sensors

MDPI

Article

A Novel Key Distribution for Mobile Patient Authentication Inspired by the Federated Learning Concept and Based on the Diffie–Hellman Elliptic Curve

Orieb AbuAlghanam [1,*], Hadeel Alazzam [2], Wesam Almobaideen [1,3], Maha Saadeh [4] and Heba Saadeh [1]

[1] Department of Computer Science, The University of Jordan, Amman 77110, Jordan; wxacad@rit.edu (W.A.); heba.saadeh@ju.edu.jo (H.S.)
[2] Department of Information Technology, Yarmouk University, Irbid 21110, Jordan; hadeel.alazzam@yu.edu.jo
[3] Department of Electrical Engineering and Computing Sciences, Rochester Institute of Technology, Dubai P.O. Box 341055, United Arab Emirates
[4] Department of Computer Engineering and Informatics, Middlesex University Dubai, Dubai P.O. Box 500697, United Arab Emirates; m.saadeh@mdx.ac.ae
* Correspondence: o.abualganam@ju.edu.jo

Abstract: Ensuring secure communication for mobile patients in e-healthcare requires an efficient and robust key distribution mechanism. This study introduces a novel hierarchical key distribution architecture inspired by federated learning (FL), enabling seamless authentication for patients moving across different healthcare centers. Unlike existing approaches, the proposed system allows a central healthcare authority to share global security parameters with subordinate units, which then combine these with their own local parameters to generate and distribute symmetric keys to mobile patients. This FL-inspired method ensures that patients only need to store a single key, significantly reducing storage overhead while maintaining security. The architecture was rigorously evaluated using SPAN-AVISPA for formal security verification and BAN logic for authentication protocol analysis. Performance metrics—including storage, computation, and communication costs—were assessed, demonstrating that the system minimizes the computational load and reduces the number of exchanged messages during authentication compared to traditional methods. By leveraging FL principles, the solution enhances scalability and efficiency, particularly in dynamic healthcare environments where patients frequently switch between facilities. This work bridges a critical gap in e-healthcare security, offering a lightweight, scalable, and secure key distribution framework tailored for mobile patient authentication.

Keywords: authentication; AVISPA; BAN logic; e-healthcare system; federated learning (FL)

Received: 21 February 2025
Revised: 27 March 2025
Accepted: 2 April 2025
Published: 8 April 2025

Citation: AbuAlghanam, O.; Alazzam, H.; Almobaideen, W.; Saadeh, M.; Saadeh, H. A Novel Key Distribution for Mobile Patient Authentication Inspired by the Federated Learning Concept and Based on the Diffie–Hellman Elliptic Curve. *Sensors* **2025**, *25*, 2357. https://doi.org/10.3390/s25082357

1. Introduction

In the age of interconnected healthcare systems, the Internet of Medical Things (IoMT) has emerged as a transformative force, revolutionizing the way healthcare data are collected, analyzed, and used [1]. From wearable devices that monitor vital signs to smart implants that transmit real-time health information, IoMT has ushered in an era of unprecedented medical data generation [2]. However, with this abundance of data comes a pressing need for robust security and authentication mechanisms to ensure the integrity and privacy of patient information.

The benefits of IoMT are indeed profound. It enables remote patient monitoring, facilitating the continuous collection of health data outside traditional clinical settings [3]. This empowers healthcare providers to offer personalized and proactive care, detect health

problems early, and make data-driven decisions, ultimately improving patient outcomes. In addition, IoMT improves the efficiency of medical services, reducing the burden on healthcare facilities by enabling telemedicine, remote consultations, and even the possibility of timely interventions through predictive analytics [4].

At the same time, alongside these remarkable advantages, IoMT introduces a series of complex challenges. Security and privacy vulnerabilities are large as data flows between numerous devices, networks, and cloud platforms [5]. Unauthorized access to medical data poses serious risks, making robust authentication mechanisms paramount. Interoperability issues also arise as various devices and platforms must communicate and share data seamlessly while maintaining data integrity [6].

In this intricate landscape, the need for secure and efficient patient authentication mechanisms becomes increasingly evident. Ensuring that healthcare providers have access to the right patient's data is not just a matter of convenience, but a fundamental requirement to provide safe and effective care [7]. Traditional methods of patient authentication, often relying on static identifiers such as usernames and passwords, are not suited to the dynamic and interconnected world of IoMT. They leave room for vulnerabilities and may hinder the full realization of IoMT's potential [8].

Patient authentication within the IoMT ecosystem is a multifaceted challenge with far-reaching implications. It encompasses the methods and technologies used to verify the identity of patients and healthcare providers who access IoMT devices and the data they generate [9]. The fundamental goals of patient authentication in IoMT are twofold: to ensure the security and integrity of medical data and to protect patient privacy [10].

IoMT introduces unique challenges to patient authentication. The diverse array of devices, ranging from wearable sensors to implantable medical devices, requires authentication methods that can accommodate various form factors and communication protocols. These devices must seamlessly integrate into the broader healthcare infrastructure while ensuring the privacy and security of patient data [11].

Additionally, the real-time nature of IoMT data transmission necessitates authentication mechanisms that can operate swiftly and efficiently, without causing delays in medical data access or decision-making processes. Balancing security, speed, and usability is paramount in the IoMT environment [12].

One critical aspect of securing data transmitted within the IoMT ecosystem is the use of robust encryption techniques [10]. Encryption ensures that data are protected from unauthorized access while in transit. However, effective encryption relies on the secure distribution of encryption keys.

Key distribution in the context of IoMT involves the secure exchange and management of cryptographic keys between devices, users, and healthcare systems. These keys are essential for encrypting and decrypting data as they move between IoMT devices, ensuring that sensitive medical information remains confidential and tamper-proof during transmission.

Establishing a secure and efficient key distribution system is paramount for the overall data security within IoMT [13]. It involves addressing challenges such as key management, key generation, and key revocation, while ensuring that patient authentication seamlessly integrates with the encryption process [14].

Federated Learning (FL) is a collaborative, decentralized, and distributed iterative procedure in which individuals work together to train machine learning models while protecting the confidentiality of their personal data [15]. This approach is exemplified by its implementation in Google Gboard [16]. Rather than transmitting raw data, terminal devices (clients) locally process their information and then forward only the modifications to a central system (server). Subsequently, the server compiles and aggregates the modifications from all clients to generate an updated global model, which is then redistributed to the

clients [17]. This iterative process persists until an optimal global model is achieved allowing for continuous improvement of the global model without exposing sensitive data to a centralized entity, making federated learning a promising technique for training machine learning models in privacy-sensitive scenarios [18]. FL has applications in various domains, such as healthcare, finance, Internet of Things (IoT), and edge computing, where data privacy, network bandwidth limitations, or regulatory constraints make centralized training impractical or undesirable [19,20].

A privacy-preserving domain refers to an area or context where techniques, methods, or systems are employed to protect and maintain the privacy of individuals' sensitive information. In such a domain, measures are taken to ensure that personal data are securely handled and are used in a way that minimizes the risk of unauthorized access, disclosure, or misuse [21,22]. This can be achieved using different approaches such as data anonymization, encryption, differential privacy, access controls, secure multi-party computation (SMPC), and FL [23].

The process of FL passes through different steps as illustrates in Figure 1. It begins with the Initialization Phase, where a central server initializes a global model and distributes it to participating devices or servers. In the subsequent local model training phase, each device trains the model independently using its local data, safeguarding the raw data's privacy [24]. The updated models' parameters are then transmitted back to the central server during the model aggregation phase, where they are combined from all participating devices or servers [25]. Following this, the central server performs a model update, incorporating the aggregated parameters into the global model. The updated global model is subsequently sent back to the participating devices, commencing the next round of local training [26,27]. This iterative process, including local training, model aggregation, and model updates, continues until the desired level of model performance is achieved or convergence is reached [15,27,28].

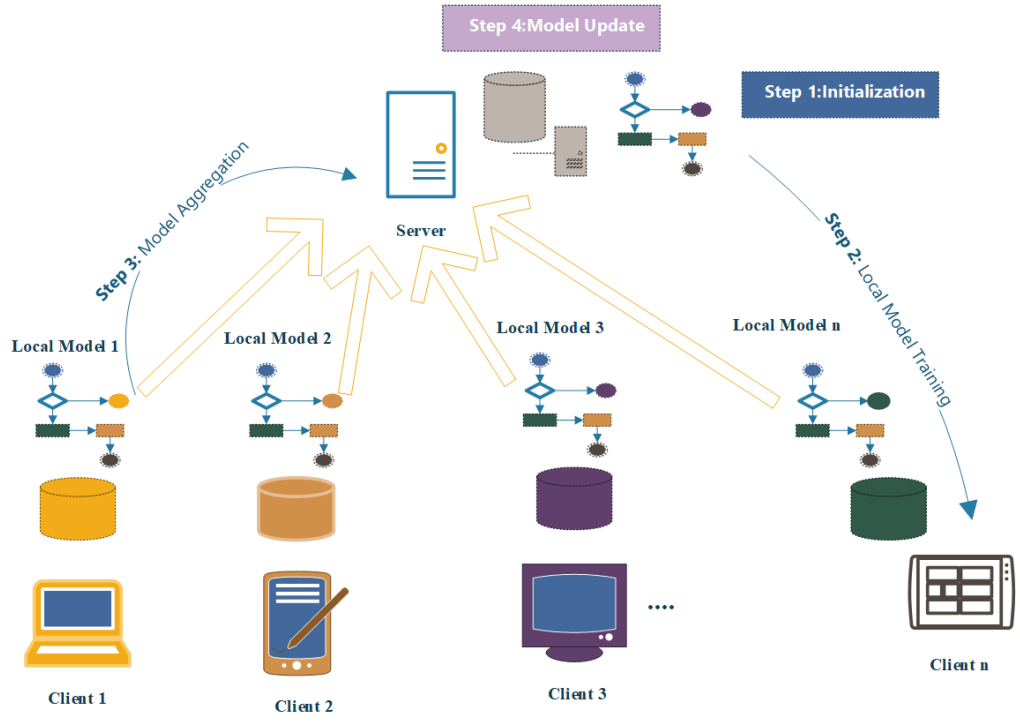

Figure 1. Federated learning steps.

1.1. Challenges to Ensuring Authentication and Confidentiality in Healthcare

Securing healthcare systems is a critical endeavor, but it comes with several challenges such as data privacy, cybersecurity threats, interoperability, mobile and IoT devices, human errors, legacy systems, resource constraints, emerging technologies, patient mobility, and regulatory compliance [29].

Ensuring security in various applications, particularly in healthcare systems, has become a prominent area of research. This entails addressing multiple security facets, including confidentiality, authentication, and privacy. Preserving patient privacy is of the utmost importance. Establishing a secure connection through data encryption between the sender and receiver is crucial. Moreover, accommodating patient mobility is a significant consideration [30,31].

In this paper, we will focus on three challenges: patient mobility, privacy, and resource constraints.

1.2. Contributions

1. Proposing a hierarchical key distribution architecture in the context of an e-healthcare system to enhance the security and privacy of patients based on FL.
2. Proposing a novel key distribution protocol based on FL to exchange the local and global model between the root public key generator $Root_{PKG}$ and sub-public key generator Sub_{PKG}.
3. Designing a lightweight key establishing approach between Sub_{PKG} and mobile patient using Diffie–Hellman elliptic curve algorithm.
4. Providing patient authentication using a private key and several secure parameters also providing identity preservation using the concept of a local model for each Sub_{PKG}.
5. Enhancing the performance for the mobile node in terms of the number of exchange messages [32].

Our work primarily follows a scientific approach, as it focuses on enhancing the security and privacy of patients by proposing a hierarchical key distribution architecture in the context of an e-healthcare system with simulation and formal security proofs.

2. Related Work

This section provides a summary of the authentication techniques discussed in the literature, as illustrated in Table 1. It highlights key methodologies, their effectiveness, and the contexts in which they are applied. Additionally, a comparative analysis is presented to examine the authentication techniques and the environments for which they are designed.

Table 2 provides a comparative analysis of different healthcare systems discussed in the literature. The comparisons have been performed using various architectures, security goals, verification methods, and performance aspects, offering insights into their strengths and limitations.

Many authentication techniques have been used in the literature. For instance, the authors of [33] employ fingerprint recognition as a biometric modality and extract specific features to create a shared cryptographic key. Their proposed methodology includes three phases: System Initialization, Patient Registration, and Mutual Authentication with Session Key Agreement. The System Initialization Phase involves two key steps: extracting a cancelable biometric template and generating system parameters. The authors of [34] propose a lightweight and robust authentication scheme utilizing a simple hash cryptographic function, with public–private key pairs designed specifically for IoMT devices. They utilize formal analysis techniques like BAN logic and ProVerif2.02, in conjunction with informal pragmatic illustration, to validate the effectiveness of their proposed protocol. Moreover,

the study conducts a performance analysis to showcase the delicate balance achieved between security and efficiency, an aspect frequently overlooked in existing solutions.

Another study that uses biometric data for patient authentication is [35]. The authors propose a framework that integrates wearable sensors to monitor vital signs and authenticate patients using biometric data alongside traditional credentials, secured by the SHA-512 algorithm. Sensor data transmission to the cloud is encrypted with the Substitution-Ceaser cipher and improved Elliptical Curve Cryptography (IECC), with enhanced security from an additional secret key. Despite increased complexity, the approach remains computationally efficient, with encryption and decryption times of 1.032 μ and 1.004 μ, respectively, and performance analysis shows strong algorithm reliability compared to RSA and ECC.

The authors of [36] introduce a lightweight anonymous mutual authentication and key agreement scheme for Wireless Body Area Networks (WBANs). This scheme relies solely on hash function operations and XOR operations. The authors employ the automatic security verification tool ProVerif to verify the security properties of their scheme, complemented by informal security analysis. Furthermore, they conduct a comparative analysis of their proposed scheme against several related works. The results demonstrate that their scheme either offers superior advantages in terms of computation cost, energy consumption, and communication cost, or presents lower security risks compared to existing approaches.

Alzahrani et al. [37] present a review of the patient healthcare monitoring and authentication protocol designed for Wireless Body Area Network (WBAN) environments proposed by Xu et al. in [36]. While Xu et al.'s scheme demonstrates efficiency in terms of computation by employing lightweight operations, the conducted analysis uncovers several security loopholes. It is revealed that Xu et al.'s protocol is susceptible to various threats, including replay attacks, key compromise impersonation (KCI) attacks, and privacy concerns. In response to these vulnerabilities, they propose a new authenticated key agreement protocol tailored for WBANs. The security properties of the improved protocol are formally verified and validated through BAN logic analysis and the ProVerif automated simulation tool.

Table 1. Mobile patient authentication in the literature.

Reference	Authentication Technique	Environment
[38]	Biometric+ECC	Mobile healthcare environments
[34]	Shared key	Wireless Medical Sensor Networks
[35]	ECC	IoT
[37]	Improved mutual authentication	Wireless Body Area Networks
[36]	Lightweight anonymous mutual authentication	Wireless Body Area Networks
[39]	Federated learning and blockchain	IoT healthcare
Our proposal	Federated learning and key distribution	IoT healthcare system

The authors of [38] propose a secure and lightweight remote patient authentication scheme tailored for mobile healthcare settings. Their approach translates patient biometric data into Elliptic Curve Cryptography (ECC)-based keys, enabling secure and cost-effective authentication without the need to store or transmit biometric templates. Moreover, the proposed approach provides mutual authentication with session key agreement and resists various types of attacks. Singh et al. in [39] explore the applications of federated learning in establishing a distributed secure environment within smart cities. In addition, they propose a secure architecture for privacy preserving in smart healthcare, utilizing blockchain and FL technologies. Blockchain-based IoT cloud platforms enhance security and privacy, while FL enables scalable machine learning applications, particularly in healthcare. Importantly,

users can access well-trained machine learning models without compromising personal data through federated learning.

Table 2. Different healthcare systems in the literature.

Reference	Architecture	Security Goal	Verification Method	Performance Analysis
[40]	Cloud of things	Prevent Man-in-the-Middle (MITM)	Scyther	Anonymity, Authentication, Authorization, Accountability, Confidentiality, Integrity, Non-repudiation
[41]	IoT-based M-Health system	Signature, Encryption, and Signcryption	Mathematical proof	Computational Cost, Communication Cost
[42]	WBANs	Authentication	Mathematical proof	Storage Overhead, Computation Cost, Communication Cost
[43]	E-healthcare	Authentication	AVISPA and BAN logic	Storage Cost, Communication Cost, Computation Cost
Our proposal	IoT healthcare	Authentication, Confidentiality, and Privacy	AVISPA and BAN logic	Storage Cost, Communication Cost, Computation Cost

3. Proposed System

3.1. Architecture Overview

This section proposes a key distribution architecture inspired by FL, as illustrated in Figure 2. Additionally, Table 3 presents the abbreviations used in this paper. The architecture consists of Root PKG, several Sub-PKGs, and patients. In each Sub-PKG, there is a local model to keep the data for each patient who belongs to this party. In the Root PKG, there is a global model that handle all local models without having knowledge about the patients in each Sub-PKG. In this architecture, several challenges have been addressed, such as computational complexity, communication demands, and storage requirements for mobile patients. Additionally, it prioritizes the secure key distribution process to safeguard patient privacy, even in situations where patients change their positions, all while minimizing unnecessary complexity.

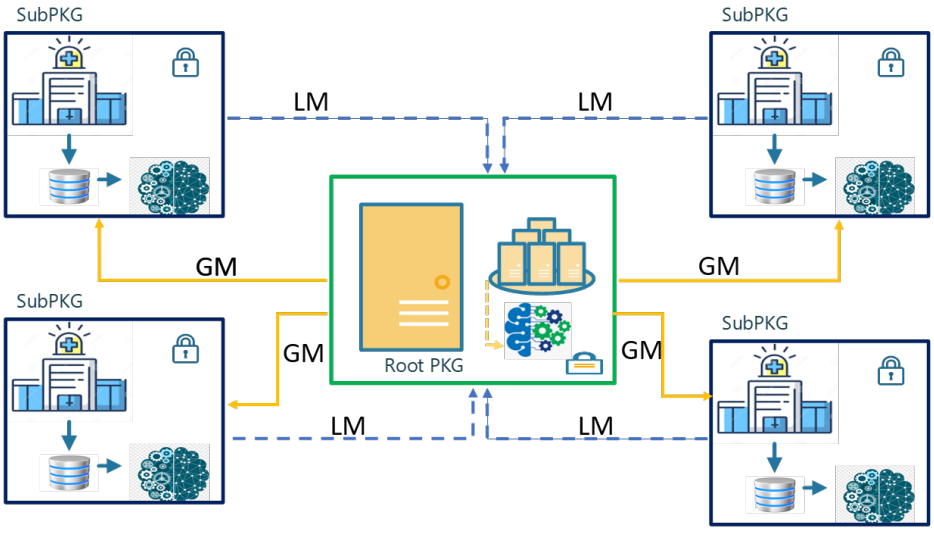

Figure 2. Hierarchy architecture for key distribution using federated learning.

Table 3. List of symbols and abbreviations used in the paper for notation and computation purposes.

Notations	Description
GM	Global model
LM	Local model
PKG	Public key generator
PK_{Root}	Public key for the $Root_{PKG}$
Pr_{Root}	Private key for the $Root_{PKG}$
DC_{Root}	Digital signature for $Root_{PKG}$
DC_{Sub}	Digital signature for Sub_{PKG}
ID	Real identity of the patient
H	Hash function
ϵ	Elliptic curve range
η	Ranges for group signature
γ	Root sign value
Y	Sub-PKG sign value

It can be noticed that the Root PKG has a global model that aims to distribute common parameters for all Sub-PKGs based on the received local models from Sub-PKGs. Moreover, each Sub-PKG authenticates the patient using a shared common key between them. Thus, any mobile patient moving to another Sub-PKG can be authenticated without asking the original Sub-PKG.

In this architecture, we will employ the concept of FL to design a lightweight protocol to provide a secure connection between the patient and the hospital. Thus, the patient can establish a shared key using Elliptic Curve Diffie–Hellman (ECDH) and can be authenticated by the other parts (Sub-PKGs). ECDH is a cryptographic key exchange and authentication protocol commonly used to establish a secure communication channel between two parties over an insecure network [44].

3.2. The Proposed Protocol Overview

The proposed protocol consists of three phases: the first phase, called the Initialization Phase, between $Root_{PKG}$ and Sub_{PKG}. In the second phase, the patient registers for an official Sub_{PKG}. The third phase is the Mobile Patient Authentication Phase, which allows an easy way for any mobile patient node to access any Sub_{PKG} without returning to the old one. On the other hand, further details on Elliptic Curve Diffie–Hellman can be found in Appendix A.1. The following subsections discuss each phase in detail.

3.2.1. Initialization Phase Between $Root_{PKG}$ and Sub_{PKG}

During this phase, the $Root_{PKG}$ initializes and distributes specific set of parameters for each Sub_{PKG} to assist them in creating their local models. After each Sub_{PKG} has finished building its local model, it sends its local model back to the Root PKG. The Root PKG, in turn, utilizes these local models to combine them into the ultimate model, often known as the global model. Figure 3 illustrates the process of the Initialization Phase, which consists of a two-step Initialization Phase and the model aggregation phase.

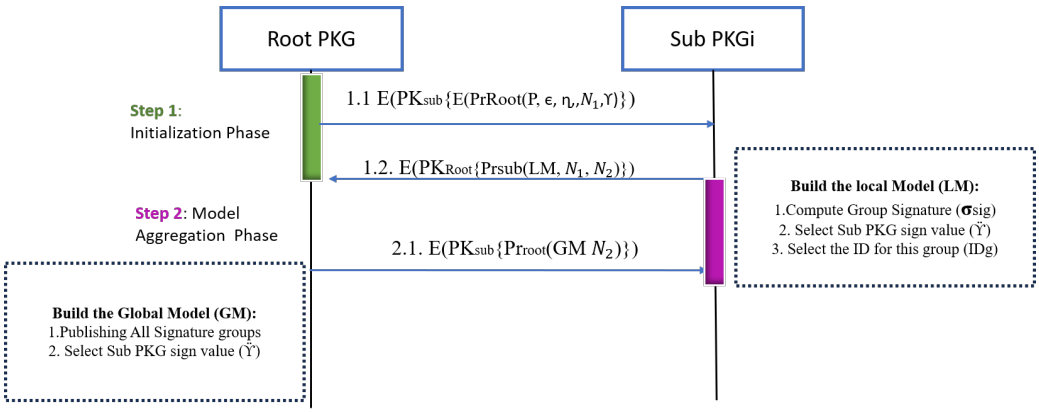

Figure 3. The proposed protocol at Initialization Phase.

Each Sub_{PKG} has its own public key and private key, while the same applies to the $Root_{PKG}$. $Root_{PKG}$ is considered as one trusted point for all other Sub_{PKG}s. Thus, for any further connection between any Sub_{PKG}s, they should exchange their digital signature which is represented in Equation (1):

$$DC(SubPKGi)_{(root,sub_i)} = \left\{ E_{Pr_{root},(PK_{sub_i},ID_{sub},T')} \right. \quad (1)$$

The following steps conclude the details of each connection:

1. $Root_{PKG}$ initiates a request to send information about the whole system, which consists of all Sub-PKGs and patients. This information is essential for the Sub-PKGs to construct their local models while considering these parameters.
2. Root parameters are elliptic curve ranges for each Sub-PKG (ϵ), ranges for group signature (η), and root sign value (γ).
3. The parameters are securely handled. Initially, ensuring nonrejection properties, the $Root_{PKG}$ encrypts these parameters using its private key, to ensure integrity and authentication. Subsequently, it encrypts them once more using the public key provided by the respective Sub-PKG to maintain confidentiality.
4. After each Sub_{PKG} receives the parameters from the root and based on the number of patients that it needs to deal with, the Sub_{PKG} determines the group signature σ_{sig}, Sub-PKG signing value Y, and identity for the group ID_g. After that, the Sub_{PKG} sends a local model via its private key and encapsulates via the root public key.
5. The root aggregates multiple local models from various Sub-PKGs, each equipped with its own set of parameters. This collective information is then used to construct the global model, ensuring a comprehensive consideration of all these individual parameters.

3.2.2. Patient Registration and Key Generation Phase

Figure 4 illustrates this phase; each patient is assigned to their respective Sub_{PKG}. They must agree on various parameters and establish a shared key. This shared key will then be utilized for future connections. The Sub_{PKG} needs to authenticate patients. When the patient asks for the Sub_{PKG} for the first time for registration to be part of a group, the patient must show the Sub_{PKG} acceptable credentials.

The Sub_{PKG} thoroughly reviews it and checks whether anything about the patient is suspicious. If the Sub_{PKG} is fully satisfied with the background verification, it accepts the patient's request to be part of the group. The Sub_{PKG} and the patient create a shared key using **ECDH** as follows:

1. The Sub_{PKG} selects generator G based on the range that was created by the $Root_{PKG}$, prime number p, and selects a private number d_{sub} to determine the pubic key as $Psub = G^{d_{sub}} modp$.
2. The patient selects the private key $d_{patient}$ then determines its public key as $Ppatient = G^{d_{patient}} modp$ and sends the public key to the Sub_{PKG}. Moreover, it determines the shared key as $SK = Psub * d_{patient}$.
3. After the Sub_{PKG} receives the patient's public key it will determine the shared key as $SK = Ppatient * d_{sub}$.

After the shared key has been established between the patient and the Sub_{PKG}, the Sub_{PKG} sends an encrypted message using the shared key that holds the group signature (σ_{GS}), the identity of the group (ID_g), the root sign (γ), and the Sub-PKG sign (Y).

The patient computes their own signature after receiving the parameters from Sub_{PKG} using ECDSA Sign based on the following points:

1. Compute a message digest of the data you want to sign, often using a cryptographic hash function like SHA-256.
2. The patient computes the digest for the message that equals $\sigma_{sig} + \Upsilon + \gamma$; this will be as one block while $\varkappa$ is unique for each patient.

$$Message = \sigma_{sig} + \Upsilon + \gamma + \varkappa \quad (2)$$

3. Generate a random number k in the range [1, n − 1].
4. Compute the point (X_1,Y_1) = k*G.
5. Calculate r = X_1 mod n.
6. Calculate s = $\frac{H(M)+d_{patient}*r}{K}$ mod n.
7. The patient 's signature (P_{sig}) is (r + $\varkappa$,s + $\varkappa$).

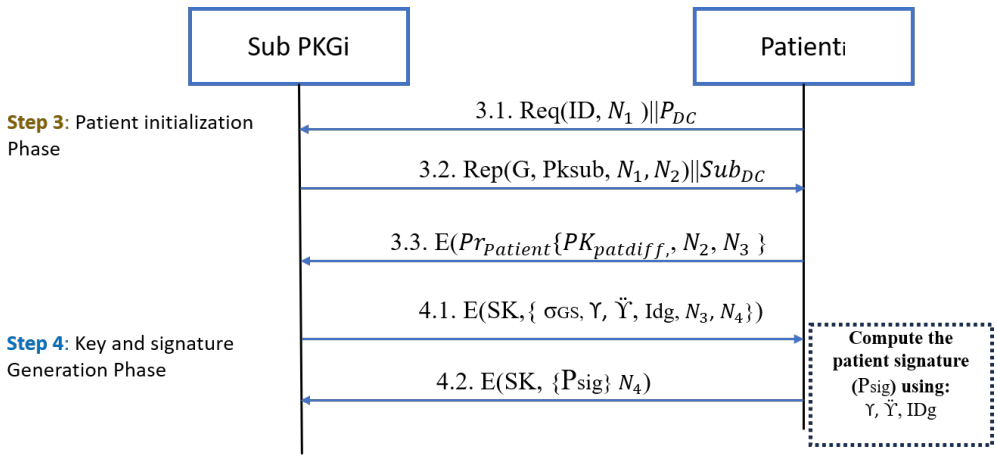

Figure 4. Patient Registration Phase.

3.2.3. Mobile Patient Authentication

Figure 5 illustrates the steps that are needed for any mobile patient that needs to change its location. The bottleneck in [32] is that if any mobile moves to a different Sub-PKG domain, it needs around nine messages to distribute a shared key, and the mobile node should ask the gateway. In this protocol, the patient has a signature and a group identity. Once the patient changes their location and wants to join another Sub-PKG domain, the following procedure is applied.

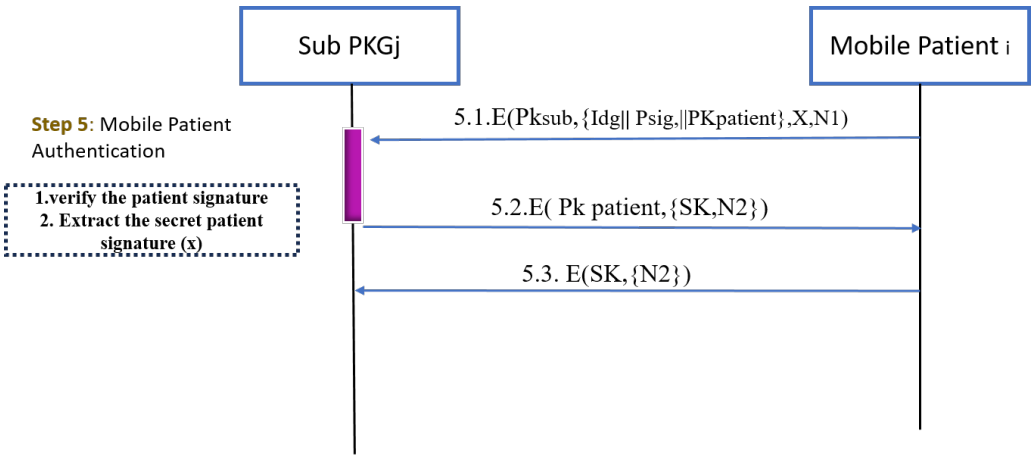

Figure 5. The mobile Patient Authentication Phase.

For verification, the Sub-PKG tries first to extract the patient secret value $\varkappa$ as follows:

1. A mobile patient sends a request for Sub_{PKGj} to obtain authentication and to be allowed to enter this domain.
2. To achieve confidentiality, the mobile patient encrypts its request via Sub_{PKGj}'s public key.
3. The request contains the patient signature, patient public key, group identity, hashed secret value, and nonce.
4. Sub_{PKGj} extracts the $\varkappa$ assuming that the Sub-PKG has initial values for each $\sigma_{sig}, \mathrm{Y}, and \gamma$ that exist in the global model. $x = M - (\sigma_{sig} + \mathrm{Y} + \gamma)$ then checks this value by hashing it then compares it with the received digest. After that, it hashes the message using hash algorithm h = H(M).
5. Sub_{PKGj} find the exact value of (r,s) after subtracting the patient's secret value from each $\varkappa$.
6. Calculate the modular inverse of the signature proof $s1 = s^{(-1)} modn$.
7. Recover the random point used during the signing: R′ = (h ∗ s1) ∗ G + (r ∗ s1) ∗ pubKey.
8. Take from R′ its x-coordinate: r′ = R′.x.
9. Calculate the signature validation result by comparing whether r′ == r.

4. Security Analysis

4.1. Security Proof Using AVISPA Tool

We conduct a comprehensive security verification using Automated Validation of Internet Security Protocols and Applications (AVISPA), which yields outcomes aligned with the protocol's objectives. Initially scripted in CAS+, the protocol is subsequently translated into HLPSL code to scrutinize the integrity and confidentiality of crucial components such as keys, signatures, and confidential messages. As an illustration of our methodology, we instantiate three entities: $Root_{pkg}$, Sub_{pkg}, and the mobile patient. In the first phase, we focus on the authentication between $Root_{pkg}$ and Sub_{pkg} as well as the secrecy of the local model of the Sub_{pkg}. The next two phases in the proposed protocol are the Patient Authentication Phase and the Patient Registration Phase. The results are simulated and indicate that the protocol is safe against attacks, see Appendix A.2.

4.2. BAN Logic

BAN logic [45,46] is used to formally verify the proposed protocol. BAN logic is a logic of authentication proposed by Burrows, Abadi, and Needham [45]. It uses inference rules and it is based on some initial assumptions to infer new facts that can lead to the aims being achieved. BAN logic has been used to analyze the security of authentication

protocols against some of the most common attacks, like the Man-in-the-Middle attack, Intercept-and-Resend attack, and replay attack [47]. The proposed authentication protocol is verified using BAN logic, since it is useful to verify such protocols, which are based on fresh values and trust [48]. For further details regards BAN logic rules and notations, please refer to [45,49]. The full proof and analysis of the BAN logic are included in Appendix A.3.

4.3. Security Analysis Against Well-Known Attack

This section discussed the threat model and how the proposed scheme is secure against the most common attacks.

- **Brute Force Attack:** The attacker cannot reveal the private keys or the symmetric keys in a reasonable time. The EC private key cannot be calculated from the EC public key since it is an Elliptic Curve Discrete Logarithm Problem (ECDLP) [49,50]. The session key SK is generated using the ECDH algorithm which is based on the use of an EC private key for either the patient node or the SubPKG. Breaking ECDH session keys requires solving ECDLP, which is infeasible with classical computers. Consequently, brute force attacks will fail.
- **Man-In-The Middle and Eavesdropping Attack:** The attacker can intercept the communication between the patient's node and the Sub_{PKG} to read the data shared between these two entities; however, the attack will fail. In the Patient Authentication Phase, all messages are encrypted either using ECC to encrypt message 5.1 with the Sub_{PKG} public key, or ECDH to encrypt messages 5.2 and 5.3 with the symmetric key SK. To decrypt the messages, the attacker should have the Sub_{PKG} private key which is known only to the Sub_{PKG} and the symmetric key SK which is known only to the patient node and the Sub_{PKG}. Consequently, this attack will fail.
- **Replay Attack:** The attacker will try to perform a replay attack by resending a valid message to the Sub_{PKG}; however, the Sub_{PKG} can detect this attack. Replay attacks can be detected using nonce values. In the Patient Authentication Phase, whenever the patient node should be authenticated, a nonce N1 is generated at the node side and passed to the Sub_{PKG}. Another nonce N2 is also generated at the Sub_{PKG} side and passed to the node. By verifying the freshness of these nonce values, both the patient's node and the Sub_{PKG} can detect the replay attack. A similar approach is used to detect replay attacks in the Patient Registration Phase; when the signature is calculated, a nonce value is generated by the node and passed to the Sub_{PKG} along with the signature. By verifying the freshness of the nonce value, the Sub_{PKG} will make sure that the signature is newly generated; if not, a replay attack is detected due to nonce verification failure.
- **Signature Forgery Attack:** The attacker tries to impersonate the patient's node by forging the patient's node signature; however, this attack will fail. In order to forge a signature, the attacker needs to know the group signature (σ_{GS}), the identity of the group (ID_g), the root sign (γ), and the Sub-PKG sign (Y). These parameters are shared by the Sub_{PKG} in an encrypted message during the Patient Registration Phase. The message is encrypted using a session key SK, which is known only to the patient node and the Sub_{PKG}; consequently, the attacker will not be able to know the parameters. In addition to these parameters, the attacker should know the patient node's private key, which is known only to the node.
- **Unauthorized Access and Identity Theft:** If the attacker gains access to a patient's signature, they could impersonate a patient and access personal health information. To mitigate this threat, all communication messages between the patient's node and the Sub_{PKG} are encrypted. Consequently, the attacker will not be able to access the patient's signature.

5. Experimental Results and Discussion

In this section, a deeper analysis of the proposed protocol's complexity in terms of storage, computation, and communication cost is presented. Moreover, a comparison with other related works in terms of several performance metrics is conducted.

5.1. Storage Cost

The storage cost refers to the amount of memory required to store the parameters necessary for establishing a shared key for the **mobile patient node**. Furthermore, the incurred cost varies based on the specific type of parameter stored within the mobile node.

The **Total Storage Cost (TSC)** is measured in bits by using the default sizes for each of them. The size of the symmetric key is fixed to 128 bits, and the identifier and the group signatures are saved in 256 bits for the mobile patient node. The public key size is 256 bits assuming that NIST P-256 curve (secp256r1) has been used [51].

In the proposed architecture, we focus on reducing the memory cost of the mobile node in terms of the number of keys and the number of initial parameters that each node should have. The following equation shows the exact complexity for the mobile patient node.

Table 4 presents the size of each parameter that has been used in this paper. The digital signature that has been used in the proposed protocol is based on ECDSA and NIST P-256 curve where r and s have 32 bytes. Therefore, the digital signature's size is 64 bytes, while the public key and the private key have 64 bytes. Unique Identifiers (UIDs: 16 bytes) are used to identify each patient node.

Table 4. The size in byte for each abbreviation in the proposed protocol.

Abbreviation	Size (Byte)
Digital Signature	64
Public Key	64
Private Key	64
Unique Identifiers	16
Shared Key	32
Patient Signature	64
Root Signature (γ)	64
Patient Secret Value ($\varkappa$)	64

Elliptic Curve Diffie–Hellman (ECDH) is used, so the size of the shared key is 32 bytes. The root and patient signature have 64 bytes while there is 64 bytes for the patient's secret value [52–54]. Table 5 presents the storage complexity for each node in our architecture. It can be seen that the main objective of the proposed protocol is to reduce the storage complexity of the patient node due to the fact that each $Root_{PKG}$ and $SubPKG$ are considered powerful devices.

Table 5. The storage complexity for each node in the proposed protocols.

Node Type	Storage	Size in (Bytes)
$Root_{PKG}$	PK_{Root}, Pr_{Root}, DC_{Root}, root parameters ϵ, η, γ, n*Y.	384 + 64 n
Sub_{PKG}	PK_{Sub}, Pr_{Sub}, DC_{Sub}, m*SK, γ, Y	320 + 32 m
$Mobile_{Patient}$	SK, Id_i, P_{sig}, γ, $\varkappa$	240

Table 6 presents a comparative study in terms of the memory storage costs required by the mobile nodes and different related protocols. The authors of [55] propose a new and improved group signature scheme base on federated learning. The main goal of their proposal is to reduce the commutation and computation cost to provide efficient privacy.

In [56], a full dynamic secret sharing is proposed for federated learning. It can be noticed that our approach outperforms the others due to the fact that only one shared key is stored in the patient node. In [57], a Remote Authentication Method (RAM) with Autonomous Shared Keys (ASK) is introduced in the context of medical applications.

Table 6. The total number of shared keys that should be stored in each node.

Scheme	Number of Shared Keys
Our Proposal	1
[57]	1
[55]	2
[32]	3
[56]	2

5.2. Computation Cost

The proposed scheme aims to authenticate the patient's mobile node based on the computed patient signature. The computation cost analysis considers the patient signature generation cost P_{sig}, which is generated once in the Patient Registration Phase and reused in the Patient Authentication Phase, and the patient signature verification cost which is performed by the Sub_{PKG} in the Patient Registration Phase and the Patient Authentication Phase. As discussed in Section 3.2.2, signature generation and verification are based on ECDSA. Table 7 presents the notations used to measure the computation cost.

Table 7. Notation used in computational cost analysis.

Metric	Description
TM	The computing time of the modular multiplication operation.
TA	The computing time of the modular addition operation (negligible).
TIN	The computing time of the modular inversion operation.
TEM	The computing time of the elliptic curve multiplication operation.
TEA	The computing time of the elliptic curve addition operation.

ECDSA Analysis: The cost and efficiency of ECDSA schemes depend on the number of operations used in the methods. Let TM represent the computing time required for the modular multiplication operation, TA the computing time required for the modular addition operation, and TIN the computing time required for the modular inversion operation. TEM represents the computing time required for the elliptic curve multiplication operation. Finally, TEA is the computing time required for the elliptic curve addition operation. According to the literature [58,59], TA is much less than TM and can be negligible. Assuming that TM is the main operation, the computation cost of TIN, TEM, and TEA in terms of modular multiplication operation TM can be calculated as follows: TIN = 11.6 TM, TEM = 29 TM, and TEA = 0.12 TM [58–60].

According to the computation cost required for different operations discussed above, we can analyze the computation cost of the ECDSA signature generation and verification as follows: To generate the signature on the mobile patient node, ECDSA requires one modular multiplication operation, one elliptic curve multiplication operation, and one modular inversion operation which equals TM + TEM + TIN = 41.6 TM. To verify the signature on the Sub_{PKG} side, ECDSA requires two modular multiplication operations, two elliptic curve multiplication operations, one modular inversion operation, and one elliptic curve addition operation which equals 2TM + 2TEM + TIN + TEA = 71.72 TM. Note that, in our calculation, the modular addition operation is ignored as it is negligible [58,59].

Table 8 shows the computation cost compared with other signature-based authentication schemes from the literature. Note that n indicates whenever a node needs to be authenticated. The main advantage of our scheme is that even the signature generation cost is slightly higher than some related schemes; however, it is computed once on the mobile patient's node. On the other hand, in the other schemes, the signature is generated every time a node/client should be authenticated.

Table 8. Computation cost for signature generation and verification.

Schemes	Signing Cost	Verification Cost	Node Cost	Server Cost
Our scheme	TM + TEM + TIN = 41.6 TM	2TM + 2TEM + TIN + TEA = 71.72 TM	41.6 TM (once)	71.72 TM*n
[58]	TM + TEM + TIN = 41.6 TM	2TM + 2TEM + TIN + TEA = 71.72 TM	41.6 TM*n	71.72 TM*n
[60]	TM + TEM = 30 TM	2 TEM + TEA = 58.12 TM	(30 TM + 58.12 TM)*n	(30 TM + 58.12 TM)*n
[59]	TM + 2TEM = 59 TM	2TEM + TEA = 58.12 TM	59 TM*n	58.12 TM*n

5.3. Communication Cost

The communication cost in any network refers to the resources consumed during data transmission between devices or nodes [61]. In this paper, the number of exchanged messages required for any mobile patient node to be authenticated by its associated Sub_{PKG} is analyzed. Figure 6 illustrates the differences in communication costs across various protocols for the mobile patient node in terms of the total number of messages and the number of nodes that should work together in order to distribute the key. It can be noticed that our proposed method demonstrates a lower communication overhead compared to the other related proposals.

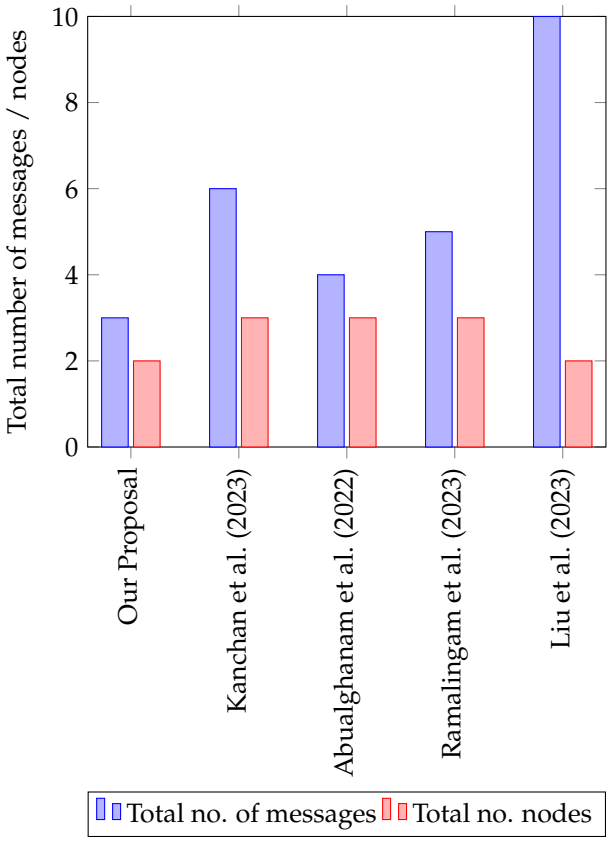

Figure 6. The communication cost of various protocols for the mobile patient node. The references are: Kanchan et al. (2023) [55], Abualghanam et al. (2022) [32], Ramalingam et al. (2023) [57], and Liu et al. (2023) [56].

In [56], the highest number of messages is shown for distributing the key between two nodes. In [32], the proposed approach maintains a moderate balance between the number of messages and nodes, while [57] shows a noticeable increase in total messages, though the number of nodes remains controlled. In contrast, our proposal effectively reduces both message exchanges and node involvement, enhancing efficiency in mobile environments.

6. Conclusions

Maintaining privacy and security is vital in healthcare systems, especially those that support mobile patients who need to be authenticated and protected while on the move. This study investigated the integration of the concept inspired by federated learning and the hierarchical structure of public key generators used in layered healthcare systems to authenticate different mobile patients. Both the BAN logic and the SPAN-AVISPA tool were used to verify the validity of the proposed architecture and the performance of key distribution protocols designed based on it. In addition, the storage requirement and computation and communication costs of the mobile patient nodes were used to assess the performance of the proposed protocol. The effectiveness of the proposed key distribution protocol outperformed other similar protocols in terms of reducing computation and communication costs. In addition, it reduced the number of messages needed for an authentication protocol and the number of shared keys that must be exchanged and stored at each patient's mobile node to only one key.

Author Contributions: O.A. and M.S.: Conceptualization, validation, methodology, writing—original draft preparation, software. H.A., W.A. and H.S.; methodology, validation, review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Appendix A.1. Elliptic Curve Diffie–Hellman

Elliptic Curve Diffie–Hellman (ECDH) is a cryptographic key exchange protocol that is widely used to establish a secure communication channel between two parties over an insecure network [44]. It is an extension of the original Diffie–Hellman key exchange protocol, leveraging the mathematical properties of elliptic curves to provide strong security with relatively small key sizes and efficient computations. ECDH is particularly important in modern cryptography as it forms the basis for secure communication in various applications, including secure messaging, secure web browsing (HTTPS), and more.

The elliptic curve process can be summarized by the following [62]:

- Elliptic Curves: At the heart of ECDH is the use of elliptic curves over finite fields. An elliptic curve is a mathematical structure defined by Equation (A1).

$$y^2 = x^3 + ax + b \tag{A1}$$

  where a and b are constants. The curve is defined over a finite field, which means all calculations are performed modulo a prime number p.
- Key Generation: Each party involved in the key exchange process generates its own elliptic curve key pair. This consists of a private key (a randomly chosen number) and

a corresponding public key (calculated by multiplying the base point of the curve by the private key).

- Key Exchange: When two parties want to establish a shared secret key, they exchange their public keys over an insecure communication channel.
- Shared Secret Calculation: Each party uses their private key and the received public key to independently compute a shared secret point on the elliptic curve. The magic of elliptic curve mathematics ensures that these independently calculated shared secrets are equal.
- Shared Secret Derivation: The shared secret point is then used as an input to a key derivation function (KDF) to produce a shared secret key. This shared secret can be used for the symmetric encryption and decryption of data between the two parties.

Elliptic Curve Diffie–Hellman (ECDH) offers several distinct advantages when compared to the traditional Diffie–Hellman key exchange. Firstly, ECDH provides robust security despite employing relatively compact key sizes, resulting in efficient computations and reduced bandwidth usage [63]. Moreover, it boasts resilience against quantum attacks, positioning it as a promising candidate for post-quantum cryptography. This protocol excels in terms of efficiency due to the superior performance of elliptic curve operations compared to the conventional modular exponentiation operations used in standard Diffie–Hellman. The compact nature of ECDH keys, considerably shorter than their RSA or DSA counterparts of equivalent strength, renders them ideal for resource-constrained devices and systems. Lastly, ECDH enjoys widespread adoption across modern cryptography standards and protocols, including TLS for secure web browsing, PGP for secure email communication, and numerous others, affirming its significance in ensuring secure data exchange over insecure networks.

Appendix A.2. Simulation Code of the Security Proof

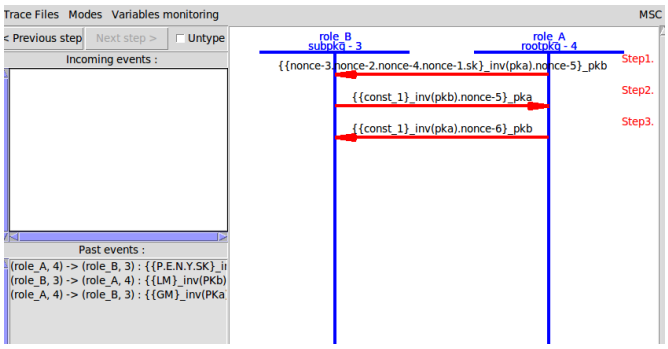

Figure A1. The simulation for the Initialization Phase between $RootPKG$ and Sub_{PKG}.

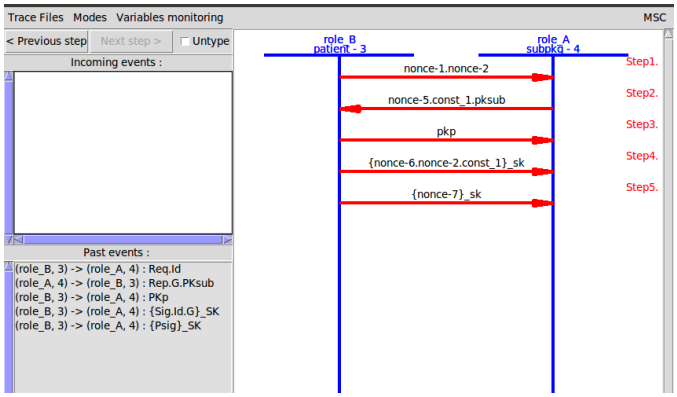

Figure A2. The simulation for Patient Registration Phase between Sub_{PKG} and patient.

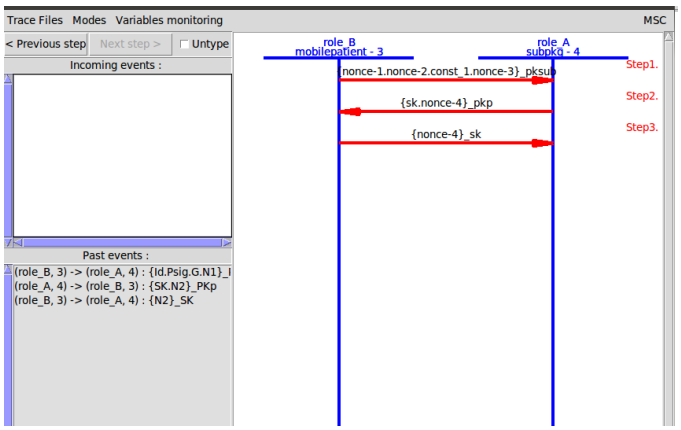

Figure A3. The simulation for Mobile Patient Authentication Phase.

```
SUMMARY
SAFE

DETAILS
BOUNDED_NUMBER_OF_SESSIONS
TYPED_MODEL

PROTOCOL
/home/span/span/testsuite/results/hlpsl/GenFile.if

GOAL
As Specified

BACKEND
CL-AtSe

STATISTICS
Analysed  : 698 states
Reachable : 536 states
Translation: 0.00 seconds
Computation: 0.00 seconds
```

Figure A4. The result for the Initialization Phase between $Root_{PKG}$ and Sub_{PKG} Protocol.

```
SUMMARY
SAFE

DETAILS
BOUNDED_NUMBER_OF_SESSIONS
TYPED_MODEL

PROTOCOL
/home/span/span/testsuite/results/hlpsl/GenFile.if

GOAL
As Specified

BACKEND
CL-AtSe

STATISTICS
Analysed  : 432 states
Reachable : 387 states
Translation: 0.00 seconds
Computation: 0.00 seconds
```

Figure A5. The result for the Patient Registration Phase between Sub_{PKG} and patient Protocol.

```
SUMMARY
SAFE

DETAILS
BOUNDED_NUMBER_OF_SESSIONS
TYPED_MODEL

PROTOCOL
/home/span/span/testsuite/results/hlpsl/GenFile.if

GOAL
As Specified

BACKEND
CL-AtSe

STATISTICS
Analysed  : 278 states
Reachable : 124 states
Translation: 0.00 seconds
Computation: 0.00 seconds
```

Figure A6. The result for Mobile Patient Authentication Phase Protocol.

Appendix A.3. BAN Logic

BAN Logic Notations that are used in this study:

1. P$|\equiv$ X: P believes X. This means that P considers X to be true and acts accordingly.
2. P $\triangleleft$ X: P sees X, i.e., when P receive a message contains X, then P sees X.
3. P $|\sim$ X: P said X.
4. P $\|\sim$: P recently said X.
5. P$\Rightarrow$ X: P has jurisdiction over X or P controls X.
6. #(X): The formula X is fresh. This means that X has not been sent in a message at any time before the current run of the protocol.
7. xk: This means that formula X is encrypted using the key k.
8. x k^{-1}: This represents that formula X is encrypted using the inverse key of k, i.e., if k is a public key, then k^{-1} is the private key that corresponds to k.
9. PK (k, P): k is a public key for P and there exists a unique key that corresponds to k.
10. II(P): P has a private key that is known only to P.
11. σ(X, P): X is signed by P's private key.
12. A $\longleftrightarrow^{k}$ B , K is a shared key between only A and B.

Initialization Phase: Assume that Sub_i is Sub_{PKGi}, $Root_i$ is Root PKGi, PK$Root_i$ is the public key of $Root_i$, PKSub_i is the public key of the Sub_i, Pr$Root_i$ is the private key of $Root_i$, PrSub_i is the private key of Sub_i, M_1 is the first message between $Root_i$ and Sub_i, M_2 is the second message sent from Sub_i to $Root_i$, and M_3 is the third message sent from $Root_i$ to Sub_i.

Idealized messages: The following are the idealized messages for this phase based on Figure 3.

- M_1: $Root_i \rightarrow Sub_i$: {{P, N_1, ϵ, η,γ} Pr_{Root_i}} PK_{Sub_i}.
- M_2: $Sub_i \rightarrow Root_i$:{{LM, N_1, N_2} Pr_{Sub_i}} PK_{Root_i}.
- M_3: $Root_i \rightarrow Sub_i$ {{GM, N_2,N_3} Pr_{Root_i}}PK_{Sub_i}.

Assumptions:

- A1: $\text{Root}_i \mid \equiv \text{PK}(\text{Sub}_i, \text{PK}_{\text{Sub}_i})$ means that Root PKG Root_i believes that PK_{Sub_i} is the public key of the Sub-PKG_i Sub_i.
- A2: $\text{Root}_i \mid \equiv II(\text{Sub}_i)$ means that Root PKG Root_i believes that the Sub_i has a private key that corresponds to its public key.
- A3: $\text{Sub}_i \mid \equiv \text{PK}(\text{Root}_i, \text{PK}_{\text{Root}_i})$ means that Sub_i believes that $\text{PK}_{\text{Root}_i}$ is the public key of Root_i.
- A4: $\text{Sub}_i \mid \equiv II(\text{Root}_i)$ means that Sub_i believes that Root_i has a private key that corresponds to its public key.
- A5: $\text{Root}_i \mid \equiv \text{Sub}_i \Rightarrow (M_2, \text{LM})$ means that Root PKG Root_i believes that Sub_i controls the generation of message M_2 and local model LM.
- A6: $\text{Sub}_i \mid \equiv \text{Root}_i \Rightarrow (M_1, P)$ means that Sub_i believes that Root_i controls the generation of message M_1 and the parameters P.
- A7: $\text{Sub}_i \mid \equiv \#(M_1)'$ means that Sub_i believes that part of message M_1 is fresh and has not been sent previously.
- A8: $\text{Root}_i \mid \equiv \#(M_2)'$ means that Root PKG Root_i believes that part of message M_2 is fresh and has not been sent previously.
- A9: $\text{Sub}_i \mid \equiv \#(M_3)'$ means that Sub_i believes that part of message M_3 is fresh and has not been sent previously.
- A10: $\text{Sub}_i \mid \equiv \text{Root}_i \Rightarrow (M_3, \text{GM})$ means that Sub_i believes that Root_i controls the generation of message M_3 and the global model GM.

Goals: The goal of the Initialization Phase is to guarantee the mutual authentication between Root_i and Sub_i. Moreover, Root_i must trust the LM and believe that it is true, and Sub_i must trust the GM and believe that it is true. The following are the goals to be achieved in this phase:

- G1: $\text{Sub}_i \mid \equiv \text{Root}_i \mid \sim M_1$ means that Sub_i should believe that Root_i has said message M_1. So, Sub_i authenticates Root_i.
- G2: $\text{Sub}_i \mid \equiv M_1$ means that Sub_i should believe message M_1, which includes the parameters, and consider it true.
- G3: $\text{Sub}_i \mid \equiv \#M_1$ means that Sub_i should believe that message M_1 is fresh and not sent before.
- G4: $\text{Root}_i \mid \equiv \text{Sub}_i \mid \sim M_2$ means that the Root PKG Root_i should believe that Sub_i has said message M_2, which includes the local model LM. So, Root_i authenticates the Sub_i.
- G5: $\text{Root}_i \mid \equiv M_2$ means that the Root PKG Root_i should believe that message M_2, which includes the LM is true.
- G6: $\text{Root}_i \mid \equiv \#M_2$ means that the Root PKG Root_i should believe that message M_2 is fresh and not sent before.
- G7: $\text{Sub}_i \mid \equiv \text{Root}_i \mid \sim M_3$ means that Sub_i should believe that Root_i has said message M_3.
- G8: $\text{Sub}_i \mid \equiv M_3$ means that Sub_i should believe message M_3, which includes the global model GM and consider it true.
- G9: $\text{Sub}_i \mid \equiv \#M_3$ means that Sub_i should believe that message M_3 is fresh and not sent before.

Analysis:

Goals G1, G2, and G3: G1 is achieved via applying the following signing rule on A3, A4, and M_1. That is, if Sub_i receives the signed message M_1 that is signed using $\text{PK}^{-1}_{\text{Root}_i}$ from Root_i, and Sub_i believes that $\text{PK}_{\text{Root}_i}$ is the public key of Root_i and that Root_i has a private key that corresponds to $\text{PK}_{\text{Root}_i}$ which is $\text{PK}^{-1}_{\text{Root}_i}$, then Sub_i should believe that Root_i said message M_1, which includes the parameters P, and Sub_i authenticates Root_i.

$$\begin{aligned}&\big(\mathrm{Sub}_i \mid \equiv\ \mathrm{PK}(\mathrm{Root}_i, \mathrm{PK}_{\mathrm{Root}_i}),\ \mathrm{Sub}_i \mid \equiv II(\mathrm{Root}_i),\\ &\quad \mathrm{Sub}_i \mid \rhd\ \sigma(M_1, \mathrm{Root}_i))/(\mathrm{Sub}_i \mid \equiv\ \mathrm{Root}_i \mid \sim M_1),\ G1 \text{ is achieved.}\end{aligned}$$

G3 is achieved by applying the below freshness rule to A7. That is, if Sub_i believes that part of message M_1 is fresh (which is N_1), then message M_1 is fresh.

$$\big(\mathrm{Sub}_i \mid \equiv\ \#(M_1)'\big)/(\mathrm{Sub}_i \mid \equiv\ \#M_1),\ \text{so G3 is achieved.}$$

G2 is achieved if the following rules are applied. From G1 and G3, we can infer that Sub_i believes that Root_i believes message M_1; then, the control rule is applied to A6. Specifically, if Sub_i believes that Root_i controls message M_1, and Sub_i believes that Root_i believes message M_1, then Sub_i should believe that message M_1 is true.

$$(\mathrm{Sub}_i \mid \equiv\ \mathrm{Root}_i \mid \sim M_1,\ \mathrm{Sub}_i \mid \equiv\ \#M_1)/(\mathrm{Sub}_i \mid \equiv\ \mathrm{Root}_i \mid \equiv M_1),$$

and

$$(\mathrm{Sub}_i \mid \equiv\ \mathrm{Root}_i\ \Rightarrow (M_1, P),\ \mathrm{Sub}_i \mid \equiv\ \mathrm{Root}_i \mid \equiv M_1)/(\mathrm{Sub}_i \mid \equiv M_1),\ \text{G2 is achieved.}$$

Goals G4, G5, and G6:

G4 is achieved by applying the below signing rule on A1, A2, and M_2. That is, if Root_i receives the signed message M_2 that is signed using $\mathrm{PK}_{\mathrm{Sub}_i}^{-1}$ from Sub_i, and Root_i believes that $\mathrm{PK}_{\mathrm{Sub}_i}$ is the public key of Sub_i and that Sub_i has a private key that corresponds to $\mathrm{PK}_{\mathrm{Sub}_i}$ which is $\mathrm{PK}_{\mathrm{Sub}_i}^{-1}$, then Root_i should believe that Sub_i said message M_2, which includes the local model LM, and Root_i authenticates Sub_i.

$$\begin{aligned}&\big(\mathrm{Root}_i \mid \equiv\ \mathrm{PK}(\mathrm{Sub}_i, \mathrm{PK}_{\mathrm{Sub}_i}),\ \mathrm{Root}_i \mid \equiv II(\mathrm{Sub}_i),\\ &\quad \mathrm{Root}_i \mid \rhd\ \sigma(M_2, \mathrm{Sub}_i))/(\mathrm{Root}_i \mid \equiv\ \mathrm{Sub}_i \mid \sim M_2),\\ &\quad G4 \text{ is achieved.}\end{aligned}$$

G6 is achieved if the below freshness rule on A8 is applied. That is, if Root_i believes that part of message M_2 is fresh (which is N_2), then message M_2 is fresh.

$$\big(\mathrm{Root}_i \mid \equiv\ \#(M_2)'\big)/(\mathrm{Root}_i \mid \equiv\ \#M_2),\ \text{so G6 is achieved.}$$

G5 is achieved if the following rules are applied. From G4 and G6, we can infer that Root_i believes that Sub_i believes message M_2; then, the control rule is applied to A5. Specifically, if Root_i believes that Sub_i controls message M_2, and Root_i believes that Sub_i believes message M_2, then Root_i should believe that message M_2 is true.

$$(\mathrm{Root}_i \mid \equiv\ \mathrm{Sub}_i \mid \sim M_2,\ \mathrm{Root}_i \mid \equiv\ \#M_2)/(\mathrm{Root}_i \mid \equiv\ \mathrm{Sub}_i \mid \equiv M_2),$$

and

$$\begin{aligned}&(\mathrm{Root}_i \mid \equiv\ \mathrm{Sub}_i\ \Rightarrow (M_2, \mathrm{LM}),\\ &\quad \mathrm{Root}_i \mid \equiv\ \mathrm{Sub}_i \mid \equiv M_2)/(\mathrm{Root}_i \mid \equiv M_2),\ \text{G5 is achieved.}\end{aligned}$$

Goals G7, G8, and G9:

G7 is achieved by applying the following signing rule on A3, A4, and M_3. That is, if Sub_i receives the signed message M_3 that is signed using $\text{PK}_{\text{Root}_i}^{-1}$ from Root_i, and Sub_i believes that $\text{PK}_{\text{Root}_i}$ is the public key of Root_i and that Root_i has a private key that corresponds to $\text{PK}_{\text{Root}_i}$ which is $\text{PK}_{\text{Root}_i}^{-1}$, then Sub_i should believe that Root_i said message M_3, which includes the global model GM.

$$\begin{aligned}&\left(\text{Sub}_i \mid \equiv \text{PK}(\text{Root}_i, \text{PK}_{\text{Root}_i}),\ \text{Sub}_i \mid \equiv II(\text{Root}_i),\right.\\&\left.\text{Sub}_i \triangleright \sigma(M_3, \text{Root}_i)\right) / (\text{Sub}_i \mid \equiv \text{Root}_i \mid \sim M_3),\\&G7 \text{ is achieved.}\end{aligned}$$

By applying the following freshness rule to A9, G9 is achieved. That is, if Sub_i believes that part of message M_3 is fresh (which is N_3), then message M_3 is fresh.

$$\left(\text{Sub}_i \mid \equiv \#(M_3)'\right) / (\text{Sub}_i \mid \equiv \#M_3), \text{ so G9 is achieved.}$$

G8 is achieved via applying the following rules. From G7 and G9, we can infer that Sub_i believes that Root_i believes message M_3; then, the control rule is applied to A10. Specifically, if Sub_i believes that Root_i controls message M_3, and Sub_i believes that Root_i believes message M_3, then Sub_i should believe that message M_3 is true.

$$(\text{Sub}_i \mid \equiv \text{Root}_i \mid \sim M_3,\ \text{Sub}_i \mid \equiv \#M_3) / (\text{Sub}_i \mid \equiv \text{Root}_i \mid \equiv M_3),$$

and

$$\begin{aligned}&(\text{Sub}_i \mid \equiv \text{Root}_i \Rightarrow (M_3, \text{GM}),\ \text{Sub}_i \mid \equiv \text{Root}_i \mid \equiv M_3) /\\&(\text{Sub}_i \mid \equiv M_3),\ \text{G8 is achieved.}\end{aligned}$$

Patient Registration Phase

In this section, assume the following:

- Sub_i is Sub-PKGi
- PK_{Sub_i} is the public key of Sub_i
- Pr_{Sub_i} is the private key of Sub_i
- $\text{PK}_{\text{Patient}_i}$ is the public key of Patient_i
- $\text{Pr}_{\text{Patient}_i}$ is the private key of Patient_i
- SK_{Sub_i} is the session key between Sub_i and Patient_i
- $\text{PKDH}_{\text{Patient}_i}$ is the public component of Patient_i
- $\text{PKDH}_{\text{Sub}_i}$ is the public component of Sub_i
- DC_{Sub_i} is the digital certificate for Sub_i
- $\text{DC}_{\text{Patient}_i}$ is the digital certificate of Patient_i
- $M1$ is the first message in this phase between Patient_i and Sub_i
- $M2$ is the second message in this phase sent from Sub_i to Patient_i
- $M3$ is the third message in this phase sent from Patient_i to Sub_i
- $M4$ is the fourth message in this phase sent from Sub_i to Patient_i
- $M5$ is the fifth message sent from Patient_i to Sub_i

Idealized messages:

The following are the idealized messages for this phase based on Figure 4:

- $M1$: $\text{Patient}_i \rightarrow \text{Sub}_i$: ID $||$ N1
- $M2$: $\text{Sub}_i \rightarrow \text{Patient}_i$: $\text{DC}_{\text{Sub}_i} || \{\text{G}, \text{N1}, \text{N2}, \text{PKDH}_{\text{Sub}_i}\}\text{Pr}_{\text{Sub}_i}$
- $M3$: $\text{Patient}_i \rightarrow \text{Sub}_i$: $\text{DC}_{\text{Patient}_i} || \{\text{PKDH}_{\text{Patient}_i}, \text{N2}\}\text{Pr}_{\text{Patient}_i}$
- $M4$: $\text{Sub}_i \rightarrow \text{Patient}_i$: $\{\sigma_{\text{GS}}, \text{Y}, \gamma, \text{Idg}, \text{N3}\}\text{SK}_{\text{Sub}_i}$
- $M5$: $\text{Patient}_i \rightarrow \text{Sub}_i$: $\{\text{Psig}, \text{N3}\}\text{SK}_{\text{Sub}_i}$

Assumptions:

- A1: $\text{Sub}_i| \equiv \text{Sub}_i\Box(\leftrightarrow (\text{SK}_{\text{Sub}_i}))\text{Patient}_i$ means that Sub_i believes that SK_{Sub_i} is a key shared between Patient_i and Sub_i
- A2: $\text{Patient}_i| \equiv \text{Sub}_i\Box(\leftrightarrow (\text{SK}_{\text{Sub}_i}))\text{Patient}_i$ means that Patient_i believes that SK_{Sub_i} is a key shared between Patient_i and Sub_i

Goals:

This phase aims to achieve mutual authentication between the Sub-PKG and the patient, and to generate the session key shared between them. Therefore, both Patient_i and Sub_i should believe and trust that the session key is true. Note that public Diffie–Hellman components $\text{PKDH}_{\text{Patient}_i}$ and $\text{PKDH}_{\text{Sub}_i}$ are shared in signed messages ($M2$ and $M3$) which means the authenticity of these components can be proved in the same way as performed in the previous phase. In this section, we will focus on the trust of the generated session key. The following are the goals to be achieved:

- G1: $\text{Patient}_i| \equiv \text{Sub}_i| \sim M_4$ means that Patient_i should believe that Sub_i has sent message M_4, which includes N3. So, Patient_i authenticates Sub_i.
- G2: $\text{Sub}_i| \equiv \text{Patient}_i| \sim M_5$ means that Sub_i should believe that Patient_i has sent message M_5 which includes the same nonce N3. So, Sub_i authenticates Patient_i.

Analysis:

G1 is achieved via applying the below symmetric rule on A2 and M_4. That is, if Patient_i receives message M_4 that is encrypted using SK_{Sub_i} from Sub_i, and Patient_i believes that SK_{Sub_i} is the shared key between Patient_i and Sub_i, then Patient_i should believe that Sub_i sent message M_4 and so Sub_i is authenticated.

$$\begin{aligned}&\left(\text{Patient}_i \vdash \{M_4\}\text{SK}_{\text{Sub}_i},\quad \text{Patient}_i| \equiv \text{Sub}_i\Box(\leftrightarrow (\text{SK}_{\text{Sub}_i}))\text{Patient}_i\right)\\ &/(\text{Patient}_i| \equiv \text{Sub}_i| \sim M_4),\quad G1 \text{ is achieved.}\end{aligned}$$

Similarly, G2 is achieved if the below symmetric rule is applied to A1 and M_5. That is, if Sub_i receives message M_5 that is encrypted using SK_{Sub_i} from Patient_i, and Sub_i believes that SK_{Sub_i} is the shared key between Patient_i and Sub_i, then Sub_i should believe that Patient_i sent message M_5 and so Patient_i is authenticated.

$$\begin{aligned}&\left(\text{Sub}_i \vdash \{M_5\}\text{SK}_{\text{Sub}_i},\quad \text{Sub}_i| \equiv \text{Sub}_i\Box(\leftrightarrow (\text{SK}_{\text{Sub}_i}))\text{Patient}_i\right)\\ &/(\text{Sub}_i| \equiv \text{Patient}_i| \sim M_5),\quad G2 \text{ is achieved.}\end{aligned}$$

Patient Authentication Phase

Assume that Sub_i is Sub-PKG$_i$, PK_{Sub_i} is the public key of Sub_i, Pr_{Sub_i} is the private key of Sub_i, $\text{PK}_{\text{Patient}_i}$ is the public key of Patient_i, $\text{Pr}_{\text{Patient}_i}$ is the private key of Patient_i, and SK is the session key between Sub_i and Patient_i.

DC_{Sub_i} is the digital certificate for Sub_i, $\text{DC}_{\text{Patient}_i}$ is the digital certificate for Patient_i, M_1 is the first message between Patient_i and Sub_i, M_2 is the second message sent from Sub_i to Patient_i, and M_3 is the third message in this phase sent from Patient_i to Sub_i.

Idealized messages: The below are the idealized messages based on Figure 4.

- M_1:
$$\text{Patient}_i \rightarrow \text{Sub}_i : \{\text{DC}_{\text{Patient}_i}||\{\text{Idg}, \text{Psig}, X, N1\}\text{Pr}_{\text{Patient}_i}\}\text{PK}_{\text{Sub}_i}$$
- M_2: $\text{Sub}_i \rightarrow \text{Patient}_i$: $\{\{\text{SK}, N1, N2\}\text{Pr}_{\text{Sub}_i}\}\text{PK}_{\text{Patient}_i}$
- M_3: $\text{Patient}_i \rightarrow \text{Sub}_i$: $\{N2\}\text{SK}$

Assumptions:

- A1: $\text{Patient}_i| \equiv \text{PK}(\text{Sub}_i, \text{PK}_{\text{Sub}_i})$ means that Patient_i believes that PK_{Sub_i} is the public key of Sub_i.
- A2: $\text{Patient}_i| \equiv II(\text{Sub}_i)$ means that Patient_i believes that Sub_i has a private key that corresponds to its public key.
- A3: $\text{Sub}_i| \equiv \text{PK}(\text{Patient}_i, \text{PK}_{\text{Patient}_i})$ means that Sub_i believes that $\text{PK}_{\text{Patient}_i}$ is the public key of Patient_i.
- A4: $\text{Sub}_i| \equiv II(\text{Patient}_i)$ means that Sub_i believes that Patient_i has a private key that corresponds to its public key.
- A5: $\text{Sub}_i| \equiv \#(M_1)'$ means that Sub_i believes that part of message M_1 is fresh and has not been sent previously.
- A6: $\text{Patient}_i| \equiv \#(M_2)'$ means that Patient_i believes that part of message M_2 is fresh and has not been sent previously.
- A7: $\text{Sub}_i| \equiv \text{Patient}_i \Rightarrow (M_1, \text{Psig})$ means that Sub_i believes that Patient_i controls the generation of message M_1 which includes Psig.
- A8: $\text{Patient}_i| \equiv \text{Sub}_i \Rightarrow (M_2, \text{SK})$ means that Patient_i believes that Sub_i controls the generation of message M_2 and the session key SK.
- A9: $\text{Sub}_i| \equiv \text{Sub}_i \Box(\leftrightarrow (\text{SK}))\text{Patient}_i$ means that Sub_i believes that SK is a key shared between Patient_i and Sub_i.
- A10: $\text{Patient}_i| \equiv \text{Sub}_i \Box(\leftrightarrow (\text{SK}))\text{Patient}_i$ means that Patient_i believes that SK is a key shared between Patient_i and Sub_i.

Goals: The goal of this phase is to authenticate the patient and to generate the session key SK shared between them. Therefore, both Patient_i and Sub_i should believe and trust that the session key is true. The following are the goals to be achieved in this phase.

- G1: $\text{Sub}_i| \equiv \text{Patient}_i| \sim M_1$ means that Sub_i should believe that Patient_i has sent message M_1, which includes the patient's signature Psig.
- G2: $\text{Sub}_i| \equiv M_1$ means that Sub_i should believe message M_1.
- G3: $\text{Sub}_i| \equiv \#M_1$ means that Sub_i should believe that message M_1 is fresh and has not been sent before.
- G4: $\text{Patient}_i| \equiv \text{Sub}_i| \sim M_2$ means that Patient_i should believe that Sub_i has sent message M_2, which includes the session key SK.
- G5: $\text{Patient}_i| \equiv M_2$ means that Patient_i should believe message M_2 which includes the session key SK.
- G6: $\text{Patient}_i| \equiv \#M_2$ means that Patient_i should believe that message M_2, which includes the session key SK, is fresh and has not been sent before.
- G7: $\text{Sub}_i| \equiv \text{Patient}_i| \sim M_3$ means that Sub_i should believe that Patient_i has sent message M_3, which includes the same nonce $N2$. So, Sub_i authenticates Patient_i.

Analysis:

Goals G1, G2, and G3: G1 is achieved via applying the below signing rule to A3, A4, and M_1. That is, if Sub_i receives the signed message M_1 that is signed using $\text{PK}_{\text{Patient}_i}$ from Patient_i, and Sub_i believes that $\text{PK}_{\text{Patient}_i}$ is the public key of Patient_i and that Patient_i has a private key corresponding to $\text{PK}_{\text{Patient}_i}$, then Sub_i should believe that Patient_i sent message M_1, which includes the patient's signature Psig.

$$
\begin{aligned}
&(\text{Sub}_i| \equiv \text{PK}(\text{Patient}_i, \text{PK}_{\text{Patient}_i})),\\
&(\text{Sub}_i| \equiv II(\text{Patient}_i)),\\
&(\text{Sub}_i \vdash \sigma(M_1, \text{Patient}_i))/\\
&(\text{Sub}_i| \equiv \text{Patient}_i| \sim M_1), \quad \text{G1 is achieved.}
\end{aligned}
$$

G3 is achieved by applying the below freshness rule to A5. That is, if Sub_i believes that part of message M1 is fresh (which is NI), then message M1 is fresh.

$$(Sub_i \equiv \#(M_1)',)/(Sub_i \equiv \#M_1), \text{so G3 is achieved.}$$

G2 is achieved via applying the below rules. From G1 and G3, we can infer that Sub_i believes that $Patient_i$ believes message M1; then, the control rule is applied to A7. Specifically, if Sub_i believes that $Patient_i$ controls message M1, and Sub_i believes that $Patient_i$ believes message M1, then Sub_i should believe that message M1 is true.

$$\begin{aligned}&(Sub_i \equiv Patient_i| \sim M_1,\ Sub_i \equiv \#M_1)/\\&(Sub_i \equiv Patient_i \equiv M_1),\\&\text{and } (Sub_i \equiv Patient_i \Rightarrow (M_1, Psig),\\&Sub_i \equiv Patient_i \equiv M_1)/(Sub_i \equiv M_1), \text{G2 is achieved.}\end{aligned}$$

Goals G4, G5, and G6:

Similarly, G4 is achieved by applying the below signing rule to A1, A2, and M2. That is, if $Patient_i$ receives the signed message M2 that is signed using PKsubi-1 from Sub_i, and $Patient_i$ believes that PKsubi is the public key of Sub_i and that Sub_i has a private key corresponding to PKsubi which is PKsubi-1, then $Patient_i$ should believe that Sub_i said message M2, which includes the session key SK.

$$\begin{aligned}&(Patient_i \equiv PK(Sub_i, PK_{Sub_i}),\\&Patient_i \equiv II(Sub_i),\ Patient_i \vdash \sigma(M_2, Sub_i))/\\&(Patient_i \equiv Sub_i| \sim M_2), \text{G4 is achieved.}\end{aligned}$$

G6 is also achieved by applying the following freshness rule to A6. That is, if $Patient_i$ believes that part of message M2 is fresh (which is N2), then message M2 is fresh.

$$(Patient_i \equiv \#(M_2)',)/(Patient_i \equiv \#M_2), \text{so G6 is achieved.}$$

G5 is achieved via applying the below rules. From G4 and G6, we can infer that $Patient_i$ believes that Sub_i believes message M2; then, we apply the control rule to A8. Specifically, if $Patient_i$ believes that Sub_i controls message M2, and $Patient_i$ believes that Sub_i believes message M2, then $Patient_i$ should believe that message M2 is true.

$$\begin{aligned}&(Patient_i \equiv Sub_i| \sim M_2,\ Patient_i \equiv \#M_2)/\\&(Patient_i \equiv Sub_i \equiv M_2), \text{and } (Patient_i \equiv Sub_i \Rightarrow (M_2, SK),\\&\quad Patient_i \equiv Sub_i \equiv M_2)/(Patient_i \equiv M_2), \text{G5 is achieved.}\end{aligned}$$

Goal G7:

Finally, G7 is achieved by applying the below symmetric rule to A9 and M3. That is, if Sub_i receives message M3 that is encrypted using SK from $Patient_i$, and Sub_i believes that SK is the shared key between Sub_i and $Patient_i$, then Sub_i should believe that $Patient_i$ said message M3, which includes the nonce N2, so $Patient_i$ is authenticated.

$$\begin{aligned}&(Sub_i \vdash \{M_3\}SK,\ Sub_i \equiv Sub_i\Box(\leftrightarrow (SK))\ Patient_i)/\\&(Sub_i \equiv Patient_i| \sim M_3), \text{G7 is achieved.}\end{aligned}$$

References

1. Razdan, S.; Sharma, S. Internet of medical things (IoMT): Overview, emerging technologies, and case studies. *IETE Tech. Rev.* **2022**, *39*, 775–788. [CrossRef]
2. Mishra, P.; Singh, G. Internet of Medical Things Healthcare for Sustainable Smart Cities: Current Status and Future Prospects. *Appl. Sci.* **2023**, *13*, 8869. [CrossRef]
3. Manickam, P.; Mariappan, S.A.; Murugesan, S.M.; Hansda, S.; Kaushik, A.; Shinde, R.; Thipperudraswamy, S. Artificial intelligence (AI) and internet of medical things (IoMT) assisted biomedical systems for intelligent healthcare. *Biosensors* **2022**, *12*, 562. [CrossRef]
4. Ullah, M.; Hamayun, S.; Wahab, A.; Khan, S.U.; Rehman, M.U.; Haq, Z.U.; Rehman, K.U.; Ullah, A.; Mehreen, A.; Awan, U.A.; et al. Smart Technologies used as Smart Tools in the Management of Cardiovascular Disease and their Future Perspective. *Curr. Probl. Cardiol.* **2023**, *48*, 101922. [CrossRef]
5. Hireche, R.; Mansouri, H.; Pathan, A.S.K. Security and privacy management in Internet of Medical Things (IoMT): A synthesis. *J. Cybersecur. Priv.* **2022**, *2*, 640–661. [CrossRef]
6. Omolara, A.E.; Alabdulatif, A.; Abiodun, O.I.; Alawida, M.; Alabdulatif, A.; Hamdan Alshoura, W. Arshad, H. The internet of things security: A survey encompassing unexplored areas and new insights. *Comput. Secur.* **2022**, *112*, 102494. [CrossRef]
7. Abouelmehdi, K.; Beni-Hessane, A.; Khaloufi, H. Big healthcare data: Preserving security and privacy. *J. Big Data* **2018**, *5*, 1–18. [CrossRef]
8. Borgia, E. The Internet of Things vision: Key features, applications and open issues. *Comput. Commun.* **2014**, *54*, 1–31. [CrossRef]
9. Aminizadeh, S.; Heidari, A.; Toumaj, S.; Darbandi, M.; Navimipour, N.J.; Rezaei, M.; Talebi, S.; Azad, P.; Unal, M. The applications of machine learning techniques in medical data processing based on distributed computing and the Internet of Things. *Comput. Methods Programs Biomed.* **2023**, *241*, 107745. [CrossRef]
10. Hasan, M.K.; Ghazal, T.M.; Saeed, R.A.; Pandey, B.; Gohel, H.; Eshmawi, A.; Abdel-Khalek, S.; Alkhassawneh, H.M. A review on security threats, vulnerabilities, and counter measures of 5G enabled Internet-of-Medical-Things. *IET Commun.* **2022**, *16*, 421–432. [CrossRef]
11. Alhaj, T.A.; Abdulla, S.M.; Iderss, M.A.E.; Ali, A.A.A.; Elhaj, F.A.; Remli, M.A.; Gabralla, L.A. A survey: To govern, protect, and detect security principles on internet of medical things (iomt). *IEEE Access* **2022**, *10*, 124777–124791.
12. Alsaeed, N.; Nadeem, F. Authentication in the Internet of Medical Things: Taxonomy, Review, and Open Issues. *Appl. Sci.* **2022**, *12*, 7487. [CrossRef]
13. Rasool, R.U.; Ahmad, H.F.; Rafique, W.; Qayyum, A.; Qadir, J. Security and privacy of internet of medical things: A contemporary review in the age of surveillance, botnets, and adversarial ML. *J. Netw. Comput. Appl.* **2022**, *201*, 103332.
14. Abualghanam, O.; Qatawneh, M.; Almobaideen, W. A survey of key distribution in the context of internet of things. *J. Theor. Appl. Inf. Technol.* **2019**, *97*, 3217–3241.
15. Li, L.; Fan, Y.; Tse, M.; Lin, K.Y. A review of applications in federated learning. *Comput. Ind. Eng.* **2020**, *149*, 106854. [CrossRef]
16. Yang, T.; Andrew, G.; Eichner, H.; Sun, H.; Li, W.; Kong, N.; Ramage, D.; Beaufays, F. Applied federated learning: Improving google keyboard query suggestions. *arXiv* **2018**, arXiv:1812.02903.
17. Rieke, N.; Hancox, J.; Li, W.; Milletari, F.; Roth, H.R.; Albarqouni, S.; Bakas, S.; Galtier, M.N.; Landman, B.A.; Maier-Hein, K.; et al. The future of digital health with federated learning. *NPJ Digit. Med.* **2020**, *3*, 119.
18. Zhang, C.; Xie, Y.; Bai, H.; Yu, B.; Li, W.; Gao, Y. A survey on federated learning. *Knowl.-Based Syst.* **2021**, *216*, 106775.
19. Kanagavelu, R.; Li, Z.; Samsudin, J.; Yang, Y.; Yang, F.; Goh, R.S.M.; Cheah, M.; Wiwatphonthana, P.; Akkarajitsakul, K.; Wang, S. Two-phase multi-party computation enabled privacy-preserving federated learning. In Proceedings of the 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID), IEEE, Melbourne, Australia, 11–14 May 2020; pp. 410–419.
20. Mo, F.; Haddadi, H.; Katevas, K.; Marin, E.; Perino, D.; Kourtellis, N. PPFL: Privacy-preserving federated learning with trusted execution environments. In Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services, Virtual, 24 June–2 July 2021; pp. 94–108.
21. Hsu, C.Y.; Lu, C.S.; Pei, S.C. Image feature extraction in encrypted domain with privacy-preserving SIFT. *IEEE Trans. Image Process.* **2012**, *21*, 4593–4607.
22. Ji, J.; Wang, H.; Huang, Y.; Wu, J.; Xu, X.; Ding, S.; Zhang, S.; Cao, L.; Ji, R. Privacy-preserving face recognition with learnable privacy budgets in frequency domain. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 475–491.
23. Chen, C.; Wu, H.; Su, J.; Lyu, L.; Zheng, X.; Wang, L. Differential private knowledge transfer for privacy-preserving cross-domain recommendation. In Proceedings of the ACM Web Conference 2022, Lyon France, 25–29 April 2022; pp. 1455–1465.
24. Li, A.; Sun, J.; Zeng, X.; Zhang, M.; Li, H.; Chen, Y. Fedmask: Joint computation and communication-efficient personalized federated learning via heterogeneous masking. In Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems, Coimbra, Portugal, 15–17 November 2021; pp. 42–55.

25. Zhang, W.; Yang, D.; Wu, W.; Peng, H.; Zhang, N.; Zhang, H.; Shen, X. Optimizing federated learning in distributed industrial IoT: A multi-agent approach. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 3688–3703.
26. Lim, W.Y.B.; Luong, N.C.; Hoang, D.T.; Jiao, Y.; Liang, Y.C.; Yang, Q.; Niyato, D.; Miao, C. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 2031–2063. [CrossRef]
27. Fang, C.; Guo, Y.; Hu, Y.; Ma, B.; Feng, L.; Yin, A. Privacy-preserving and communication-efficient federated learning in internet of things. *Comput. Secur.* **2021**, *103*, 102199. [CrossRef]
28. Liu, Y.; Yuan, X.; Xiong, Z.; Kang, J.; Wang, X.; Niyato, D. Federated learning for 6G communications: Challenges, methods, and future directions. *China Commun.* **2020**, *17*, 105–118. [CrossRef]
29. Al-Issa, Y.; Ottom, M.A.; Tamrawi, A. eHealth cloud security challenges: A survey. *J. Healthc. Eng.* **2019**, *2019*, 7516035. [CrossRef]
30. Usak, M.; Kubiatko, M.; Shabbir, M.S.; Viktorovna Dudnik, O.; Jermsittiparsert, K.; Rajabion, L. Health care service delivery based on the Internet of things: A systematic and comprehensive study. *Int. J. Commun. Syst.* **2020**, *33*, e4179. [CrossRef]
31. Somasundaram, R.; Thirugnanam, M. Review of security challenges in healthcare internet of things. *Wirel. Netw.* **2021**, *27*, 5503–5509. [CrossRef]
32. AbuAlghanam, O.; Qatawneh, M.; Almobaideen, W.; Saadeh, M. A new hierarchical architecture and protocol for key distribution in the context of IoT-based smart cities. *J. Inf. Secur. Appl.* **2022**, *67*, 103173. [CrossRef]
33. Mohammed, I.A. Cloud identity and access management–A model proposal. *Int. J. Innov. Eng. Res. Technol.* **2019**, *6*, 1–8.
34. Jan, S.U.; Ali, S.; Abbasi, I.A.; Mosleh, M.A.; Alsanad, A.; Khattak, H. Secure patient authentication framework in the healthcare system using wireless medical sensor networks. *J. Healthc. Eng.* **2021**, *2021*, 9954089. [CrossRef]
35. Khan, M.A.; Quasim, M.T.; Alghamdi, N.S.; Khan, M.Y. A secure framework for authentication and encryption using improved ECC for IoT-based medical sensor data. *IEEE Access* **2020**, *8*, 52018–52027. [CrossRef]
36. Xu, Z.; Xu, C.; Chen, H.; Yang, F. A lightweight anonymous mutual authentication and key agreement scheme for WBAN. *Concurr. Comput. Pract. Exp.* **2019**, *31*, e5295. [CrossRef]
37. Alzahrani, B.A.; Irshad, A.; Albeshri, A.; Alsubhi, K. A provably secure and lightweight patient-healthcare authentication protocol in wireless body area networks. *Wirel. Pers. Commun.* **2021**, *117*, 47–69. [CrossRef]
38. Mohammedi, M.; Omar, M.; Bouabdallah, A. Secure and lightweight remote patient authentication scheme with biometric inputs for mobile healthcare environments. *J. Ambient. Intell. Humaniz. Comput.* **2018**, *9*, 1527–1539. [CrossRef]
39. Singh, S.; Rathore, S.; Alfarraj, O.; Tolba, A.; Yoon, B. A framework for privacy-preservation of IoT healthcare data using Federated Learning and blockchain technology. *Future Gener. Comput. Syst.* **2022**, *129*, 380–388. [CrossRef]
40. Alkeem, E.A.; Shehada, D.; Yeun, C.Y.; Zemerly, M.J.; Hu, J. New secure healthcare system using cloud of things. *Clust. Comput.* **2017**, *20*, 2211–2229. [CrossRef]
41. Ullah, I.; Amin, N.U.; Khan, M.A.; Khattak, H.; Kumari, S. An efficient and provable secure certificate-based combined signature, encryption and signcryption scheme for internet of things (IoT) in mobile health (M-health) system. *J. Med. Syst.* **2021**, *45*, 1–14. [CrossRef]
42. Tan, H.; Chung, I. Secure authentication and group key distribution scheme for WBANs based on smartphone ECG sensor. *IEEE Access* **2019**, *7*, 151459–151474. [CrossRef]
43. Ali, R.; Pal, A.K. Cryptanalysis and biometric-based enhancement of a remote user authentication scheme for e-healthcare system. *Arab. J. Sci. Eng.* **2018**, *43*, 7837–7852. [CrossRef]
44. Wang, S.; Cao, Z.; Strangio, M.A.; Wang, L. Cryptanalysis and improvement of an elliptic curve Diffie-Hellman key agreement protocol. *IEEE Commun. Lett.* **2008**, *12*, 149–151. [CrossRef]
45. Burrows, M.; Abadi, M.; Needham, R. A logic of authentication. *ACM Trans. Comput. Syst. (TOCS)* **1990**, *8*, 18–36. [CrossRef]
46. Shang, T.; Liu, J. *Secure Quantum Network Coding Theory*; Springer: Berlin/Heidelberg, Germany, 2020.
47. Yu, S.; Park, K.; Lee, J.; Park, Y.; Park, Y.; Lee, S.; Chung, B. Privacy-preserving lightweight authentication protocol for demand response management in smart grid environment. *Appl. Sci.* **2020**, *10*, 1758. [CrossRef]
48. Sierra, J.M.; Hernández, J.C.; Alcaide, A.; Torres, J. Validating the Use of BAN LOGIC. In Proceedings of the Computational Science and Its Applications–ICCSA 2004: International Conference, Assisi, Italy, 14–17 May 2004; Proceedings, Part I 4; Springer: Berlin/Heidelberg, Germany, 2004; pp. 851–858.
49. Saadeh, M.; Sleit, A.; Sabri, K.E.; Almobaideen, W. Hierarchical architecture and protocol for mobile object authentication in the context of IoT smart cities. *J. Netw. Comput. Appl.* **2018**, *121*, 1–19.
50. Wesam Almobaideen, M.S. Lightweight Authentication for Mobile Users in the Context of Fog Computing. *Int. J. Adv. Comput. Eng. Netw.* **2018**, *6*, 17–22.
51. Bos, J.W.; Halderman, J.A.; Heninger, N.; Moore, J.; Naehrig, M.; Wustrow, E. Elliptic curve cryptography in practice. In Proceedings of the Financial Cryptography and Data Security: 18th International Conference, FC 2014, Christ Church, Barbados, 3–7 March 2014; Revised Selected Papers 18; Springer: Berlin/Heidelberg, Germany, 2014; pp. 157–175.
52. Adalier, M.; Teknik, A. Efficient and secure elliptic curve cryptography implementation of curve p-256. In Proceedings of the Workshop on Elliptic Curve Cryptography Standards, NIST, Gaithersburg, MA, USA, 11 June 2015; Volume 66, pp. 2014–2017.

53. Al-Zubaidie, M.; Zhang, Z.; Zhang, J. Efficient and secure ECDSA algorithm and its applications: A survey. *arXiv* **2019**, arXiv:1902.10313.
54. Maimuţ, D.; Matei, A.C. Speeding-Up Elliptic Curve Cryptography Algorithms. *Mathematics* **2022**, *10*, 3676. [CrossRef]
55. Kanchan, S.; Jang, J.W.; Yoon, J.Y.; Choi, B.J. Efficient and privacy-preserving group signature for federated learning. *Future Gener. Comput. Syst.* **2023**, *147*, 93–106. [CrossRef]
56. Liu, W.; Zhang, Y.; Han, G.; Cao, J.; Cui, H.; Zheng, D. Secure and efficient smart healthcare system based on federated learning. *Int. J. Intell. Syst.* **2023**, *2023*, 8017489. [CrossRef]
57. Ramalingam, P.; Pabitha, P. Ask-ram-imot: Autonomous shared keys based remote authentication method for internet of medical things applications. *Wirel. Pers. Commun.* **2023**, *131*, 273–293.
58. Jiby, J. Puthiyidam, Shelbi Joseph, B.B. Enhanced authentication security for IoT client nodes through T ECDSA integrated into MQTT broker. *J. Supercomput.* **2024**, *80*, 8898–8932.
59. Jiby, J. Puthiyidam, Shelbi Joseph, B.B. Temporal ECDSA: Atime stamp and signature mask enabled ECDSA algorithm for IoT client node authentication. *Comput. Commun.* **2024**, *216*, 307–323.
60. Yang, X.b.; Liu, Y.; Wu, J.s.; Han, G.; Liu, Y.x.; Xi, X.q. Nomop-ecdsa: A lightweight ecdsa engine for internet of things. *Wirel. Pers. Commun.* **2021**, *121*, 171–190. [CrossRef]
61. Logeshwaran, J.; Shanmugasundaram, N.; Lloret, J. Energy-efficient resource allocation model for device-to-device communication in 5G wireless personal area networks. *Int. J. Commun. Syst.* **2023**, *36*, e5524. [CrossRef]
62. Subramanian, E.; Tamilselvan, L. Elliptic curve Diffie–Hellman cryptosystem in big data cloud security. *Clust. Comput.* **2020**, *23*, 3057–3067. [CrossRef]
63. Kumar, M. A secure and efficient authentication protocol based on elliptic curve diffie-hellman algorithm and zero knowledge property. *Int. J. Soft Comput. Eng.* **2013**, *3*, 137–142.