Hindawi
Security and Communication Networks
Volume 2020, Article ID 8899409, 18 pages
https://doi.org/10.1155/2020/8899409

Research Article

Enhancing Transaction Security for Handling Accountability in Electronic Health Records

Chian Techapanupreed and Werasak Kurutach

Faculty of Information Science and Technology, Mahanakorn University of Technology, Bangkok 10530, Thailand

Correspondence should be addressed to Chian Techapanupreed; wiztech_ict@yahoo.com

Received 14 March 2020; Revised 28 July 2020; Accepted 8 August 2020; Published 1 September 2020

Academic Editor: Petros Nicopolitidis

Electronic healthcare systems have received extensive attention during the last decade due to the advancement of digital technology. Using these systems in the healthcare industry can improve the quality of healthcare services tremendously. However, a major issue that needs to be concerned, when utilizing this kind of system, is accountability. Employments of electronic health records, the core of the systems, without accountability can be a big risk to both patients and service personals and, consequently, to the entire society. Accountability in electronic health records is essential to creating trust among parties. Many researchers have been introduced to the accountability protocol. However, most of them still lack some essential security property that is mutual authentication. This leads to both information traceability and nonrepudiation which are necessary for resolving any conflict that may arise. In this paper, we propose accountability protocol for electronic health records; the protocol employs both asymmetric and symmetric encryptions to ensure that the electronic health records are having confidentiality, integrity, authentication, and authorization. The accountability analysis and performance analysis show that the proposed protocol is more capable and effective than others. The novel aspect of this idea lies in the inclusion of certain forms of security that are necessary to protect the patient's electronic health records. To the best of our knowledge, the proposed protocol consumes less cost, energy, and time compared with the existing protocols. A proof of concept of our protocol is also presented in this paper by using BAN logic, an automated security protocol proof tool named Scyther, and AVISPA

1. Introduction

During the past decade, electronic health records (EHRs) have been an attractive topic in the healthcare industry. It has been recognized that EHRs can improve the quality of healthcare services tremendously. However, the realization of the systems in the real world is not straightforward because it still faces a major obstacle to transaction accountability concerning patient data. Obviously, with no accountability, patient privacy could be violated and confidential data could be leaked. As a result, their personal life could be ruined. Thus, in order to implement EHRs successfully, the accountability issue needs to be treated properly. The formal definitions of accountability in information systems have been presented in various ways. For example, Feigenbaum et al. and Weitzner et al. [1, 2] defined accountability as referring to an entity that is accountable concerning a certain policy. If this entity violates accountability, a punishment will be raised. According to Gajanayake et al. [3–5], information accountability concerns the use of information where the user is held liable to explain, justify, or answer for its use when so requested by the party to whom the information belongs. Accountability in the computer security systems is the requirement that actions of an entity may be traced uniquely to that entity and directly supports nonrepudiation, deterrence, fault isolation, intrusion detection and prevention, and after-action recovery and legal action that involve confidentiality, integrity, authentication, and authorization of the transaction by all relevant parties [6]. However, the accountability in healthcare is an audit trail, a record of exactly what was done, who did it, and how they did it; in other words, verify, analyze, and investigate users' actions [7, 8].

Basically, the systems must be able to authenticate every party who is authorized to carry out a transaction and also to keep all transactional data confidential and integrity. When a dispute arises, the systems can disclose necessary information to clarify all activities within the transaction in order to resolve it. This will make all involved parties unable to deny their actions in the transaction. Until recently, there have been a lot of researchers investigating accountability in healthcare records. However, most of them [9–16] proposed protocols that cannot handle full accountability because of the lack of mutual authentication, which is a preliminary mechanism for nonrepudiation. Hence, when a dispute arises, transactional information cannot be traced, and an engaging party may be able to deny its action. That is to say, the dispute cannot be resolved, and therefore, there is no accountability in the system. In this paper, we proposed an accountability model and a novel protocol that can handle the accountability, as defined by the model, in healthcare records and their transactions. The protocol possesses all necessary properties of security and can provide confidential evidence to a trustworthy party in case of a dispute. The details of existing protocols will be discussed in Section 2, and the comparison of the existing protocols and the proposed protocol is given in Section 4.

This paper is organized as follows. Section 2 discusses some related works and existing protocols. Section 3 presents the techniques of our proposed protocol. Section 4 discusses security analysis using BAN logic, Scyther tool, and AVISPA. Section 5 discusses the accountability analysis of the protocol. Finally, Section 6 concludes our work.

2. Related Works

Accountability in computer security is a crucial security property that leads to nonrepudiation of engaging parties relevant to the transactions. Hence, many researchers have proposed a security protocol for electronic health records to eliminate any barriers or disputes that may arise after the transaction is complete. Although accountability has been used in many different ways in terms of information accountability for electronic health records, all of these approaches have the same goal. Gajanayake et al. [15, 17, 18] proposed the role-based access control to control the use of patient health records by designing fixed roles for each position in the hospital. Anyway, the role can be replaced by the administrator of the system. This can lead to a serious security problem. Mashima and Ahamad [9, 10] proposed a patient's centric protocol for monitoring the uses of patient health records. The protocol was flexible and helpful to the patient for monitoring who was using his/her health records. However, the protocol still lacks integrity and nonrepudiation properties. Hou and Yeh [11] proposed authentication schemes to ensure authentication between the user, the authentication server, and a trusted third-party authority before using the patient's data. The proposed schemes were lightweight but lacked integrity, nonmutualize authentication, and did not support accountability. Al Alkeem et al. and Al Ameen et al. [12, 19] proposed the systems using the cryptographic protocol to send and receive the patient's data via the cloud system. The system was well proposed but with weak authentication. The strong asymmetric encryption protocol was proposed by Lo et al. [13] with fourteen messages sent via the network. This makes the authors' protocol very secure and encountered all security properties; however, it is rising the computational cost, cryptographic operations cost, energy consumption and time consumption. However, Ibrahim et al. [14] proposed a framework to exchange information between healthcare providers using a hybrid cryptographic operation to offer the security of the system. The proposed framework is complying with most security properties but still lacks accountability, is not resistant for a replay attack and man-in-the-middle attack, and does not support mutual authentication. Hu et al. [20] proposed the hybrid public key infrastructure solution for HIPPA privacy and security regulations. The proposed method was very strong and secure which was complied with confidentiality, integrity, patient control, and consent exception when in emergencies by using mutual authentication. However, the proposed methods were affecting the performance of the system. Blobel et al. [21] proposed the cross-security platform that consists of seven basic components using a public key infrastructure for authentication and a prototypical privilege management infrastructure for authorization and access control to secure web-based electronic health applications. The author applied an idea from their research in [22, 23]. The proposed system was secure as designed; however, it requires greater computation, communication cost, energy consumption, and time consumption. As discussed above, many proposed protocols are secure and with strong encryption. Nevertheless, some protocols are weak in accountability and security properties such as confidentiality, integrity, and authentication. Meanwhile, some protocols are secure and achieve all security properties but have affected the performance of the systems. Moreover, when a dispute arises, these protocols cannot resolve.

3. Proposed Model and Protocol for Electronic Health Records

In the context of information security, many researchers define accountability as involving confidentiality, authorization, authentication, integrity, and nonrepudiation. However, we argue that traceability is also one of the most important features that needs to be part of the systems' accountability. It allows the trustworthy proof of the identity of any participating party and his/her activities in transaction processing. This traceable information will help to solve any dispute that may arise after the end of the transaction. This notion is essential for the success of any electronic healthcare records. Therefore, in our work, we will concentrate on a model of accountability and its supporting protocol to accommodate traceability in electronic health records. In this section, the proposed model and protocol for electronic health records are described.

Figure 1 illustrates conceptually how three engaging parties interact with each other in the healthcare transaction. Whenever C (a hospital, insurance company, or technical lab) needs to access P's health records, a request is sent to the *HCP* for authorization (Step 1). When the *HCP* receives the request

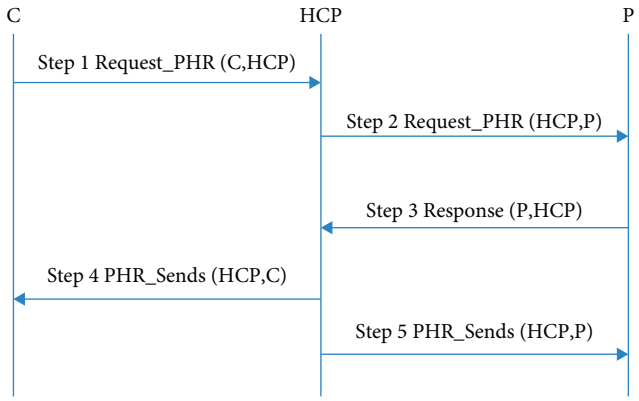

Figure 1: The conceptual view of the healthcare transaction.

from *C*, the identity of *C* is validated. If correct, the *HCP* will forward a request to *P* (Step 2); if not, the *HCP* sends a notification back to *C*, and the transaction will end. When the patient receives a request for authorization from the *HCP*, he/she decides whether to accept or reject this request. *P* will then notify the *HCP* of this decision (Step 3). In case of acceptance, the encrypted patient health records are sent to both *C* and *P* (Steps 4 and 5, respectively). If rejected, *HCP* will send a denial of the request to *C*, and the transaction will end. The details of the protocol are explained in the next section.

3.1. Accountability Model. In this section, we propose a new accountability model for electronic health records. As illustrated previously in Figure 1, there are three parties involved in the healthcare transactions: healthcare professional (*HCP*), the patient (*P*), and an information consumer (*C*). The model defines the accountability of each party and is a baseline to design the proposed protocol of healthcare transactions. The details of the proposed model are described as follows.

Healthcare professional's accountability:

HCP CanProve (C authorized ReqPHR(C, HCP)) to V ∧
HCP CanProve (P authorized PHR(P, HCP)) to V ∧
HCP CanProve (HCP authorized PHR(HCP, C)) to V
⟶ HCP is accountable for PHR to V

The above statements show that the *HCP* is accountable for a *PHR*. The details of *HCP's* accountability can be explained as follows. First, *HCP* needs to prove that the request to access the *P's* health records is sent from *C*. Second, *HCP* must also prove that the *HCP* is allowed by *P* to provide the health records to *C*. Finally, *HCP* needs to prove that the health records have been sent to *C* on behalf of *P's* authorization.

Patient's accountability:

P CanProve (HCP authorized ReqPHR(HCP, P)) to V ∧
P CanProve (P authorized PHR(P, HCP) to V∧
P CanProve (HCP authorized PHR(HCP, C)) to V
⟶ P is accountable for PHR to V

The above messages show that *P* is accountable for a *PHR*. The details of the patient's accountability are explained as follows. First, *P* must prove that the request to use his/her personal health record has come from the *HCP*. Second, *P* needs to prove that he/she gives the authorization to use his/her personal health records to the *HCP*. Finally, *P* also needs to prove that his/her personal health records are sent to *C* by the HCP as requested.

Information consumer's accountability:

C CanProve (HCP authorized PHR(HCP, C)) to V ∧
C CanProve (P authorized PHR(P, C)) to V ∧
⟶ C is accountable for PHR to V

From the above statements, *C* is accountable for a *PHR* if *C* can prove that *HCP*, on behalf of *P*, allows C to use *P's* health records and, also, *C* is allowed to use *P's* health records via *HCP*.

3.2. Accountability Protocol. Based on the accountability model introduced in Section 3.1, electronic healthcare records need a protocol that allows access to *P's* health records with traceability and confidentiality. In designing the protocol that is correct and complete, the accountability aspects consisting of, as mentioned in [24], accountability confidentiality, integrity, authorization, authentication, and nonrepudiation need to be considered. The existing protocol in [16] is secure and complete and can protect *P's* health records but to improve our proposed protocol to consuming fewer resources than the protocol proposed in [16]. So, we decide to propose such an accountability protocol that is based on both symmetric and asymmetric operations. The mechanism of the protocol will be discussed in detail in the following sections, and the notations used to describe it are summarized in Table 1.

3.2.1. The Session Keys' Generation and Update. Session keys are one of the core components employed in our protocol. Basically, the session keys need to be generated and shared between two communicating entities. Therefore, before starting the protocol, it is necessary to create and update the session keys between involved parties. This section will describe how this process works.

(i) **C and HCP:**

Step 1: C ⟶ HCP: C_{ID}, HCP_{ID}, $\{\{n_1\}_{Pri\text{-}C}\}_{Pub\text{-}HCP}$, $h(C_{ID}, HCP_{ID}, n_1)$
Step 2: HCP ⟶ C: n_2, $EXPT_{C\text{-}HCP}$, $h(C_{ID}, HCP_{ID}, n_1, n_2)$
Step 3: C ⟶ HCP: $\{EXPT_{C\text{-}HCP}\}SK_{C\text{-}HCP}$, where $SK_{C\text{-}HCP} = h(C_{ID}, HCP_{ID}, n_1, n_2, EXPT_{C\text{-}HCP})$

In Step 1, *C* sends information, including the identities of *C* (C_{ID}) and *HCP* (HCP_{ID}), the nonce n_1 doubly encrypted with the *C*'s private key and the *HCP's* public key, respectively, and the hash value of C_{ID}, HCP_{ID}, and n_1, to *HCP*. The double encryption is to ensure the mutual authentication between *C* and *HCP*, and the hash value is to ensure message integrity.

Table 1: Notations used in the proposed protocol.

Symbol	Definition
P	The subject and owner of the health records. A patient may have more than one PHR and EHR
C	Information consumer: an external entity that accesses PHRs, for example, a hospital, a lab, an emergency medical technician (EMT), or an insurance company
HCP	Healthcare professional: a hospital-based or clinical issuer of patient EHRs
Q	Any party that is involved in the transaction
V	Verifier, external third party
P_{ID}	Identity of the patient
C_{ID}	Identity of the information consumer
HCP_{ID}	Identity of healthcare professionals
Pri-$_Q$	A private key of party Q, issued by a certificate authority
Pub-$_Q$	A public key of party Q, issued by a certificate authority
T_1, T_2	Timestamps
h(M)	One-way hash function of message M
$\{M\}_{Pub\text{-}Q}$	Message M encrypted with the public key of Q
$\{M\}_{Pri\text{-}Q}$	Message M signed with the private key of Q
$SK_{(A\text{-}B)}$	The session keys shared between party A and party B
ReqPHR	The request to use patient health records sent from the information consumer
PHR	Patients' health information required by an involved party
Allow/ NotAllow	The message status is sent from the patient to the healthcare professional to notify that the patient is allowed or not allowed to use the patient health records

In Step 2, after receiving the message in Step 1, *HCP* will decrypt $\{\{n_1\}_{Pri\text{-}C}\}_{Pub\text{-}HCP}$ to obtain n_1 so that it can validate the hash value $h(C_{ID}, HCP_{ID}, n_1)$. Moreover, after integrity of the message has been confirmed, *HCP* will generate a nonce n_2 and, then, send information, composed of n_2, $EXPT_{C\text{-}HCP}$ (expiry date and time of the session key), and the hash value of $h(C_{ID}, HCP_{ID}, n_1, n_2)$, to *C*. In contrast, if integrity is disconfirmed, *HCP* will terminate the connection with *C*.

In Step 3, after receiving the message from *HCP*, *C* will check the correctness of the hash value $h(C_{ID}, HCP_{ID}, n_1, n_2)$. If the hash value is invalid, *C* will decline communication. Otherwise, *C* sends the encrypted message $\{EXPT_{C\text{-}HCP}\}SK_{C\text{-}HCP}$ to *HCP*. Note that $SK_{C\text{-}HCP}$ denotes the hash value of C_{ID}, HCP_{ID}, n_1, n_2, and $EXPT_{C\text{-}HCP}$ that is the shared session key of *C* and *HCP*.

(ii) **C and P:**

Step 1: C $\longrightarrow$ **P**: C_{ID}, P_{ID}, $\{\{n_1\}_{Pri\text{-}C}\}_{Pub\text{-}P}$, $h(C_{ID}, P_{ID}, n_1)$s
Step 2: P $\longrightarrow$ **C**: n_2, $EXPT_{C\text{-}P}$, $h(C_{ID}, P_{ID}, n_1, n_2)$
Step 3: C $\longrightarrow$ **P**: $\{EXPT_{C\text{-}P}\}SK_{C\text{-}P}$, where, $SK_{C\text{-}P} = h(C_{ID}, P_{ID}, n_1, n_2, EXPT_{C\text{-}P})$

In Step 1, *C* sends information, including the identities of *C* (C_{ID}) and *P* (P_{ID}), the nonce n_1 doubly encrypted with the *C's* private key and the *P's* public key, respectively, and the hash value of C_{ID}, P_{ID}, and n_1, to *P*. The double encryption is to ensure the mutual authentication between *C* and *P*, and the last hash value is to ensure message integrity.

In Step 2, after receiving the message in Step 1, *P* will decrypt $\{\{n_1\}_{Pri\text{-}C}\}_{Pub\text{-}P}$ to obtain n_1 so that it can validate the hash value $h(C_{ID}, P_{ID}, n_1)$. Moreover, after integrity of the message has been confirmed, *P* will generate a nonce n_2 and, then, send information, composed of n_2, $EXPT_{C\text{-}P}$ (expiry date and time of the session key), and the hash value of $h(C_{ID}, P_{ID}, n_1, n_2)$, to *C*. In contrast, if integrity is disconfirmed, *P* will terminate the connection with *C*.

In Step 3, after receiving the message from *P*, *C* will check the correctness of the hash value $h(C_{ID}, P_{ID}, n_1, n_2)$. If the hash value is invalid, *C* will decline communication. Otherwise, *C* will send the encrypted message $\{EXPT_{C\text{-}P}\}SK_{C\text{-}P}$ to *P*. Note that $SK_{C\text{-}P}$ denotes the hash value of C_{ID}, P_{ID}, n_1, n_2, and $EXPT_{C\text{-}P}$ that is the shared session key of *C* and *P*.

(iii) **HCP and P:**

Step 1: HCP $\longrightarrow$ **P**: HCP_{ID}, P_{ID}, $\{\{n_1\}_{Pri\text{-}HCP}\}_{Pub\text{-}P}$, $h(HCP_{ID}, P_{ID}, n_1)$
Step 2: P $\longrightarrow$ **HCP**: n_2, $EXPT_{HCP\text{-}P}$, $h(HCP_{ID}, P_{ID}, n_1, n_2)$
Step 3: HCP $\longrightarrow$ **P**: $\{EXPT_{HCP\text{-}P}\}SK_{HCP\text{-}P}$, where, $SK_{HCP\text{-}P} = h(HCP_{ID}, P_{ID}, n_1, n_2, EXPT_{HCP\text{-}P})$

In Step 1, *HCP* sends information, including the identities of *HCP* (HCP_{ID}) and *P* (P_{ID}), the nonce n_1 doubly encrypted with the *HCP's* private key and the *P's* public key, respectively, and the hash value of HCP_{ID}, P_{ID}, and n_1, to *P*. The double encryption is to ensure the mutual authentication between *HCP* and *P*, and the last hash value is to ensure message integrity.

In Step 2, after receiving the message in Step 1, *P* will decrypt $\{\{n_1\}_{Pri\text{-}HCP}\}_{Pub\text{-}P}$ to obtain n_1 so that it can validate the hash value $h(HCP_{ID}, P_{ID}, n_1)$. Moreover, after integrity of the message has been confirmed, *P* will generate a nonce n_2 and, then, send information, composed of n_2, $EXPT_{HCP\text{-}P}$ (expiry date and time of the session key), and the hash value of $h(HCP_{ID}, P_{ID}, n_1, n_2)$, to *HCP*. In contrast, if integrity is disconfirmed, *P* will terminate the connection with *HCP*.

In Step 3, after receiving the message from *P*, *HCP* will check the correctness of the hash value $h(HCP_{ID}, P_{ID}, n_1, n_2)$. If the hash value is invalid, *HCP* will decline communication. Otherwise, *HCP* will send the encrypted message $\{EXPT_{HCP\text{-}P}\}SK_{HCP\text{-}P}$ to *P*. Note that $SK_{HCP\text{-}P}$ denotes the hash value of HCP_{ID}, P_{ID}, n_1, n_2, and $EXPT_{HCP\text{-}P}$ that is the shared session key of *HCP* and *P*.

3.2.2. The Proposed Protocol. The proposed protocol designed based on the proposed model is explained in Section 3.1. The proposed protocol will be used in the case that whenever *C* (a hospital, patient, insurance company, or technical lab) needs to use *P*'s health records, a request is sent to the *HCP* for authorization. When the *HCP* receives the request from *C*, the identity of *C* is checked. If correct, the *HCP* will forward a request to *P*; if not, the *HCP* sends a notification back to *C*, and the transaction will end. When *P* receives a request for authorization from an *HCP*, he/she decides whether to accept or reject this request. *P* will then notify the *HCP* of this decision. In the event of acceptance, *HCP* will send the message of confirmation and request information to *C* and also to *P* for confirmation what information is sent to *C*. If rejected, the *HCP* will send a denial of the request to *C*, and the transaction will end. The proposed protocol handled all security properties with cryptographic techniques, especially for confidentiality, integrity, and authentication. For the confidentiality of the message, we use the public key of the receiver to ensure that only who has the private key can read the message. The integrity of the message can be satisfied by using a hash function. However, for authentication of the message, we use an asymmetric key with a symmetric key to ensure that the sender and the receiver can be identified. The details of the proposed protocol can be explained as follows.

Step 1: C ⟶ HCP: $ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1, \{h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)\}_{Pri\text{-}C}, h(\{h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)\}_{Pri\text{-}C}, SK_{C\text{-}HCP}), h(\{h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)\}_{Pri\text{-}C}, SK_{C\text{-}P})$

In Step 1, *C* generates a request and sends it to *HCP* to get permission for accessing *P* personal health records. This message consists of the following data:

(i) $\{h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)\}_{Pri\text{-}C}$: the data package $(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)$ is hashed and, then, encrypted by *C's* private key. This is to ensure that *C* is the creator of the message and to validate the message integrity.

(ii) $h(\{h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)\}_{Pri\text{-}C}, SK_{C\text{-}HCP})$: these data are considered as a message authentication code between *C* and *HCP*. They can also ensure the integrity of the transmitted data.

(iii) $h(\{h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)\}_{Pri\text{-}C}, SK_{C\text{-}P})$: these data are considered as a message authentication code between *C* and *P*. Due to the inclusion of the session key $SK_{C\text{-}P}$, they can ensure that the originator and the receiver of the message are *C* and *P*, respectively.

Step 2: HCP ⟶ P: $ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1, \{h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)\}_{Pri\text{-}C}, h(h(\{h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)\}_{Pri\text{-}C}, SK_{C\text{-}P}), SK_{HCP\text{-}P})$

From Step 1, after receiving the message from *C*, *HCP* will compute the hash value from the plaintext $ReqPHR, C_{ID}, P_{ID}, HCP_{ID}$, and T_1 and, then, compare it with the received hash value $h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)$. If both hash values are not equivalent, *HCP* will terminate the session. Otherwise, *HCP* will proceed to Step 2, where it will forward the plaintext message as well as the two hash values, $\{h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)\}_{Pri\text{-}C}$, and $h(h(\{h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)\}_{Pri\text{-}C}, SK_{C\text{-}P}), SK_{HCP\text{-}P})$, to *P*. The first hash value is used to authenticate *C* as the message generator and the second to mutually authenticate the sender (HCP) and the receiver (*P*).

Step 3: P ⟶ HCP: $Allow, T_2, \{h(Allow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2)\}_{Pri\text{-}P}, h(\{h(Allow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2)\}_{Pri\text{-}P}, SK_{P\text{-}HCP}), h(\{h(Allow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2)\}_{Pri\text{-}P}, SK_{C\text{-}P})$

From Step 2, after receiving the message from *HCP*, *P* will compute the hash value from $ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1$, and session key of $SK_{C\text{-}P}$ and $SK_{P\text{-}HCP}$ and, then, compare it with $h(\{h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)\}_{Pri\text{-}C}, SK_{C\text{-}P}, SK_{P\text{-}HCP})$. If they are a mismatch, *P* will send *NotAllow* a message and reject this session. The step to reject the session is described in Step 3 of the reject session. Otherwise, *P* will decide whether to give consent to use his/her personal health records or not and send the message to *HCP* as in Step 3. The message includes the following:

(i) $Allow, T2, h(\{h(Allow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2)\}_{Pri\text{-}P}, SK_{P\text{-}HCP})$: *P* will send this message to *HCP* to inform *HCP* that *P* allows *C* to use the personal health record as requested. Also, due to the use of the private key of *P* in encrypting the hash value, the message ensures that *P* is its originator.

(ii) $h(\{h(Allow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2)\}_{Pri\text{-}P}, SK_{C\text{-}P})$: *P* also sends the hash value of $\{h(Allow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2)\}_{Pri\text{-}P}$ encrypted with $SK_{C\text{-}P}$. The purpose of this message is to ensure the integrity of the message and to ensure that *P* is the sender, and *HCP* is the receiver.

Step 4: HCP ⟶ C: $Allow, T_2, \{Allow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2, PHR\}SK_{C\text{-}HCP}, \{h(Allow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2, PHR)\}_{Pri\text{-}HCP}$

After receiving consent from *P* (via Allow message) in Step 3, *HCP* will proceed to Step 4, where *PHR* is confidentially sent to *C*. More specifically, the following are included in the sent message:

(i) $\{Allow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2, PHR\}SK_{C\text{-}HCP}$: this message is the encryption of the data, consisting of $Allow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2$, and *PHR*, using the session key $SK_{C\text{-}HCP}$ to ensure that *C* is the only person who can decrypt the message and read the data.

(ii) $\{h(Allow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2, PHR)\}_{Pri\text{-}HCP}$: this message is the encryption of the hash value using the *HCP's* private key in order to authenticate *HCP* as the creator of the message.

In the event that the message in Step 2 is incorrect or *P* is an inconvenience to allow *C* to use his/her *PHR*, he/she will send the notification message to *HCP* that *P* is not allowed to use his/her *PHR* and terminate this

communication. The message on Step 3 and Step 4 is described as follows:

Step 3: P ⟶ HCP: *NotAllow, T_2, {NotAllow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T2}$_{Pri\text{-}P}$, h({NotAllow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T2}$_{Pri\text{-}P}$, $SK_{P\text{-}HCP}$), h({NotAllow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T2}$_{Pri\text{-}P}$, $SK_{C\text{-}P}$)*

From Step 2, after receiving the message from *HCP*, *P* will compute a hash value from *ReqPHR*, C_{ID}, P_{ID}, HCP_{ID}, T_1, and session key of $SK_{C\text{-}P}$, $SK_{P\text{-}HCP}$ and compare with *h({h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)}$_{Pri\text{-}C}$, $SK_{C\text{-}P}$, $SK_{P\text{-}HCP}$)* that is received from *HCP*. If *P* found that the comparison is incorrect, *P* will send the following message to *HCP* and terminate the session.

(i) *h({NotAllow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T2}$_{Pri\text{-}P}$, $SK_{P\text{-}HCP}$)*: *P* will send this message to *HCP* to inform *HCP* that *P* does not allow *C* to use his/her personal health record as requested. Besides, due to the use of the private key of *P* in encrypting the hash value, the message ensures that *P* is its originator, while the session key $SK_{P\text{-}HCP}$ is to ensure that *P* is the sender, and *HCP* is the receiver of the message.

(ii) *h({NotAllow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T2}$_{Pri\text{-}P}$, $SK_{C\text{-}P}$)*: *P* also sends the hash value of *{h(NotAllow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2)}$_{Pri\text{-}P}$* encrypted with session key $SK_{C\text{-}P}$ to *HCP*. The purpose of this message is to ensure the integrity of the message and to ensure that *P* is the sender, and *C* is the receiver and notify *HCP* that *P* does not allow *C* to use his/her personal health records. When *HCP* received the message from *P*, *HCP* will forward the message to C. This can be sure that *HCP* cannot see the message. This is because the session key $SK_{C\text{-}P}$ is shared only between *P* and *C*.

Step 4: HCP ⟶ C: Not*Allow, T_2, {NotAllow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T2}$_{Pri\text{-}P}$, h({NotAllow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T2}$_{Pri\text{-}P}$, $SK_{C\text{-}P}$)*

After receiving dissent from *P* (via NotAllow message) in Step 3, *HCP* will send the following message to *C*:

(i) *{NotAllow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2}$_{Pri\text{-}P}$*: this message is the encryption of the data, consisting of *NotAllow, C_{ID}, P_{ID}, HCP_{ID}, T_1*, and T_2, using *P's* private key to ensure that *P* is the originator of the message.

(ii) *h({NotAllow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2}$_{Pri\text{-}P}$, $SK_{C\text{-}P}$)*: this message will send *NotAllow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2* encrypt with *P's* private key and session key $SK_{C\text{-}P}$ to ensure that *P* is the creator of the message, and *C* is the only person who can open the message.

Step 5: HCP ⟶ P: *Allow, T_2, {Allow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2, PHR}$SK_{P\text{-}HCP}$, {h(Allow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2, PHR)}$_{Pri\text{-}HCP}$*

After receiving consent from *P* (via Allow message) in Step 3, *HCP* will proceed to Step 5, where *PHR* is confidentially sent to *P* to confirm that *P's* health records are sent to *C*. More specifically, the following are included in the sent message:

(i) *{Allow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2, PHR}$SK_{P\text{-}HCP}$*: this message is the encryption of the data, consisting of *Allow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2*, and *PHR*, using session key $SK_{P\text{-}HCP}$ to ensure that *P* is the only person who can decrypt the message and read the data.

(ii) *{h(Allow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2, PHR)}$_{Pri\text{-}HCP}$*: this message is the encryption of the hash value using the *HCP's* private key to authenticate *HCP* as the creator of the message.

If any dispute arises, the originating party may need to resolve it. In this case, the party can send the transaction to the third party to investigate the problem. The third party may be a court, lawyer, or trusted company. For example, in case that *C* needs to prove that *HCP* has already been granted to provide *P* health records to *C*, *C* needs to send the message, in Step 4, to the third party, while *HCP* needs to send the message, in Step 2 and Step 3, to the third party to prove that *P* has permitted his/her *P* health records to *HCP*. After the third party receives all implicated evidence, they will consider the evidence and notify the result to the involved parties. On the contrary, if *P* needs to prove that *P* did not give any consent to *HCP* and *C*, *P* needs to send a transaction message to the third party to prove that *HCP* or *C* violates *P*'s health records.

4. Security Analysis and Performance Analysis

To analyze the security of the proposed protocol, we address the security concerns of patients: the confidentiality and the integrity of *PHRs*, authentication, authorization, and nonrepudiation of the transactions between the parties involved. An analysis of the proposed protocol is given in the following.

4.1. Security Analysis. The proposed protocol uses asymmetric encryption to ensure that the involved party cannot deny their action. The advantages of using asymmetric encryption are that there is no need to exchange keys, message authentication and nonrepudiation (in which the user cannot deny sending a message) are ensured, and tampering can be detected if the message is altered by an intruder or hacker. This assumes that the private key of each party is not compromised, and the message was successfully sent to the involved party.

The details of security analysis are given as follows:

(a) Confidentiality of the message: this is to protect the message from unauthorized disclosure. For example, in Step 4, *HCP* sends the message *Allow, T_2, {Allow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2, PHR}$SK_{C\text{-}HCP}$, {h(Allow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2, PHR)}$_{Pri\text{-}HCP}$*. The confidentiality of the message can be satisfied by using the secret key $SK_{C\text{-}HCP}$ shared between *C* and *HCP* since the key is

held only by C and HCP, meaning that HCP and C can read this message.

(b) Message integrity: this is to assure that information is changed only in a specified and authorized manner of the message. This can be ensured by the message authentication code value. For example, in Step 1, C sends the message $ReqPHR$, C_{ID}, P_{ID}, HCP_{ID}, T_1, $\{h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)\}_{Pri\text{-}C}$, $h(\{h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)\}_{Pri\text{-}C}, SK_{C\text{-}HCP})$, $h(\{h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)\}_{Pri\text{-}C}, SK_{C\text{-}P})$ to HCP. This is used to check whether or not the hash values are equal. If the hash values of the message are equal, the message integrity is satisfied; if not, HCP will reject the message.

(c) Mutual authentication: this is a two-way authentication, i.e., the information in a message can authenticate both originator and receiver. The session key shared between two parties is a mechanism for the mutual authentication. In our protocol, we employ three session keys $SK_{C\text{-}HCP}$, $SK_{C\text{-}P}$, and $SK_{P\text{-}HCP}$. Consider the following message:

Step 1: C ⟶ HCP: $ReqPHR$, C_{ID}, P_{ID}, HCP_{ID}, T_1, $\{h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)\}_{Pri\text{-}C}$, $h(\{h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)\}_{Pri\text{-}C}, SK_{C\text{-}HCP})$, $h(\{h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)\}_{Pri\text{-}C}, SK_{C\text{-}P})$

In the proposed protocol, it can be seen that the originator and the receiver of any message can be identified and authenticated. C and HCP share the session key $SK_{C\text{-}HCP}$, C, and P shares the secret key $SK_{C\text{-}P}$. It can be seen that C cannot deny that C is the originator of the message. This is because C possesses the shared key $SK_{C\text{-}HCP}$ and $SK_{C\text{-}P}$ that indicates C is the only one who can create this message. So, this can infer that C is the originator of this message because both $SK_{C\text{-}HCP}$ and $SK_{C\text{-}P}$ are known only by C. It can be seen that our proposed protocol satisfies all necessary properties. Moreover, the proposed protocol can prevent the replay attack by using a fresh timestamp [25] that can be used only once and man-in-the-middle attack. By using asymmetric cryptography to authenticate transmission and the secret key shared between the sender and the receiver, an attacker cannot impersonate a relevant party. This is proved using the Scyther verification tool [26], and the results of the proposed protocol are shown in Figures 2 and 3. Table 2 shows the security comparison of the proposed protocol and existing protocols.

4.2. Security Proof. In this section, we use the traditional and well-known authentication approach known as BAN logic [27], the Scyther verification tool [26], and AVISPA [28] to prove the soundness and security of the proposed protocol.

4.2.1. Authentication Proof Based on BAN Logic. BAN logic is an authentication proof of a protocol for both symmetric and asymmetric encryption algorithms. BAN logic is proposed to verify the security protocol in [27, 29–35]. The details of the formalization of this logic can be found in [27]. In this section, we will describe only an encryption algorithm as used in the proposed protocol. The notations used in BAN logic are given in Table 3.

Figure 2: Scyther verification of the proposed protocol.

BAN logic rules:

R1: message meaning rule:

$$\frac{Q \text{ believes } \xrightarrow{K} P, Q \text{ sees} \{X\}^{K^{-1}}}{Q \text{ believes } P \text{ said } X}. \tag{1}$$

R2: nonce verification rule:

$$\frac{P \text{ believes fresh}(X), P \text{ believes } Q \text{ said } X}{P \text{ believes } Q \text{ believes } X}. \tag{2}$$

R3: juristic rule:

$$\frac{P \text{ believes } Q \text{ controls } X, P \text{ believes } Q \text{ believes } X}{P \text{ believes } X}. \tag{3}$$

R4: freshness rule:

Figure 3: Scyther autoverification of the proposed protocol.

Table 2: Security comparison of the proposed protocol and existing protocols.

Security aspects	[9]	[11]	[19]	[12]	[13]	[14]	[20]	[21]	[15]	[16]	Proposed
Confidentiality	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Integrity	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Mutual authentication	No	No	No	No	No	No	Yes	Yes	No	Yes	Yes
Authorization	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Nonrepudiation	No	No	No	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Replay attack	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
Man-in-the-middle	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
Dispute resolution	No	No	No	No	No	No	Yes	Yes	No	Yes	Yes

Table 3: Notations of BAN logic [27].

Symbol	Definition
P, Q, R	Principals (parties)
X, Y	Statements
K	Encryption keys
P believes X	P believes X or is entitled to believe X
P sees X	P can read and repeat the message containing X
P said X	P has said X (P believes X)
P controls X	P has jurisdiction over X
Fresh (X)	Formula X is fresh
$\longrightarrow^{K} P$	Public key
$\longrightarrow^{K^{-1}} P$	Private key
$\{X\}_K$	Formula X is encrypted by the public key
$\{X\}_{K^{-1}}$	Formula X is encrypted by the private key
$(X)_K$	The hash value of X using K as a key

$$\frac{P \text{ believes fresh}(X)}{P \text{ believes fresh}(X, Y)}. \tag{4}$$

R5: decryption rule:

$$\frac{P \text{ believes } \xrightarrow{K} P, P \text{ sees} \{X\}_K}{P \text{ sees } X}, \quad \frac{P \text{ believes } \xrightarrow{K} Q, P \text{ sees} \{X\}_{K^{-1}}}{P \text{ sees } X}. \tag{5}$$

R6: belief rule:

$$\frac{P \text{ believes } (X), P \text{ believes } Y}{P \text{ believes } (X, Y)}. \quad (6)$$

Idealizing the protocol:

Step 1: C ⟶ HCP: *ReqPHR*, C_{ID}, P_{ID}, HCP_{ID}, T_1, $\{h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)\} \xrightarrow{K^{-1}} C$, $h(\{h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)\} \xrightarrow{K^{-1}} C, C \overset{SK_{CHCP}}{\longleftrightarrow} \text{HCP})$, $h(\{h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)\} \xrightarrow{K^{-1}} C, C \overset{SK_{CP}}{\longleftrightarrow} P)$

Step 2: HCP ⟶ P: *ReqPHR*, C_{ID}, P_{ID}, HCP_{ID}, T_1, $\{h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)\} \xrightarrow{K^{-1}} C$ $h(\{h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)\} \xrightarrow{K^{-1}} C, \overset{SK_{CP}}{\longleftrightarrow} P$, $P \overset{SK_{PHCP}}{\longleftrightarrow} \text{HCP})$

Step 3: P ⟶ HCP: *Allow*, T_2, $\{h(Allow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2)\} \xrightarrow{K^{-1}} P$, $h(\{h(Allow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2)\} \xrightarrow{K^{-1}} P$, $P \overset{SK_{PHCP}}{\longleftrightarrow} \text{HCP})$, $h(\{h(Allow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2)\} \xrightarrow{K^{-1}} P, C \overset{SK_{CP}}{\longleftrightarrow} P)$

Step 4: HCP ⟶ C: *Allow*, T_2, $\{Allow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2, PHR\} C \overset{SK_{CHCP}}{\longleftrightarrow} \text{HCP}$, $\{h(Allow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2, PHR)\} \xrightarrow{K} \text{HCP}$

Step 5: HCP ⟶ P: *Allow*, T_2, $\{Allow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2, PHR\} P \overset{SK_{PHCP}}{\longleftrightarrow} \text{HCP}$, $\{h(Allow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2, PHR)\} \xrightarrow{K} \text{HCP}$

Initial assumptions:

A. C believes $\xrightarrow{K} \text{HCP}$, B. C believes $\xrightarrow{K^{-1}} C$, C. HCP believes $\xrightarrow{K} C$

D. HCP believes $\xrightarrow{K^{-1}} \text{HCP}$, E. HCP believes $\xrightarrow{K} P$, F. P believes $\xrightarrow{K} \text{HCP}$

G. P believes $\xrightarrow{K^{-1}} P$, H. C believes fresh (T1), I. HCP believes fresh (T1)

J. P believes fresh (T1), K. C believes fresh (T2), L. HCP believes fresh (T2)

M. P fresh (T2)

The goal of the analysis is to prove the correlation between the relevant parties. We set five goals for the analysis given as follows:

G1: *HCP* believes $\{h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)\} \xrightarrow{K^{-1}} C$ from Step 1

G2: *P* believes $h(\{h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)\} \xrightarrow{K^{-1}} C, \overset{SK_{CP}}{\longleftrightarrow} P, P \overset{SK_{PHCP}}{\longleftrightarrow} \text{HCP})$ from Step 2

G3: *HCP* believes $h(\{h(Allow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2)\} \xrightarrow{K^{-1}} P, P \overset{SK_{PHCP}}{\longleftrightarrow} \text{HCP})$ from Step 3

G4: *C* believes $\{h(Allow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2, PHR)\} \xrightarrow{K^{-1}} \text{HCP}$, from Step 4

G5: *P* believes $\{h(Allow, C_{ID}, P_{ID}, HCP_{ID}, T_1, T_2, PHR)\} \xrightarrow{K^{-1}} \text{HCP}$ from Step 5

Due to the limited space, the details of this proof will explain only G1 and G2 to prove that *HCP* and *C* are interacting with each other. That is, *C* sends a request to *HCP*, and *HCP* sends back the required *P* health records to *C* (with consent from *P*). However, we can provide some guidelines to prove G1 and G2 of our protocol as follows:

G1: HCP believes $\{h(C_{ID}, P_{ID}, T1)\} \xrightarrow{K^{-1}} C$:

(1) HCP sees $\{h(C_{ID}, P_{ID}, T1)\} \xrightarrow{K^{-1}} C$

(2) 1, *R*1: HCP believes $\{h(C_{ID}, P_{ID}, T1)\} \xrightarrow{K^{-1}} C$

(3) 2, *L*: HCP believes C_{ID}, P_{ID}

(4) 1, *R*5, *C*: HCP believes $h(C_{ID}, P_{ID}, T1)$
HCP sees $h(C_{ID}, P_{ID}, T1)$

(5) 3, 4: HCP believes $h(C_{ID}, P_{ID}, T1) = h(C_{ID}, P_{ID}, T1)$
5, *R6: HCP* believes $\{h(C_{ID}, P_{ID}, T1)\} \xrightarrow{K^{-1}} C$

It can be seen that goal G1 is proven. This is because *HCP* believes that $\{h(C_{ID}, P_{ID}, T1)\} \xrightarrow{K^{-1}} C$ is sent from *C* to *HCP* by believing rule R6. The proof of the relevant transaction between *HCP* and *C* is given in goal G2.

G2: *C* believes $HCP_{ID}, C_{ID}, P_{ID}, \{\{PHR, T_1\} \xrightarrow{K^{-1}} \text{HCP}$

(1) *C* sees $\{PHR, T_1\}_{\text{Pri-HCP}}$

(2) 1, *R*1: *C* believes $\{PHR, T_1\}_{\text{Pri-HCP}}$

(3) 2, *P*: *C* believes PHR, T_1
3, *C* believes $\{PHR, T_1\}_{\text{Pri-HCP}}$

As shown above, *C* believes that $HCP_{ID}, C_{ID}, P_{ID}, \{\{PHR, T_1\} \xrightarrow{K^{-1}} \text{HCP}$ is sent from *HCP*. Thus, *C* believes that *P* health records are sent from *HCP*. It can be inferred that goal G2 is successfully proven. Thus, it can be concluded that all parties have satisfied secure mutual authentication.

4.2.2. Authentication Proof Based on Scyther Verification. There are many tools for formal verification, as shown in the survey in [36], but the most popular for verification are ProVerif, Scyther, and the AVISPA project. Each of these tools has certain advantages and disadvantages, and the reader can find more information in [36]. The advantage of the Scyther verification tool [26] is its graphical user interface for verification, falsification, and analysis of the cryptographic protocol. We, therefore, used the Scyther verification tool to analyze our proposed protocol. As shown in Figures 2 and 3, the proposed protocol is verified as allowing no attacks. More information about authentication claims such as Alive, Weakagree, Niagree, and Nisynch can be found in [26, 37, 38].

4.2.3. Authentication Proof Based on AVISPA. The AVISPA tool is a well-known tool in which many researchers [39–42] used to verify protocol falsification and specific goal defined in high-level protocol specification language (HLPSL) to prove the security protocol, which also allows us to indicate

the protocol's security properties to be verified. The on-the-fly model checker (OMFC) back-end can be employed for efficient checking falsification of protocols and proving the correctness of the protocol. However, constraint-logic-based attack searcher (ATSE) back-end is used to find attacks to the protocol, and attack trace generation is used to find attack of our proposed protocol. The result of the proposed protocol simulation of OFMC is shown in Figure 4, and ATSE is shown in Figure 5. Figure 6 shows attack trace generation and that our proposed protocol is safe, and Figure 7 shows the HLPSL code specific goal for authentication and goal for secrecy are specified in the goal section. The output shows that the proposed protocol is safe as the specified goals. AVISPA is described more in detail in [28, 43].

4.3. Performance Analysis. In this section, our proposed protocol is evaluated since this protocol employs asymmetric and symmetric encryption and may require a greater computation, communication cost, energy consumption, and time consumption. The communication and computational cost are evaluated to show that our proposed protocol can run in any environment, either on a personal computer or a mobile phone. Meanwhile, energy and time consumption are also appraised.

4.3.1. Communication Cost. The communication cost is calculated from the transmitted message size in our proposed protocol and protocols given in [9, 11–16, 20, 21]. The total bits of the transmitted message are calculated using the size specified in Table 4. The total number of bits of the messages transmitted in each protocol is as follows:

[9]: (M1) + (M2) + (M3) + (M4) + (M5) + (M6) + (M7) + (M8) = (80 + 512 + 80) + (80 + 1024 + 512 + 512) + (80 + 1024 + 512 + 128) + (80 + 1024 + 128 + 64 + 128) + (1024 + 128 + 128 + 64) + (512 + 1024) + (64 + 84 + (64 + 64) = 9104 ∗ n

[11]: (M1) + (M2) + (M3) + (M4) + (M5) + (M6) = = (5 + 64 + 128 + 5 + 64 + 128) + (80 + 80 + 80 + 80 + 80 + 80) + (128 + 128 + 128 + 64 + 128 + 80) + (80 + 1024 + 128 + 1024 + 80) + (80 + 1024) + (80 + 1024) + (80 + 80 + 1024 + 1024) = 8282 ∗ n

[12]: (M1) + (M2) + (M3) + (M4) + (M5) + (M6) = (80 + 80 + 80 + 128) + (80 + 80 + 80 + 80 + 80 + 128) + (128 + 80 + 128 + 80 + 80 + 80 + 128 + 128 + 80 + 128 + 80 + 80 + 80 + 80 + 128) + (128 + 80 + 128 + 80 + 80 + 80 + 128 + 128 + 80 + 128 + 80 + 80 + 80) + (128 + 80 + 80 + 80 + 80 + 128) + (128 + 1024) = 5312 ∗ n

[13]: (M1) + (M2) + (M3) + (M4) + (M5) + (M6) + (M7) + (M8) + (M9) + (M10) + (M11) + (M12) + (M13) + (M14) = (8 + 8 + 160 + 64 + 128) + (8 + 8 + 160 + 64 + 128) + (8 + 128 + 8 + 160 + 64 + 128) + (8 + 8 + 160 + 64) + (8 + 128 + 1024 + 512 + 128 + 512 + 128 + 160 + 128) + (8 + 8 + 128 + 128 + 1024 + 128 + 160 + 128) + (8 + 8 + 160 + 64) + (8 + 160 + 64) + (8 + 128 + 128 + 160 + 64) + (8 + 8 + 128 + 128 + 160 + 64) + (8 + 8 + 160 + 64) + (8 + 8 + 512 + 64) + (8 + 8 + 128 + 128 + 512 + 160 + 64) + (8 + 128 + 512 + 160) = 10016 ∗ n

[14]: (M1) + (M2) + (M3) + (M4) + (R1) = (128 + 80 + 80 + 80) + (128 + 80 + 80 + 80 + 160) + (128 + 80 + 7) + (128 + 1024 + 128) + (160 + 1024 + 1024 + 128) = 4727 ∗ n

[20]: (M1) + (M2) + (M3) + (M4) + (M5) + (M6) + (M7) + (M8) + (M9) + (M10) = (80 + 128 + 128 + 80 + 1024) + (128 + 128 + 80 + 128) + (128 + 128 + 80 + 128) + (128 + 128 + 80 + 128 + 128) + (128 + 80 + 80 + 128 + 128 + 1024) + (80 + 6 + 128 + 128 + 80) + (128 + 128 + 80 + 128) + (128 + 128 + 80 + 128) + (128 + 128 + 80 + 128 + 128) + (128 + 80 + 80 + 128 + 128 + 1024 + 128 + 160) = 8326 ∗ n

[21]: (M1) + (M2) + (M3) + (M4) + (M5) + (M6) + (M7) + (M8) + (M1) + (M2) + (M3) + (M4) + (M5) + (M6) + (M7) + (M8) + ((100 + 420 + 420 + 1024 + 420 + 500 + 40 + 1024) + (40 + 40 + 40 + 40 + 40 + 40 + 1124 + 332 + 332 + 332 + 3 + 3 + 1024)

[15]: (M1) + (M2) + (M3) + (M4) + (M5) + (M6) + (M7) = (80) + (80 + 1024) + (80) + (1024 + 1024) + (1024) + (1024) + (1024) = 6384 ∗ n

[16]: (M1) + (M2) + (M3) + (M4) + (M5) + (M6) = The proposed protocol: (M1) + (M2) + (M3) + (M4) + (M5) = (6 + 80 + 80 + 80 + 64 + 128 + 160 + 160) + (6 + 80 + 80 + 80 + 64 + 128 + 160) + (5 + 64 + 128 + 160 + 160) + (5 + 64 + 5 + 80 + 80 + 80 + 64 + 64 + 1024 + 128 + 128) + (5 + 64+5 + 80 + 80 + 80 + 64 + 64 + 1024 + 128 + 128) = 5317 ∗ n

As shown in Figure 8, on the one hand, our proposed protocol takes communication costs less than the protocols proposed in [9, 11, 13, 15, 16, 20, 21]. On the other hand, the proposed protocol takes communication costs more than the protocols proposed in [12, 14]. However, the proposed protocols in [12, 14] lack authentication, authorization, nonrepudiation, and accountability.

4.3.2. Cryptographic Operations' Cost. The cryptographic operations' cost is compared between our proposed protocol and existing protocols [9, 11–16, 20, 21]. The definitions of the cryptographic operations used are presented in Table 4. The security functions used for authentication in these protocols are calculated as follows:

[9]: (SK, SK,{}Pub, {}Pri, {}Pri), (SK, SK, H), ({}Pub, SK, H), ({}Pub,{}Pub, SK), (SK, SK, {}Pri), ({}Pri), (SK, H) = (5 + 3+3 + 3+3 + 1+2 = 20) = 20 ∗ n

[11]: (SK, H), (SK, H, H), (SK, H, H), (SK, H, H) = (2 + 3+3 + 3 = 11) = 11 ∗ n

[12]: (SK), (SK), ({}Pub, SK, SK,{}Pub, SK, {}Pub), ({}Pub, SK, SK, {}Pub, SK), ({}Pub, SK), ({}Pub) = (1 + 1 + 6 + 5 + 2 + 1 = 16) = 16 ∗ n

[13]: ({}Pub, H, SK), ({}Pub, H, SK), ({}Pub, {}Pub, H), ({}Pub, H), ({} Pub, {}Pub, {}Pub, {}Pri, {}Pub, {}Pub, H, SK), ({}Pub, {}Pub, H), ({}Pub, H), ({}Pub, {}Pub, {} Pub, {}Pri, {}Pub, {}Pub, H, SK), ({}Pub, H)}, {}Pub, H,

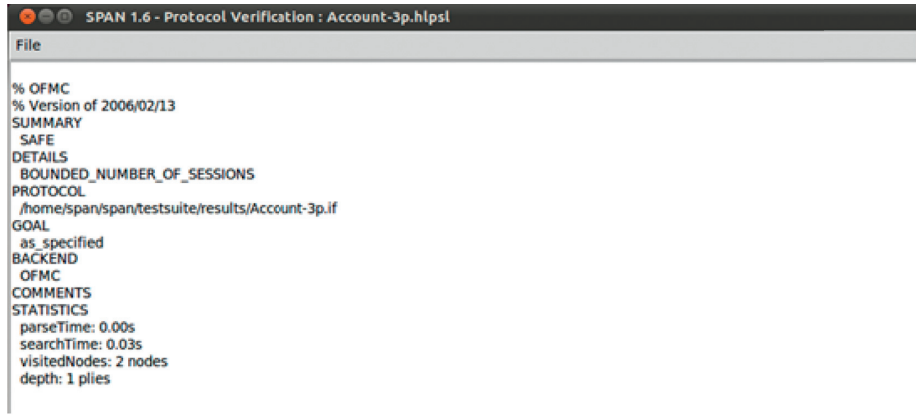

Figure 4: AVISPA OFMC result.

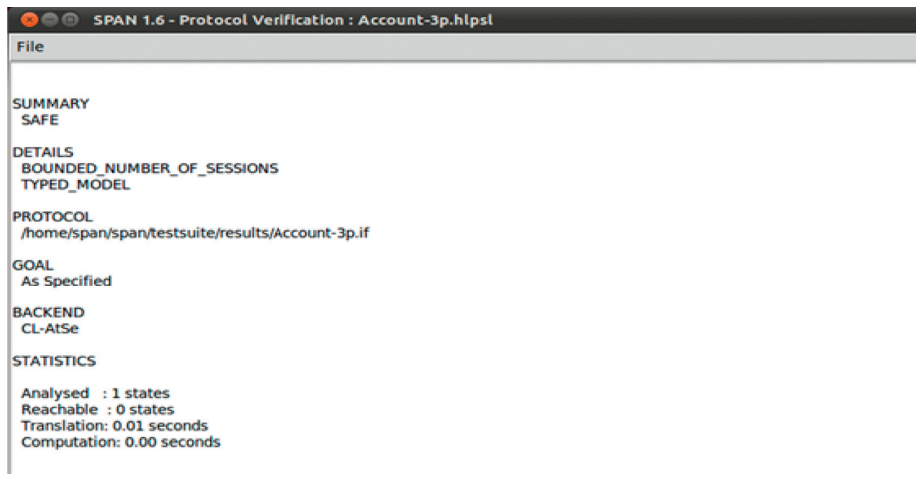

Figure 5: AVISPA ATSE result.

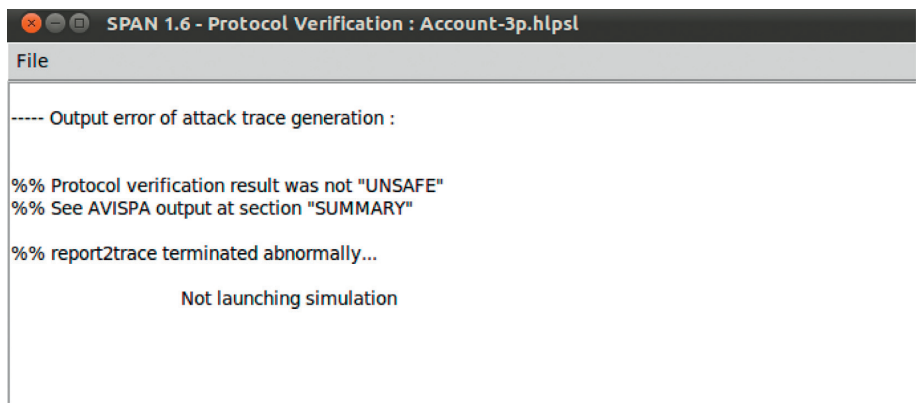

Figure 6: AVISPA attack trace generation.

SK, {}Pub), ({}Pub, {}Pub, SK, SK), ({}Pub, {}Pub, SK, H), ({}Pub, SK), ({}Pub, {}Pri), ({}Pub, {}Pub, {}Pri, {}Pri, {}Pri, {}Pub, {}Pub, ({}Pub, {}Pub, {}Pub, {}Pri, {}Pri, {}Pri, {}Pub, {}Pri,{} Pri, H) = (3 + 3+3 + 10+3 + 2+8 + 2+4 + 4+4 + 2+8 + 9+10 = 67) = 67 ∗ n

[14]: ({}Pub), ({}Pub, H), ({}Pub, H, {}Pub), ({}Pub, {}Pri) = (1 + 2 + 3 + 2 = 8) = 8 ∗ n

[20]: ({}Pub), ({}Pub), ({}Pub), ({}Pub, SK), (SK,{}Pub, H), ({}Pub), ({}Pub), ({}Pub), ({}Pub), ({}Pub, SK, SK) = (1 + 1 + 1 + 2 + 3 + 1 + 1 + 1 + 1 + 3 = 15) = 15 ∗ n

```
role environment() def= const
        hash_0:hash_func,khp:text,phr:text,allow:text,t1:text,cid:text,patient:agent,client:agent,con
st_1:public_key,kh:public_key,kc:public_key,kch:text,hcp:agent,hid:text,pid:text,t2:text,h:hash_func
,reqphr:text,kpc:text,auth_1:protocol_id,sec_2:protocol_id
        intruder_knowledge = {}
        composition
        session1(kc,kh,const_1,kch,client,hcp,patient,hid,cid,pid,t1,t2,allow,h,phr,reqphr,khp,kpc)
end role
goal
        authentication_on auth_1
        secrecy_of sec_2
end goal
environment()
```

Figure 7: HLPSL specification of role environment.

Table 4: Abbreviations and definition of symbols used in the existing and proposed protocols.

Symbol	Definition	Bits
CID/HCPID/PID/$ID_S/C_j/ID_i$	Consumer/healthcare professionals/patient and health authority identity	80
Ti, T_s	Timestamp	64
H/Hash(d)/H(.)/H()	Hash	160
{}Pub/Pub/$PKE_k(p)/K_i^p/K_{pu}^i$	Public key	128
{}Pri/Pri/PKE $k^{-1}(p)/K_i^s/K_{pr}^i$	Private key	128
$SK/SKE_k(p)/Sign_k(d)/K_S/S_j^c$	Session key	128
$PHR/D_{(i)}$/Doc	Personal health records	1024
Allow	Message allowed from the patient	5
ReqPHR/m	Request using patient health records	6
Nonce	An arbitrary number that can be used just once	128

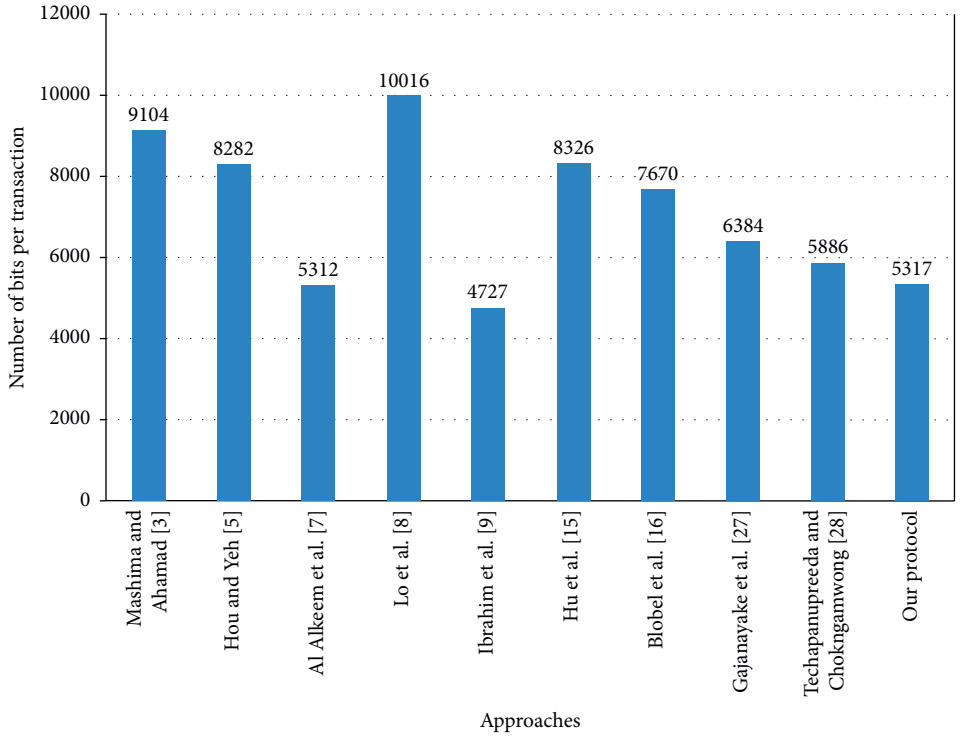

Figure 8: Communication cost.

[21]: ({}Pub, {}Pub, {}Pub, {}Pub, {}Pub, {}Pub, {}Pub, {}Pub, {}Pub, {}Pub) = (1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 = 11) = $11 * n$

[15]: ({}Pub, {}Pub, {}Pub, {}Pub, {}Pub, {}Pub, {}Pub, {}Pub, {}Pub) = (1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 = 9) = $9 * n$

[16] ({Pri}, H, {Pub}), ({Pri}, {Pub}), ({Pri}, H, {Pri}, {Pub}), ({Pri}),({Pri}, H, {Pub}, {Pri}, {Pub}), (H, {Pri}) = (3 + 2 + 4 + 1 + 5 + 2 = 17) = 17 ∗ n

The proposed protocol: (H, {}Pri, H, SK, SK), ({}Pub, H), (H, {}Pri, SK, SK), (SK, {}Pri), (SK) = (5 + 2 + 4 + 3 = 14) = 14 ∗ n

Figure 9 shows cryptographic operations' cost comparison of previous protocols in [9, 11–16, 20, 21]. It can be seen that the protocols [9, 12, 13, 16, 20] have more cryptographic operation cost than our proposed protocol. However, the proposed protocol used cryptographic operation cost more than the existing protocols given in [11, 14, 15, 21]. Nonetheless, protocols given in [11, 14, 15] lack confidentiality, integrity, authorization, nonrepudiation, and accountability. Furthermore, the proposed protocol [9, 11–15] cannot resolve any dispute because they do not provide the dispute resolution phase as mentioned in Table 2.

4.3.3. Energy Consumption. The energy consumption comparison is a comparison of the proposed protocol with the other in [9, 11–16, 20, 21]. The total bits of the transmitted message are also calculated using the size specified in Table 4. The number of messages exchanged in each protocol is 8, 6, 6, 14, 5, 10, 23, 7, and 6, respectively. Table 5 and Figure 10 show the proposed protocol and the other protocols in terms of energy consumption comparison. The comparison shows that the proposed protocol consumes less energy than [9, 12–16, 20, 21]. However, the proposed protocol consumes more energy than [11]. However, the protocol given in [11] still lacks some essential security properties.

4.3.4. Time Consumption. Table 6 and Figure 11 show time consumption comparisons between the proposed protocol and other protocols defined in [9, 11–16, 20, 21]. The comparisons are visible that the proposed protocol is consuming more time than [11]. However, notwithstanding this, the proposed protocol consumes less time than [9, 11–16, 20, 21], and our protocol fully complies with all required security properties.

5. Accountability Analysis

In this section, we investigate the proposed protocol regarding accountability properties. We also provide some guidance to prove the accountability of our proposed protocol. According to the model specified in Section 3.1, it can be seen that our protocol satisfies the accountability properties for all relevant parties: *HCP*, *C*, and *P*. In this, the logic of accountability analysis is derived and is adapted from Wang et al. and Thammarat and Kurutach [33, 34].

5.1. Terms

(i) (Q, R, V, W): the set of parties that communicate with one another in a protocol.

(ii) (X, Y): the set of messages or message components in a protocol.

(iii) (ϕ, ψ): the statements derived from protocol messages.

(iv) $\{K_W, K_W^{-1}\}$: the set of the public key and private key of a party W.

(v) $\{X\}_{KW}$: the message X encrypted with the public key of a party W.

(vi) $\{X\}_{SK}$: the message X symmetrically encrypted with a shared key SK.

(vii) h(X): the hash value of the message X.

(viii) $\xrightarrow{KQ}$ Q: the key K can be used to refer to the party Q.

(ix) W $\overset{SK}{\longleftrightarrow}$ Q: the key SK is a shared key between the parties W and Q.

(x) X-is-fingerprint-of-Y: the message X can be used as a representative (fingerprint) of Y (for example, X may be the hashed form of Y).

(xi) K-is-decrypting-key-for-$\{X\}_K$: the key SK can be used to decrypt the message $\{X\}_{SK}$.

(xii) $\{M\}K_P^{-1}$: the message M signed with the private key of the party.

(xiii) MAC(X, SK): message authentication code (MAC) of the message X with the key SK.

(xiv) $(X)_K$: the message X applied with a single-key cryptographic operation with the key SK. $(X)_K$ can be symmetric-key encryption $(X)_{SK}$, message authentication code MAC(X, SK), or hash function h(SK).

5.2. Formulae

(i) W believes ϕ: W believes that the statement ϕ is true.

(ii) W sees X: some party has sent the message X to W, and W is able to read X.

(iii) W has X: W possesses the message X. W can send X to other parties or use it for further processing.

(iv) W says X: W has sent the message X.

(v) W CanProve ϕ to Q: W can prove to Q that the statement ϕ is true.

(vi) W authorized ReqPHR(W, Q, PHR, T_1): W has authorization on ReqPHR to Q on the date of transaction timestamp.

(vii) W authorized PHR(W, Q, PHR, T_1): W has authorization on requesting Q to use PHR on the date of transaction timestamp.

5.3. Axioms

5.3.1. Inference Rules

M: if ϕ is a theorem, then P believes ϕ is a theorem, where theorem is a formula that can be derived from axioms alone

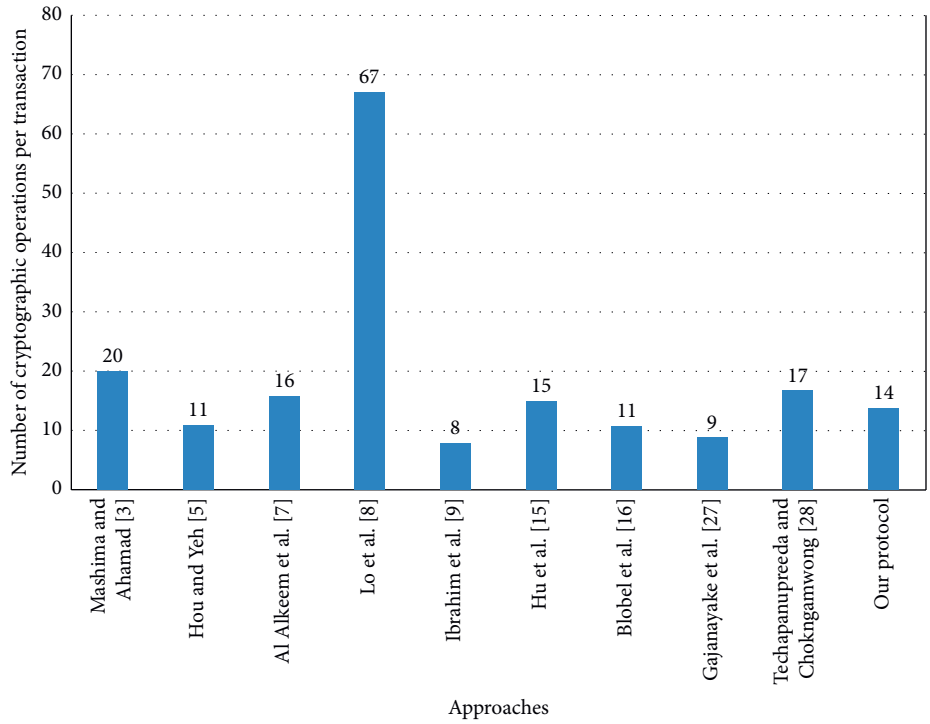

Figure 9: Cryptographic operations' cost.

Table 5: Energy consumption comparisons.

Protocol	AES ($1.21\,\mu$ · J/byte)	RSA ($546.5\,\mu$ · J/byte)	SHA1 ($0.76\,\mu$ · J/byte)	Total
[9]	10.89	4,372.00	2.28	4,385.17
[11]	4.84	—	5.32	10.16
[12]	10.89	3,825.50	—	3,836.39
[13]	10.89	25,139.00	9.12	25,159.01
[14]	—	3,279.00	1.52	3,280.52
[20]	4.84	5,465.00	0.76	5,470.60
[21]	—	6,011.50	—	6,011.50
[15]	—	4,918.50	—	4,918.50
[16]	—	7,104.50	3.04	7,107.54
Proposed	7.26	2,186.00	3.04	2,196.30

5.3.2. Comprehensions

C1: W sees X $\longrightarrow$ W believes W sees X

C2: W says X $\longrightarrow$ W believes W says X

5.3.3. Possessions

H1: W sees X $\longrightarrow$ W has X

H2: (W has $X_1 \wedge \ldots \wedge$ W has X_n) $\longrightarrow$ W has (X_1, :::, X_n), where (X_1, :::, X_n) stand for a list of messages X_1, X_2, :::, X_n, respectively

H3: W has X $\longrightarrow$ W has h(X)

H4: (W has ($\{X\}_{SK}$, SK) $\wedge$ P believes W $\overset{SK}{\longleftrightarrow}$ Q) $\longrightarrow$ W has X

H5: (W has ($\{X\}_{KW}$, K_w^{-1}) $\wedge$ W believes $\overset{KW}{\longrightarrow}$ W) $\longrightarrow$ W has X.

H6: (W has ($\{X\}$ K_w^{-1}, K_W) $\wedge$ P believes $\overset{KW}{\longrightarrow}$ W) $\longrightarrow$ W has X

5.3.4. Provability

P1: (W CanProve ($\phi \longrightarrow \psi$) to V)

$\longrightarrow$ (W CanProve ϕ to V $\longrightarrow$ W CanProve ψ to V)

P2: V-is-external-party $\wedge$ W has X $\wedge$ (V sees X $\longrightarrow$ V believes ϕ)

$\longrightarrow$ W CanProve ϕ to V

P3: W CanProve (Q says $\{X\}_{KR}$) to V $\wedge$

W CanProve ($\overset{KR}{\longrightarrow}$ R) to V $\wedge$

W CanProve (K_R^{-1}-is-decrypting-key-for-$\{X\}_{KR}$) to V

$\longrightarrow$ W CanProve (Q says (X, ID_R)) to V

P4: W CanProve (Q says (X_1, :::, X_n)) to V

$\longleftrightarrow$ [W CanProve (Q says X_1) to V $\wedge \ldots \wedge$ W CanProve (Q says X_n) to V]

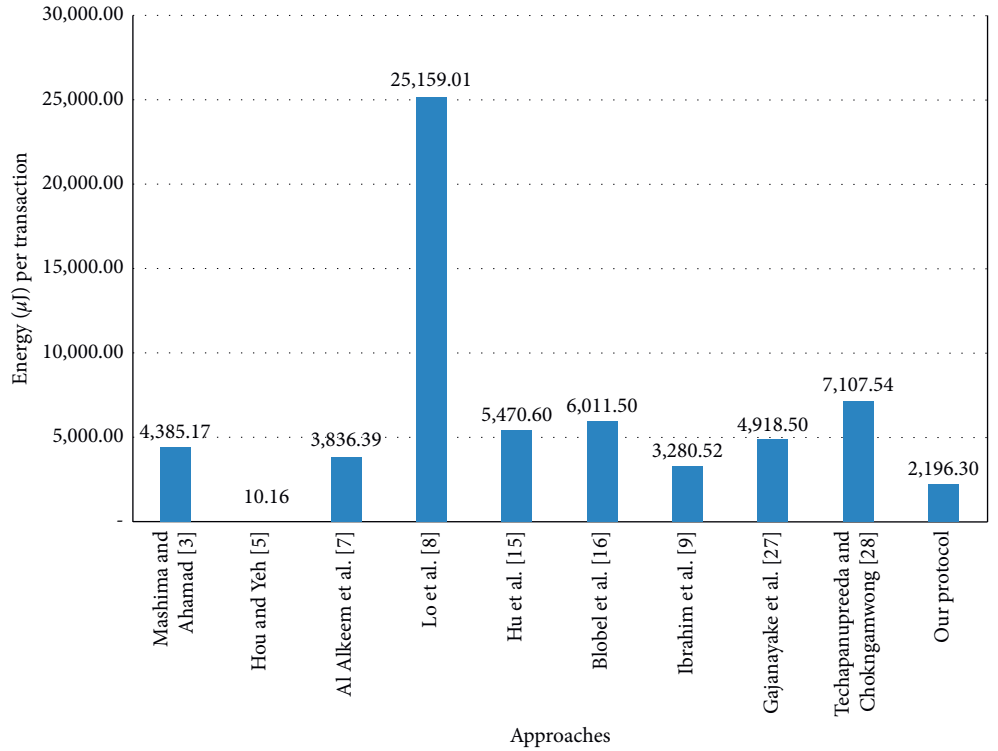

Figure 10: Energy consumption.

Table 6: Time consumption comparisons.

Protocol	AES (1.71 ms/byte)	RSA (15.21 ms/byte)	SHA1 (1.28 ms/byte)	Total
[9]	15.39	121.68	3.84	140.91
[11]	6.84	—	8.96	15.80
[12]	15.39	106.47	—	121.86
[13]	15.39	699.66	15.36	730.41
[14]	—	91.26	2.56	93.82
[20]	6.84	152.10	1.28	160.22
[21]	—	167.31	—	167.31
[15]	—	136.89	—	136.89
[16]	—	197.73	5.12	202.85
Proposed	10.26	60.84	5.12	76.22

5.4. Initial Assumptions

5.4.1. Protocol-Specific Assumption

A1: W believes [(V believes ($Q \overset{SK'}{\longleftrightarrow} R$) $\wedge$ V has ($(X)_{SK}$, SK′) $\wedge$ $X = ((X)_{SK}, SK')$

$\longrightarrow$ V believes ($Q \overset{(X)SK2}{\longleftrightarrow} R$)]

A2: W believes (V believes W′ sees $(X)_{SK}$ $\wedge$

V believes ($W' \overset{SK'}{\longleftrightarrow} Q$) $\wedge$ V has ($(X)_{SK}$, SK′) $\wedge$ $X = ((X)_{SK})_{SK'}$

$\longrightarrow$ V believes Q says (X, $ID_{W'}$))

A3: W believes (V has {}Pri_Q $\longrightarrow$ V believes ($\overset{KQ}{\longrightarrow}$ Q))

A4: W believes [(V has (X, Y) $\wedge$ X = h(Y))

$\longrightarrow$ V believes X-is-fingerprint-of-Y]

A5: W believes (V has ($\{X\}_{SK}$, SK′) $\wedge$ $X = \{\{X\}_{SK}\}_{SK'}$.

$\longrightarrow$ V believes SK′-is-decrypting-key-for-{X}SK)

5.4.2. Shared Secrets

A6: W believes ($C \overset{SK}{\longleftrightarrow} HCP$), W believes ($C \overset{(N)SKC-HCP}{\longleftrightarrow} HCP$)

W believes ($C \overset{SK}{\longleftrightarrow} P$), W believes ($C \overset{(N)SKC-P}{\longleftrightarrow} P$)

W believes ($P \overset{SK}{\longleftrightarrow} HCP$), W believes ($P \overset{(N)SKP-HCP}{\longleftrightarrow} HCP$)

A7: W believes ¬ P sees $SK_{C\text{-}HCP}$, W believes ¬ HCP sees $SK_{C\text{-}P}$

W believes ¬ C sees $SK_{P\text{-}HCP}$

5.4.3. Accountability Information

A8: W believes W has (PHR)

The *HCP* and *C* believe that they possess personal health records of *P*, where W denotes *HCP* and C. Note that C has PHR because P is authorized to use his/her PHR via the request that C sends through HCP.

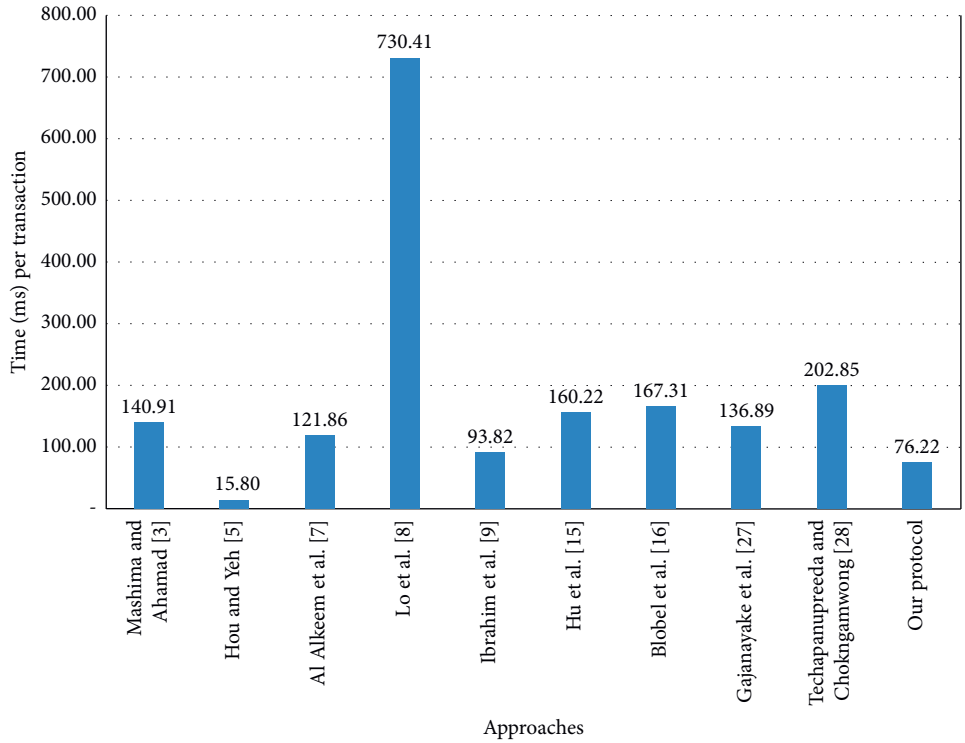

Figure 11: Time consumption.

5.4.4. Personal Health Record Authorizations

> **A9**: every involved party believes that he/she can prove to the verifier that if the *HCP* has sent the message containing *P's* health records, *C* has authorization to use *PHR* from *P*.

W believes W CanProve *(C says* (C_{ID}, *PHR*, T_1) ⟶ *HCP* authorized *PHR(HCP, PHR,* T_1)) to *V*: each party believes that he/she can prove to *V* that if *C* has sent the message containing HCP_{ID}, P_{ID}, and the timestamp of the transaction, *C* has the authorization to requite to use *PHR* from *HCP*.

W believes W CanProve *(P says* (P_{ID}, *PHR*, T_1) ⟶ *HCP* authorized *PHR(HCP, P, PHR,* T_1)) to *V*: each party believes that he/she can prove to *V* that if *HCP* has sent the message containing HCP_{ID}, P_{ID}, *PHR*, and the timestamp T_1, *HCP* has the authorization to use *PHR* from *P*.

W believes W CanProve *(P says* (P_{ID}, *PHR*, T_1) ⟶ *C* authorized *PHR(C, P, PHR,* T_1)) to *V*: each party believes that he/she can prove to *V* that if *P* has sent the message containing C_{ID}, P_{ID}, *PHR*, and the timestamp T_1, *C* has the authorization to use *PHR* from *P*.

5.4.5. Goals of the Analysis. To evaluate the accountability of our proposed protocol, we have to specify the goals that must be satisfied by each engaging party when completing each transaction. Denote that G1, G2, G3, G4, G5, and G6 are our goals. All goals have to be proved according to the proposed model described in section 3.1. The goal of the analysis defines as follows

G1: HCP believes HCP CanProve(C authorized ReqPHR(HCP, C)) to V ⟶ consider Step 1

G2: P believes P CanProve(C authorized ReqPHR(P, C)) to V ⟶ consider Step 2

G3: HCP believes HCP CanProve(P authorized PHR(HCP, P)) to V ⟶ consider Step 3

G4: C believes C CanProve(P authorized PHR(C, P)) to V ⟶ consider Step 4

G5: P believes P CanProve(HCP authorized PHR(P, HCP)) to V ⟶ consider Step 5

G6: C believes C CanProve(HCP authorized PHR(C, HCP)) to V ⟶ consider Step 4

5.5. Details of the Proof. Due to the limitation of space, we give only the analysis of accountability in G1. However, we will provide some guidelines to prove the accountability of our proposed protocol on G1 as follows:

> **G1**: HCP believes HCP CanProve(C authorized ReqPHR(HCP, C)) to V
>
> Consider message Step 1:
>
> **C ⟶ HCP**: *ReqPHR*, C_{ID}, P_{ID}, HCP_{ID}, T_1, $\{h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)\}_{Pri\text{-}C}$, $h(\{h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)\}_{Pri\text{-}C}, SK_{C\text{-}HCP})$, $h(\{h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)\}_{Pri\text{-}C}, SK_{C\text{-}P})$
>
> It can be transformed into the following:
>
> HCP sees *ReqPHR*, C_{ID}, P_{ID}, HCP_{ID}, T_1, $\{h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)\}_{Pri\text{-}C}$, $h(\{h(ReqPHR, C_{ID}, P_{ID},$

HCP_{ID}, T_1)}$_{Pri\text{-}C}$, $SK_{C\text{-}HCP}$), h({h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)}$_{Pri\text{-}C}$, $SK_{C\text{-}P}$)

(1) HCP sees ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1, {h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)}$_{Pri\text{-}C}$, h({h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)}$_{Pri\text{-}C}$, $SK_{C\text{-}HCP}$), h({h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)}$_{Pri\text{-}C}$, $SK_{C\text{-}P}$)

(2) 1, C1, M: HCP sees ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1, {h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)}$_{Pri\text{-}C}$, h({h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)}$_{Pri\text{-}C}$, $SK_{C\text{-}HCP}$), h({h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)}$_{Pri\text{-}C}$, $SK_{C\text{-}P}$)

(3) 2, H1, H2, M: HCP believes HCP has ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1, {h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)}$_{Pri\text{-}C}$, h({h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)}$_{Pri\text{-}C}$, $SK_{C\text{-}HCP}$), h({h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)}$_{Pri\text{-}C}$, $SK_{C\text{-}P}$)

(4) 3, H6, M: HCP believes HCP has Pub-C

(5) 3, 4, P3, A3, K, M: HCP believes HCP CanProve (Pub-C-is-decrypting-key-for-({h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)}$_{Pri\text{-}C}$)) to V

(6) 3, H4, A6, M: HCP believes HCP has h({h(ReqPHR, C_{ID}, P_{ID}, HCP_{ID}, T_1)}$_{Pri\text{-}C}$, $SK_{C\text{-}HCP}$)

(7) 5, 6, A4, P2, H2, M: HCP believes HCP CanProve(C $\overset{h(ReqPHR,CID,PID,HCPID,T1)}{\longleftrightarrow}$ HCP) to V

(8) 7, P4, M: HCP believes HCP CanProve(C says (C_{ID}, ReqPHR, T_1)) to V

(9) 8, A8, M: HCP believes HCP CanProve(C authorized ReqPHR(HCP, C)) to V

It can be seen that goal G1 is successfully proved. Thus, it can be concluded that the *HCP* can prove that *HCP* has given authorization to *C*, and it satisfied with the accountability between *HCP* and *C*. Note that the details of goals G2 to G6 were successfully analyzed.

6. Conclusions

In this paper, we use a BAN logic, Scyther tool, and AVISPA verification tool to prove the completeness and the soundness of the protocol. The results show that the proposed accountability model and accountability protocol achieve our goals in terms of accountability. Firstly, we ensure that the actions of each party can be traced throughout the movement of data. Secondly, we can identify and trace the user, data source, and transactions between parties. Finally, the model and protocol meet the requirements of all crucial securities, that is, confidentiality, authorization, integrity, mutual-authentication, and nonrepudiation. The advantage of our proposed protocol is safe and robust from attack. Moreover, it is having the accountability security property to ensure that the involved party will be confident in using electronic health records and having dispute resolution to resolve any argument that may arise in the future.

Data Availability

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

References

[1] J. Feigenbaum, A. D. Jaggard, and R. N. Wright, "Towards a formal model of accountability," in *Proceedings of the 2011 New Security Paradigms Workshop*, ACM, Marin County, CA, USA, pp. 45–56, September 2011.

[2] D. J. Weitzner, H. Abelson, T. Berners-Lee, J. Feigenbaum, J. Hendler, and G. J. Sussman, "Information accountability," *Communications of the ACM*, vol. 51, no. 6, pp. 82–87, 2008.

[3] R. Gajanayake, T. R. Sahama, B. Lane, and D. Grunwell, "Designing an information accountability framework for ehealth," *e-Health Technical Committee Newsletter*, vol. 2, no. 2, pp. 1-2, 2013.

[4] D. Grunwell, R. Gajanayake, and T. Sahama, "Improving usefulness of ehealth systems through information accountability," *e-Health Technical Committee Newsletter*, vol. 2, no. 6, pp. 3–5, 2013.

[5] R. Gajanayake, T. Iannella, B. Lane, and T. Sahama, *Accountable-eHealth Systems: The Next Step Forward for Privacy*, SRI Security Research Institute, Edith Cowan University, Perth, Australia, 2012.

[6] G. Stoneburner, "Underlying technical models for information technology security," (No. Special Publication (NIST SP)-800-33), National Institute of Standards and Technology, Gaithersburg, MA, USA, 2001.

[7] E. J. Emanuel and L. L. Emanuel, "What is accountability in health care?" *Annals of Internal Medicine*, vol. 124, no. 2, pp. 229–239, 1996.

[8] D. M. Strong, O. Volkoff, S. A. Johnson, L. R. Pelletier et al., "A theory of clinic-EHR affordance actualization," 2009.

[9] D. Mashima and M. Ahamad, "Enabling robust information accountability in e-healthcare systems," in *Proceedings of the 3rd USENIX Workshop on Health Security and Privacy*, Bellevue, WA, USA, August 2012.

[10] D. Mashima and M. Ahamad, "Enhancing the accountability of electronic health record usage via patient-centric monitoring," in *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, ACM, Miami, FL, USA, pp. 409–418, January 2012.

[11] J.-L. Hou and K.-H. Yeh, "Novel authentication schemes for IoT based healthcare systems," *International Journal of Distributed Sensor Networks*, vol. 11, no. 11, Article ID 183659, 2015.

[12] E. Al Alkeem, D. Shehada, C. Y. Yeun, M. J. Zemerly, and J. Hu, "New secure healthcare system using cloud of things," *Cluster Computing*, vol. 20, no. 3, pp. 2211–2229, 2017.

[13] N.-W. Lo, C.-Y. Wu, and Y.-H. Chuang, "An authentication and authorization mechanism for long-term electronic health records management," *Procedia Computer Science*, vol. 111, pp. 145–153, 2017.

[14] A. Ibrahim, B. Mahmood, and M. Singhal, "A secure framework for medical information exchange (MI-X) between healthcare providers," in *Proceedings of the 2016 IEEE International Conference on Healthcare Informatics (ICHI)*, IEEE, Chicago, IL, USA, pp. 234–243, October 2016.

[15] R. Gajanayake, R. Iannella, and T. R. Sahama, "Privacy oriented access control for electronic health records," in *Proceedings of the Data Usage Management on the Web Workshop at the Worldwide Web Conference*, ACM, Lyon, France, March 2012.

[16] C. Techapanupreeda and R. Chokngamwong, "Accountability for electronic-health systems," in *Proceedings of the 2016 IEEE Region 10 Conference (TENCON)*, pp. 2503–2506, IEEE, Singapore, November 2016.

[17] R. Gajanayake, R. Iannella, and T. Sahama, "Sharing with care: an information accountability perspective," *IEEE Internet Computing*, vol. 15, no. 4, pp. 31–38, 2011.

[18] R. Gajanayake, R. Iannella, and T. Sahama, "Privacy by information accountability for e-health systems," in *Proceedings of the 2011 6th IEEE International Conference on Industrial and Information Systems (ICIIS)*, pp. 49–53, IEEE, Kandy, Sri Lanka, August 2011.

[19] M. Al Ameen, J. Liu, and K. Kwak, "Security and privacy issues in wireless sensor networks for healthcare applications," *Journal of Medical Systems*, vol. 36, no. 1, pp. 93–101, 2012.

[20] J. Hu, H. H. Chen, and T. W. Hou, "A hybrid public key infrastructure solution (HPKI) for HIPAA privacy/security regulations," *Computer Standards & Interfaces*, vol. 32, no. 5-6, pp. 274–280, 2010.

[21] B. Blobel, P. Hoepner, R. Joop, S. Karnouskos, G. Kleinhuis, and G. Stassinopoulos, "Using a privilege management infrastructure for secure web-based e-health applications," *Computer Communications*, vol. 26, no. 16, pp. 1863–1872, 2003.

[22] B. Blobel, "Security requirements and solutions in distributed electronic health records," in *Information Security in Research and Business*, pp. 377–390, Springer, Boston, MA, USA, 1997.

[23] B. Blobel, "Architecture of secure portable and interoperable electronic health records," in *Proceedings of the International Conference on Computational Science*, Springer, Amsterdam, The Netherlands, pp. 982–994, April 2002.

[24] C. Techapanupreeda, R. Chokngamwong, C. Thammarat, and S. Kungpisdan, "An accountability model for internet transactions," in *Proceedings of the 2015 International Conference on Information Networking (ICOIN)*, pp. 127–132, IEEE, Siem Reap, Cambodia, January 2015.

[25] W. Stallings, *Cryptography and Network Security*, Pearson Education India, Bengaluru, India, 2006.

[26] C. J. Cremers, "The Scyther tool: verification, falsification, and analysis of security protocols," in *Proceedings of the International Conference on Computer Aided Verification*, Springer, Los Angeles, CA, USA, pp. 414–418, July 2008.

[27] M. Burrows, M. Abadi, and R. M. Needham, "A logic of authentication," *Proceedings of the Royal Society of London. Series A*, vol. 426, no. 1871, pp. 233–271, 1989.

[28] A. Armando, D. Basin, Y. Boichut et al., "The AVISPA tool for the automated validation of internet security protocols and applications," in *Proceedings of the International Conference on Computer Aided Verification*, pp. 281–285, Springer, Edinburgh, Scotland, July 2005.

[29] M. Burrows, M. Abadi, and R. Needham, *Authentication: A Practical Study in Belief and Action (No. UCAM-CL-TR-138)*, University of Cambridge, Computer Laboratory, Cambridge, UK, 1988.

[30] M. Abadi and M. R. Tuttle, "A semantics for a logic of authentication," in *Proceedings of the Tenth Annual ACM Symposium on Principles of Distributed Computing*, pp. 201–216, Montreal, Canada, July 1991.

[31] Y. Chen, J.-F. Martínez, P. Castillejo, and L. López, "A lightweight anonymous client-server authentication scheme for the internet of things scenario: LAuth," *Sensors*, vol. 18, no. 11, p. 3695, 2018.

[32] Y.-F. Chang, W.-L. Tai, and M.-H. Hsu, "A secure mobility network authentication scheme ensuring user anonymity," *Symmetry*, vol. 9, no. 12, p. 307, 2017.

[33] C. Wang, G. Xu, and W. Li, "A secure and anonymous two-factor authentication protocol in multiserver environment," *Security and Communication Networks*, vol. 2018, 15 pages, Article ID 9062675, 2018.

[34] C. Thammarat and W. Kurutach, "A secure fair exchange for SMS-based mobile payment protocols based on symmetric encryption algorithms with formal verification," *Wireless Communications and Mobile Computing*, vol. 2018, 21 pages, Article ID 6953160, 2018.

[35] C. Thammarat and W. Kurutach, "A lightweight and secure NFC-base mobile payment protocol ensuring fair exchange based on a hybrid encryption algorithm with formal verification," *International Journal of Communication Systems*, vol. 32, no. 12, Article ID e3991, 2019.

[36] A. H. Shinde, A. J. Umbarkar, and N. R. Pillai, "Cryptographic protocols specification and verification tools-a survey," *ICTACT Journal on Communication Technology*, vol. 8, no. 2, pp. 1533–1539, 2017.

[37] G. Lowe, "A hierarchy of authentication specifications," in *Proceedings of the 10th Computer Security Foundations Workshop (1997)*, IEEE, Rockport, MA, USA, pp. 31–43, June 1997.

[38] C. Cremers and S. Mauw, *Operational Semantics and Verification of Security Protocols*, Springer Science & Business Media, Berlin, Germany, 2012.

[39] D. Von Oheimb, "The high-level protocol specification language HLPSL developed in the EU project AVISPA," in *Proceedings of APPSEM 2005 Workshop*, pp. 1–17, Münch, Germany, September 2005.

[40] J. A. Hurtado Alegría, M. C. Bastarrica, and A. Bergel, "Analyzing software process models with AVISPA," in *Proceedings of the 2011 International Conference on Software and Systems Process*, pp. 23–32, Honolulu, HI, USA, May 2011.

[41] R. Amin, S. H. Islam, A. Karati, and G. P. Biswas, "Design of an enhanced authentication protocol and its verification using AVISPA," in *Proceedings of the 2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, pp. 404–409, IEEE, Dhanbad, India, March 2016.

[42] P. R. Yogesh and D. S. R, "Formal verification of secure evidence collection protocol using BAN logic and AVISPA," *Procedia Computer Science*, vol. 167, pp. 1334–1344, 2020.

[43] L. Viganò, "Automated security protocol analysis with the AVISPA tool," *Electronic Notes in Theoretical Computer Science*, vol. 155, pp. 61–86, 2006.