by user U, for example, T_S means a timestamp generated by the trusted server S,
- ${A_{id}}^u$ means user u's account identifier,
- U_{pass} means the user's password generated and hashed by the trusted server,
- K_u^+ is the public key of user u, for example the key K_S^+ is the public key of the trusted server,
- K_{SU} means the symmetric key shared between the trusted server and the user U, for example, K_{SP} means the symmetric key shared between the trusted server and the Patient,
- K_{PG} means the symmetric key shared between the Patient and Guard,
- N_P means a session identifier (nonce) generated by the by the user initiating communication,
- $\phi(N_P)$ means the result of ϕ-function on N_P,
- FH_P is the Patient's healthcare information,
- FH_G is the Guard's healthcare feedback.

4.2. Creating an account phase

The creating an account phase is the first phase of the protocol. This phase must be completed by each Patient and Guard who wishes to contact each other. By creating an account, further communication will be possible. The communication flow in this phase in Alice-Bob notation is as follows:

$$\begin{array}{lll} \alpha_1 & U \rightarrow S: & \{i(U) \cdot A_{id} \cdot T_U\}_{K_S^+}, \\ \alpha_2 & S \rightarrow U: & \{\{K_{SU}\}_{K_U^+} \cdot \{UID_U\}_{K_U^+} \cdot \{U_{pass}\}_{K_U^+} \cdot T_S\}_{K_U^+}, \\ \alpha_3 & U \rightarrow S: & \{T_S\}_{K_{SU}}. \end{array}$$

In the first step of this phase, a new system user reports to a trusted server his or her will to set up an account. To do this, it sends a message to the server containing its identifier, account identifier and a freshly generated timestamp. The user encrypts the entire message using the server public key. The account identifier allows us to determine whether a Patient or a Guard created the account. The system-supporting application automatically generates the identifier value. Before sending a response to the user, the trusted server performs the following actions:

- generates and hashes a 512-bit password for the user,
- generates a unique numeric user identifier,
- generates a symmetric key shared between the user and the server,
- saves user information to the database.

The user's email address is not saved in the database. In further communication, the user uses only the numerical identifier. Hence, by identifier in the following descriptions, we mean number identifier.

After performing the abovementioned operations, the trusted server sends the user a message consisting of two elements. The first element is a symmetric key encrypted with the user's public key. The second element is the user's password, encrypted with a new symmetric key. The user's public key encrypts both elements and the server timestamp. In the third step, the user confirms receipt of the message by sending its timestamp encrypted with the received symmetric key to the server.

4.3. Establishing a symmetric key and authentication phase

Two security goals are achieved by establishing a symmetric key and authentication phase. The first is establishing and distributing a symmetric key, and the second is the authentication of the Patient and Guard against a trusted server. The communication flow in this phase in Alice-Bob notation is as follows:

$$\begin{array}{lll} \alpha_1 & P \rightarrow S: & \{UID_P \cdot UID_G \cdot T_P\}_{K_{SP}}, \\ \alpha_{2_1} & S \rightarrow P: & \{A_{id}^G \cdot \{K_{PG}\}_{K_P^+} \cdot UID_P \cdot UID_G \cdot T_S\}_{K_{SP}}, \\ \alpha_{2_2} & S \rightarrow G: & \{A_{id}^P \cdot \{K_{PG}\}_{K_G^+} \cdot UID_P \cdot UID_G \cdot T_S\}_{K_{SG}}, \\ \alpha_{3_1} & P \rightarrow S: & \{T_S\}_{K_{SP}}, \\ \alpha_{3_2} & G \rightarrow S: & \{T_S\}_{K_{SG}}. \end{array}$$

Establishing a symmetric key and authentication phase consists of five steps, where the server can perform the second and third steps in parallel. Steps four and five are the responses of two different users to messages from the server. Hence, we introduced the symbol of parallel execution in the numbering of these steps.

In the first step, the Patient informs the server that he wants to contact his Guard. For this purpose, it sends its identifier, Guard identifier and a new timestamp to the server. Everything is encrypted with a symmetric key shared with the server. For executing steps α_{2_1} and α_{2_2}, the server generates a symmetric key that the Patient and Guard will share and a timestamp. Then, it prepares two messages containing this key, the Patient and Guard identifiers, and a timestamp. The first message is intended for the Patient. Therefore, the server encrypts the newly generated key by Patient's public key and the entire message using the K_{SP} key. The second message, in turn, is intended for Guard, so the server encrypts the newly generated key by Guard's public key and the entire message using the K_{SG} key. In these two messages, the server sends to users account identifiers (A_{id}^P to G and A_{id}^G to P). Based on this, the received identifiers will be associated with the appropriate keys in the device database. This will make it easier to distinguish the keys and carry out further communication. This identifier will be explicitly attached to the transport header of the network protocol. So, the devices will use it to find the appropriate key in its database.

As mentioned, the last two steps are the Patient and Guard's replies to earlier messages. In both steps, users confirm the receipt of the message and their identity towards the server by sending its timestamp in messages encrypted with appropriate symmetric keys.

If one of the users will not execute step α_{3_1} or α_{3_2} before the server timestamp expires, the server will delete information about this account from the second application. So, the second user will not be able to include this Patient or Guard in communication.

4.4. Communication phase

The most important and most complex stage of the protocol is the communication phase, during which the Patient and Guard exchange health information without the participation of a trusted server. The communication flow in this phase in Alice-Bob notation is as follows:

$$\begin{array}{lll} \alpha_1 & P \rightarrow G: & A_{id}^P \cdot \{UID_P \cdot N_P \cdot T_P\}_{K_{PG}}, \\ \alpha_2 & G \rightarrow P: & \{UID_G \cdot \phi(N_P) \cdot T_G\}_{K_{PG}}, \\ \alpha_3 & P \rightarrow G: & \{FH_P \cdot T_P\}_{K_{PG}}, \\ \alpha_4 & G \rightarrow P: & \{FH_U \cdot T_G\}_{K_{PG}}, \\ \alpha_5 & P \rightarrow G: & \{T_P \cdot T_G\}_{K_{PG}}. \end{array}$$

In the first step of the communication phase, the Patient informs the Guard that he wants to start the session. For this purpose, it sends a message to Guard containing its identifier, timestamp, and a generated random session identifier. The session identifier helps the system connect queries and responses between communicating users. Also, the Patient sends its identifier (A_{id}^P). In response (step two), the Guard sends the Patient consent to establish a connection. The message includes its identifier, timestamp, and modified session identifier. Modifying the session identifier involves executing the ϕ function on it. This function can be any freely selected and simple arithmetic operation (addition, subtraction, multiplication, division), e.g. increasing it by one. In the third step, the Patient sends his health data in JSON format with a time stamp. After reviewing the data, the Guard can prepare a set of comments and guidelines for the Patient. In step four, the guidelines are also sent in JSON format with a time stamp. In the last step of this phase, the Patient confirms receipt of the information and closes the session by sending two timestamps to the Guard. All messages in this phase are encrypted using a symmetric key shared between the Patient and Guard.

4.5. Password and key reset phase

If one of the users, for some reason, would like to obtain a new symmetric key for communication with the other user, he or she must perform the password and key reset phase. Both the Patient and the Guard can perform this phase at any time. During this phase, the trusted server generates the user's new password and the key he will share with his interlocutor. The communication flow in this phase, for a key reset request by Guard, in Alice-Bob notation is as follows:

$\alpha_1 \quad G \to S: \quad A^G_{id} \cdot \{UID_G \cdot \{U_{pass}\}_{K^+_S} \cdot UID_P \cdot T_G\}_{K_{SG}},$

$\alpha_{2_1} \quad S \to G: \quad \{\{K_{PG}\}_{K^+_G} \cdot \{U_{pass}\}_{K^+_G} \cdot T_S\}_{K_{SG}},$

$\alpha_{2_2} \quad S \to P: \quad \{\{K_{PG}\}_{K^+_P} \cdot T_S\}_{K_{SP}}.$

In the first step of this phase, the Patient submits a request to the server to reset the symmetric key. For this purpose, it sends a message that contains his identifier, a password encrypted with the server public key, the Patient's identifier and a freshly generated timestamp. The entire message is encrypted using a key shared between the server and the Guard. Also, the Guard sends its identifier (A^G_{id}). After receiving the message, the server checks the correctness of the sent data (ID and password) in its database. If the data does not match, it stops communication. If the data is correct, it generates a new password and key and notifies both the Guard and the Patient about this fact simultaneously (step two). Both messages contain a new symmetric key encrypted with the user's public key and a server timestamp. The message to Guard (the request initiator) additionally contains a new password encrypted with Guard's public key. Both messages are encrypted with appropriate keys shared with the server.

If the Patient initiates the request, the communication course will be identical. The difference will only consist of the cryptographic objects used in individual steps.

4.6. Account deletion phase

The proposed system also allows users to delete their accounts and data from the server. To delete the account and its data, any user (Patient or Guard) must complete the account deletion phase. The communication flow in this phase for the Patient's request to delete the account, in Alice-Bob notation, is as follows:

$\alpha_1 \quad P \to S: \quad A^P_{id} \cdot \{UID_P \cdot \{U_{pass}\}_{K^+_S} \cdot T_P\}_{K_{SP}},$

$\alpha_2 \quad S \to P: \quad \{T_P \cdot T_S\}_{K_{SP}},$

$\alpha_3 \quad S \to G: \quad \{UID_P \cdot T_S\}_{K_{SG}}.$

In the first step, the Patient submits a request to the server to delete the account. For this purpose, it sends a message that contains its identifier, a password encrypted with the server public key, and a freshly generated timestamp. The entire message is encrypted using a key shared between the server and the Patient. Also, the Patient sends the identifier (A^P_{id}). After receiving the message, the server checks the correctness of the sent data (ID and password) in its database. If the data does not match, it stops communication. If the data is correct, it confirms the deletion of the account by sending its timestamp to the Patient, encrypted with a key shared with the Patient (step two). Then, it deletes the Patient's account data from the database. He then notifies Guarda that no further communication with the Patient will be possible. For this purpose, step three sends the Patient's identifier and a time stamp to the Guard, encrypted with a key shared with the Guard.

If the Guard initiates the request, the communication process will be identical. The difference will only consist of the cryptographic objects used in individual steps.

5. SECURITY ANALYSIS

Next, we performed three types of security analysis: formal analysis using BAN logic [21], informal analysis and automated analysis using a tool from [14].

5.1. Formal analysis

We performed formal analysis for each phase separately. We noticed that each assumed goal was achieved from the whole protocol perspective, but not every goal was achieved in every phase. For example, assumed goals that include a trusted server may not be achieved during the communication phase because the trusted server does not appear in this phase. Based on this, we determined which goals should be achieved during the formal analysis of each phase using BAN logic. Also, we made assumptions and observations based on schemes mentioned in earlier sections. Due to the page limit, we will summarize that each assumed goal was achieved.

5.2. Informal security analysis

Subsequently, we conducted an ad hoc security study to verify that our protocol successfully attains the primary security features. We examined the subsequent security attributes:

- Mutual authentication requires the user (Patient or Guard) and the server to possess a shared symmetric key to authenticate each other. The server generates the key during the second step of the creating an account phase. During this stage, individuals verify their identity by transmitting their timestamps. During the establishing a symmetric key and authentication phase, individuals verify their identity by transmitting credentials such as timestamps of users and server timestamps.
- Anonymity – The user's identity is safeguarded through asymmetric cryptography during the creating an account phase and using an anonymized user identifier.
- The server has a revocation mechanism linked to its work schedule. It can terminate each connection in case of technical difficulties or if the user fails to authenticate.

Furthermore, we conducted an ad hoc security analysis to verify the potential vulnerabilities that could impede the effectiveness of our protocol. We analyzed the subsequent offensive scenarios:

- Our protocol effectively mitigates man-in-the-middle attacks by employing a secret key, denoted as K_{SU}, which is shared among the involved parties. The assailant is unable to extract any constituent from messages that have been encrypted with this key. If the assailant seizes a message of this nature and attempts to transmit it during a different session, the recipient will decline the message upon timestamp verification.
- Modification attack – the user's credentials, such as passwords or fingerprints, are protected by encryption using a secret key K_{SU}, preventing the attacker from making unauthorized changes. Should the attacker gain the ability to alter

the user's credentials, our protocol employs one-way hash algorithms. The server securely keeps cryptographic user credential hashes to ensure tampered communications are promptly rejected. Our technique effectively mitigates modification attacks.

- Replay attack prevention – our protocol effectively mitigates replay attacks by implementing a timestamp mechanism. Before any further action, each recipient validates the timestamp. If the message is received within the acceptable threshold, the communication will proceed; otherwise, the receiver will terminate the communication.
- Impersonation attacks are mitigated by our protocol through symmetric and asymmetric cryptography, a timestamp mechanism, and one-way hash functions.

5.3. Automated analysis

Subsequently, we conducted an automated validation of our methodology utilizing a program referenced in [14]. We have activated the feature known as timed analysis. The tool evaluates executions by employing predetermined time parameter values. The tool utilizes an abstract temporal unit that encompasses any given duration. For this verification, we have established the following assumptions:

- time of generating confidential information: $T_g = 1[tu]$,
- time of composing the message $T_c = 3[tu]$,
- asymmetric encryption time $T_e = 5[tu]$,
- asymmetric decryption time $T_d = 5[tu]$,
- symmetric encryption time $T_e = 3[tu]$,
- symmetric decryption time $T_d = 3[tu]$,
- minimal delay in the network $D_{min} = 1[tu]$,
- maximal delay in the network $D_{max} = 5[tu]$.

We considered that some steps in the three phases of the proposed protocol were performed by the server in parallel (including the establishing a symmetric key and authentication phase, during which the server sends users their shared key). This means that the execution times of these steps have been combined. However, the times of the same operations during parallel steps were counted once, for example, when generating confidential information was sent to both the Patient and the Guard. In turn, encryptions and decryptions were counted separately. During timed analysis, the mentioned tool generated the following number of executions for phases:

- creating an account: 4,
- establishing a symmetric key and authentication: 14,
- communication: 7,
- password and key reset: 8,
- account deletion: 8.

The generated executions differ in the order in which users appear in each phase and the capabilities of the Intruder according to the models included in the tool.

Tables 1 and 2 show the obtained timed results. Table 1 summarizes the timed analysis of the operation executed in each step. Minimal (T^k_{min}) and maximal (T^k_{max}) step times include times of encryption, decryption, generating, composing and delay in the network. The disparity between the minimum and maximum step time is contingent upon the delay in the network time. The timeout value (T^k_{out}) is linked to the duration of waiting for a response. Table 2 contains about minimal (T^{ses}_{min}) and maximal (T^{ses}_{max}) session time for each protocol phase using assumed delay in the network values.

Table 1

Values for T^k_{min}, T^k_{max} and T^k_{out} for each protocol phases

Phase	Step	T^k_{min}	T^k_{max}	T^k_{out}
Creating an account	α_1	15	19	19
	α_2	48	52	71
	α_3	10	14	66
Establishing a symmetric key and authentication	α_1	11	15	15
	$\alpha_{2_1}, \alpha_{2_2}$	35	39	54
	α_3	10	14	68
	α_4	10	14	82
Communication	α_1	12	16	16
	α_2	11	15	31
	α_3	10	14	45
	α_4	10	14	59
	α_5	10	14	73
Password and key reset	α_1	30	34	34
	$\alpha_{2_1}, \alpha_{2_2}$	61	65	99
Account deletion	α_1	20	24	24
	$\alpha_{2_1}, \alpha_{2_2}$	16	20	44

Table 2

Summary of minimal and maximal session times

Phase	$T^{\mathrm{ses}}_{\mathrm{min}}$	$T^{\mathrm{ses}}_{\mathrm{max}}$
Creating an account	73	85
Establishing a symmetric key and authentication	66	82
Communication	53	73
Password and key reset	91	99
Account deletion	36	44

The timed analysis showed no attack was found for the assumed time values for all phases. From the generated set of executions, only two types of executions could be executed at the time. The first type was honest executions, which were executed between honest users. The second type was executions with the Intruder as himself, during which the Intruder used his cryptographic objects, so he behaved as a typical network user. The executions during which the Intruder used cryptographic objects intercepted from other network users were impossible to execute. This means that the Intruder will not have enough time to gain appropriate knowledge to execute attacking executions.

6. CONCLUSIONS

This paper introduces a system for interactions in healthcare. The Internet of Things is supported by a novel protocol for secure communication in modern IoT networks for medicine and healthcare. The system with protocol enhances communication safety between interconnected devices, ensuring confidentiality and integrity of data and devices.

We tested the proposed protocol and validated it through formal and automated verification and informal analysis. The protocol fulfils all security goals assumed and verified during formal analysis. Informal analysis showed that the protocol guarantees identity verification, anonymity protection, and the ability to revoke access if necessary. It also protects against man-in-the-middle, modification, replay, and impersonation attacks. In turn, automated verification showed that the Intruder did not have enough time to gain appropriate knowledge to execute attacking executions. So, we did not find an attack upon this protocol.

In future work, we will focus on further tests of our system. We will implement the system and protocol in network environments to verify its correctness and security in actual conditions. Also, we will check and suggest the requirements for the system and devices because the protocol efficiency depends on the devices computational capabilities and the nature of the data. Moreover, the system can be expanded with artificial intelligence methods to support doctors' or coaches' feedback, for example, to present preliminary diagnoses or training plans.

Security solutions for the healthcare IoT sector are very important at present. This is caused by the demand for sensitive data security and the constantly increasing number of cyberattacks. The innovative and secure systems that support communication in IoT networks make the lives of the people using them more convenient and straightforward.

REFERENCES

[1] S. Szymoniak, "Key distribution and authentication protocols in wireless sensor networks: A survey," *ACM Comput. Surv.*, vol. 56, no. 6, p. 144, 2023, doi: 10.1145/3638043.

[2] H. Wu, M. Dyson, and K. Nazarpour, "Arduino-based myoelectric control: Towards longitudinal study of prosthesis use," *Sensors*, vol. 21, no. 3, p. 763, 2021, doi: 10.3390/s21030763.

[3] A. Chen *et al.*, "Machine-learning enabled wireless wearable sensors to study individuality of respiratory behaviors." *Biosens. Bioelectron.*, vol. 173, p. 112799, 2020.

[4] S. Singh, A.S. Nandan, G. Sikka, A. Malik, and A. Vidyarthi, "A secure energy-efficient routing protocol for disease data transmission using iomt," *Comput. Electr. Eng.*, vol. 101, p. 108113, 2022, doi: 10.1016/j.compeleceng.2022.108113.

[5] H. Zhou *et al.*, "Robust and sensitive pressure/strain sensors from solution processable composite hydrogels enhanced by hollow-structured conducting polymers," *Chem. Eng. J.*, vol. 403, p. 126307, 2021, doi: 10.1016/j.cej.2020.126307.

[6] A. Bag and N.-E. Lee, "Recent advancements in development of wearable gas sensors," *Adv. Mater. Technol.*, vol. 6, no. 3, p. 2000883, 2021, doi: 10.1002/admt.202000883.

[7] A. Nait Aicha, G. Englebienne, K.S. Van Schooten, M. Pijnappels, and B. Kröse, "Deep learning to predict falls in older adults based on daily-life trunk accelerometry," *Sensors*, vol. 18, no. 5, p. 1654, 2018, doi: 10.3390/s18051654.

[8] M.R. Alshammari and K. M. Elleithy, "Efficient and secure key distribution protocol for wireless sensor networks," *Sensors*, vol. 18, no. 10, p. 3569, 2018, doi: 10.3390/s18103569.

[9] H. Ye, C.-J. Lee, T.-Y. Wu, X.-D. Yang, B.-Y. Chen, and R.-H. Liang, "Body-centric nfc: Body-centric interaction with nfc devices through near-field enabled clothing," in *Designing Interactive Systems Conference*, 2022, pp. 1626–1639.

[10] T.A. Al-Amiedy, M. Anbar, B. Belaton, A.H.H. Kabla, I.H. Hasbullah, and Z.R. Alashhab, "A systematic literature review on machine and deep learning approaches for detecting attacks in rpl-based 6lowpan of internet of things," *Sensors*, vol. 22, no. 9, p. 3400, 2022, doi: 10.3390/s22093400.

[11] A. Stanforda-Clarka and A. Nipper, "Mqtt: The standard for iot messaging," 2021. [Online]. Available: https://mqtt.org/ (Accessed 2022-05-30).

[12] A. Lacava, V. Zottola, A. Bonaldo, F. Cuomo, and S. Basagni, "Securing bluetooth low energy networking: An overview of security procedures and threats," *Comput. Netw.*, vol. 211, p. 108953, 2022, doi: 10.1016/j.comnet.2022.108953.

[13] D. Upadhyay, J. Manero, M. Zaman, and S. Sampalli, "Intrusion detection in scada based power grids: Recursive feature elimination model with majority vote ensemble algorithm," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 3, pp. 2559–2574, 2021, doi: 10.1109/TNSE.2021.3099371.

[14] S. Szymoniak, "Security protocols analysis including various time parameters," *Math. Biosci. Eng.*, vol. 18, no. 2, pp. 1136–1153, 2021.

[15] D. Galinec, W. Steingartner, and V. Zebic, "Cyber rapid response team: An option within hybrid threats," in *INFORMATICS 2019 - IEEE 15th International Scientific Conference on Informatics, Proceedings*, 2019, pp. 43–49.

[16] W. Steingartner, D. Galinec, and A. Kozina, "Threat defense: Cyber deception approach and education for resilience in hybrid threats model," *Symmetry*, vol. 13, no. 4, p. 597, 2021.

[17] M. Roggenbach, S.A. Shaikh, and H.N. Nguyen, "Formal verification of security protocols," *Formal Methods for Software Engineering: Languages, Methods, Application Domains*, p. 395, 2022.

[18] X.-H. Nguyen and K.-H. Le, "Robust detection of unknown dos/ddos attacks in iot networks using a hybrid learning model," *Internet of Things*, vol. 23, p. 100851, 2023, doi: 10.1016/j.iot.2023.100851.

[19] A.F. Otoom *et al.*, "Deep learning for accurate detection of brute force attacks on iot networks," *Procedia Comput. Sci.*, vol. 220, pp. 291–298, 2023.

[20] B. Zahednejad and C.-Z. Gao, "A secure and efficient ake scheme for iot devices using puf and cancellable biometrics," *Internet Things*, vol. 24, p. 100937, 2023.

[21] M. Burrows, M. Abadi, and R.M. Needham, "A logic of authentication," *Proc. R. Soc. London Ser. A-Math. Phys. Eng. Sci.*, vol. 426, no. 1871, pp. 233–271, 1989.

[22] A. Attir, F. Naït-Abdesselam, and K.M. Faraoun, "Lightweight anonymous and mutual authentication scheme for wireless body area networks," *Computer Networks*, vol. 224, p. 109625, 2023, doi: 10.1016/j.comnet.2023.109625.

[23] J.A.H. Alegria, M.C. Bastarrica, and A. Bergel, "Avispa: a tool for analyzing software process models," *J. Softw. Evol. Process.*, vol. 26, no. 4, pp. 434–450, 2014, doi: 10.1002/smr.1578.

[24] V.O. Nyangaresi, “Privacy preserving three-factor authentication protocol for secure message forwarding in wireless body area networks,” *Ad Hoc Netw.*, vol. 142, p. 103117, 2023, doi: 10.1016/j.adhoc.2023.103117.

[25] M. Abdalla, P.-A. Fouque, and D. Pointcheval, “Password-based authenticated key exchange in the three-party setting,” in *International Workshop on Public Key Cryptography*. Springer, 2005, pp. 65–84.

[26] X. Jia, M. Luo, H. Wang, J. Shen, and D. He, “A blockchain-assisted privacy-aware authentication scheme for internet of medical things,” *IEEE Internet Things J.*, vol. 9, no. 21, pp. 21 838–21 850, 2022.

[27] X. Wang *et al.*, “A new rfid ultra-lightweight authentication protocol for medical privacy protection in smart living,” *Comput. Commun.*, vol. 186, pp. 121–132, 2022.

[28] M. Rasslan, M.M. Nasreldin, and H.K. Aslan, “Ibn sina: A patient privacy-preserving authentication protocol in medical internet of things,” *Comput. Secur.*, vol. 119, p. 102753, 2022.

[29] K. Xue, W. Meng, S. Li, D.S. Wei, H. Zhou, and N. Yu, “A secure and efficient access and handover authentication protocol for internet of things in space information networks,” *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5485–5499, 2019.

[30] M. Masud, G.S. Gaba, P. Kumar, and A. Gurtov, “A user-centric privacy-preserving authentication protocol for iot-ami environments,” *Comput. Commun.*, vol. 196, pp. 45–54, 2022, doi: 10.1016/j.comcom.2022.09.021.

[31] A. Rejeb, K. Rejeb, S.J. Simske, and J.G. Keogh, “Blockchain technology in the smart city: A bibliometric review,” *Qual. Quant.*, vol. 56, no. 5, pp. 2875–2906, 2022.

[32] A. Aljofey, A. Rasool, Q. Jiang, and Q. Qu, “A feature-based robust method for abnormal contracts detection in ethereum blockchain,” *Electronics*, vol. 11, no. 18, p. 2937, 2022.

[33] C.-M. Chen, Z. Chen, S. Kumari, and M.-C. Lin, “Lap-ioht: A lightweight authentication protocol for the internet of health things,” *Sensors*, vol. 22, no. 14, p. 5401, 2022.

[34] W. Steingartner, D. Možnik, and D. Galinec, “Disinformation campaigns and resilience in hybrid threats conceptual model,” in *2022 IEEE 16th International Scientific Conference on Informatics (Informatics)*. IEEE, 2022, pp. 287–292.

[35] M. Bakiri, C. Guyeux, J.-F. Couchot, and A.K. Oudjida, “Survey on hardware implementation of random number generators on fpga: Theory and experimental analyses,” *Comput. Sci. Rev.*, vol. 27, pp. 135–153, 2018.

[36] R.L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[37] M. Nasereddin, A. ALKhamaiseh, M. Qasaimeh, and R. Al-Qassas, “A systematic review of detection and prevention techniques of sql injection attacks,” *Inf. Secur. J.-Glob. Perspect.*, vol. 32, no. 4, pp. 252–265, 2023.

[38] C.N. Siahaan, M. Rufisanto, R. Nolasco, S. Achmad, and C.R.P. Siahaan, “Study of cross-site request forgery on web-based application: Exploitations and preventions,” *Procedia Comput.er Sci.*, vol. 227, pp. 92–100, 2023.

[39] H. Touil, N.E. Akkad, K. Satori, N.F. Soliman, and W. El-Shafai, “Efficient braille transformation for secure password hashing,” *IEEE Access*, vol. 12, pp. 5212–5221, 2024, doi: 10.1109/ACCESS.2024.3349487.