sensors

MDPI

Article

Secure and Lightweight Cluster-Based User Authentication Protocol for IoMT Deployment [†]

Xinzhong Su and Youyun Xu *

School of Communication and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China; suxinz20@163.com

* Correspondence: yyxu@njupt.edu.cn

† This is an expanded research article based on the conference paper "A Cluster-based User Authentication Protocol for Internet of Medical Things Deployment" that was presented at 2023 IEEE the 15th International Conference on Wireless Communications and Signal Processing, Hangzhou, China, 2–4 November 2023.

Abstract: Authentication is considered one of the most critical technologies for the next generation of the Internet of Medical Things (IoMT) due to its ability to significantly improve the security of sensors. However, higher frequency cyber-attacks and more intrusion methods significantly increase the security risks of IoMT sensor devices, resulting in more and more patients' privacy being threatened. Different from traditional IoT devices, sensors are generally considered to be based on low-cost hardware designs with limited storage resources; thus, authentication techniques for IoMT scenarios might not be applicable anymore. In this paper, we propose an efficient three-factor cluster-based user authentication protocol (3ECAP). Specifically, we establish the security association between the user and the sensor cluster through fine-grained access control based on Merkle, which perfectly achieves the segmentation of permission. We then demonstrate that 3ECAP can address the privilege escalation attack caused by permission segmentation. Moreover, we further analyze the security performance and communication cost using formal and non-formal security analysis, Proverif, and NS3. Simulation results demonstrated the robustness of 3ECAP against various cyber-attacks and its applicability in an IoMT environment with limited storage resources.

Keywords: Internet of Medical Things; mutual authentication; fine-grained access control; security

Citation: Su, X.; Xu, Y. Secure and Lightweight Cluster-Based User Authentication Protocol for IoMT Deployment. *Sensors* **2024**, *24*, 7119. https://doi.org/10.3390/s24227119

Academic Editors: Cong Wu, Jing Chen, Yebo Feng and Xianhao Chen

Received: 1 October 2024
Revised: 1 November 2024
Accepted: 2 November 2024
Published: 5 November 2024

1. Introduction

The number of connected devices has grown exponentially due to advances in communications technology, resulting in what is known as the Internet of Things (IoT) [1–3]. IOT technology has continued to develop and innovate, profoundly changing traditional industrial models and people's lifestyles, such as smart agriculture, smart healthcare, smart homes, and self-driving cars [4]. And healthcare is rapidly evolving, driven by an aging population, consumer demand for better services in more affordable prices, and a growing global focus on preventative health [5,6]. In recent years, IoMT has been recognized as one of the most important technologies in healthcare, which is used for systematic monitoring of patient status, enabling doctors to provide timely and appropriate treatment [7]. Specifically, IoMT sensors such as defibrillators, sphygmomanometers, and oximeters provide real-time monitoring and observation for patients' temperature, pulse, blood pressure, respiration, and more [8]. Typically, sensors in IoMT are widely accessible and can be installed across geographies as the focus has been on making them multifunctional, low-cost, and available on hardware platforms when coordinated with back-end processing systems. With these new technologies, the prospect of IoMT sensors in healthcare is extremely promising.

Despite the convenience that IoMT brings to patients in terms of treating, diagnosing, and maintaining their health, once the information carried by these sensors is accessed by attackers, it can be a great threat to patient privacy and security [9]. One of the crucial factors to ensure the security of IoMT is node authentication. Usually, the generic architecture of IoMT consists of three node participants, i.e., user, gateway, and sensor. The sensor is placed in a designated area to collect environmental parameters and then transmits these parameters to the gateway through a wireless channel [10]. The user must be authenticated to access these data as the patient data provided by sensors are analyzed and collated to make appropriate and feasible decisions for the timely treatment of the patient.

Specifically, the IoMT system can be simplified into three dimensions, i.e., perception layer, network layer, and application layer [11]. (1) Perception layer: each patient is equipped with a variety of medical sensors used to sense and monitor vital statistics. In this layer, the attacker usually utilizes a device capture attack to obtain patient information inside sensors. (2) Network layer: similar to the OSI network and transport layer, it is responsible for authentication, communication and data transfer between sensors and users via an open channel/private network. However, it is vulnerable to man-in-the-middle attacks, impersonation attacks, replay attacks and so on. (3) Application layer: in this layer, legitimate users/medical staff can realize access to patient information through authentication with the sensors, and it is the top layer of the three-layer IoMT system architecture. However, the application layer is also vulnerable to many attacks such as insider privilege attacks, privilege escalation attacks, etc. Therefore, it is necessary to ensure the privacy and security of patient information in the multilayer architecture of an IoMT system.

Once the information is compromised, the corresponding patient information (including history of illness) may also be exposed to the attacker [12]. Worse, the attacker can even illegally sell this information, thus seriously compromising the patient's personal privacy. In addition, insider attackers (i.e., medical staff) also pose a potential risk of IoMT information leakage. It is extremely necessary to implement permission segmentation according to access levels due to the differences in sensor data accessible to medical staff in different departments (e.g., neurology, gastroenterology, cardiovascular, etc.). Moreover, IoMT is susceptible to various types of attacks, including replay attacks, user privilege escalation attacks, smart card theft attacks, etc., which further compromise the security of the system. Therefore, it is urgent to design a new authentication protocol to ensure the security and privacy of IoMT.

Considering the security, low complexity, and low cost requirements of IoMT, we propose a new efficient cluster-based lightweight secure authentication protocol (3ECAP), with the ultimate goal of establishing a secure session key before participants transmit data. The specific contributions of this paper are as follows.

(1) 3ECAP implements IoMT user permission segmentation using fine-grained access control to establish a security association between the user and the sensor cluster, which reduces subsequent database access costs. Then, the user's password, biometrics and smart card are used as the three factors for authentication, where biometrics are collected through a fuzzy extractor. In addition, the communication cost and computation cost of 3ECAP are further reduced by only performing hash and dissimilarity operations.

(2) The formal security analysis of 3ECAP is demonstrated through the widely used Real or Random (RoR) model and the formal automated verification tool Proverif. In addition, 3ECAP informal security analysis is also provided, which indicates that 3ECAP is not only resistant to most known attacks but also to privilege elevation attacks from insiders (see Section 6.2).

(3) Considering the limited resources of IoMT devices, compared to other schemes, our proposed authentication protocol is not only lightweight and efficient but also resistant to a variety of complex typical attacks.

The rest of this paper is structured as follows: Section 2 presents the literature survey. Some necessary mathematical background is provided in Section 3. The system model utilized in 3ECAP is given in Section 4. Section 5 describes the phases of the designed protocol (3ECAP). In Section 6, the security of 3ECAP is ensured by using formal and informal security analysis. Section 7 presents a comparative analysis of 3ECAP and other related protocols with respect to computational cost, latency, and security characteristics. Section 8 presents a simulation analysis of 3ECAP using a network simulation tool. The last section concludes the paper and gives some future research directions.

2. Related Work

In this section, research advances in the relevant areas are provided, including the methods used and advantages and limitations.

Wang et al. [13] proposed a cloud-assisted secure user authentication scheme with various attributes such as forward secrecy and multi-factor security. However, the scheme requires high computational costs and does not ensure user privacy. Masud et al. [14] proposed a lightweight anonymous user authentication protocol for IoT, which only uses lightweight cryptographic primitives (hash). The scheme establishes a secure session for legitimate users and prohibits unauthorized user access to IoT sensor nodes. Although the protocol has low computational and communication costs, it proved to be vulnerable to attacks such as impersonation and replay. In addition, relevant existing protocols [15–17] are designed for various IoT scenarios, e.g., IoMT, smart firefighting, smart transportation, etc., with provably secure protocols that provide mutual authentication for involved nodes. However, according to recent studies [18–20], the mentioned schemes are susceptible to attacks such as man-in-the-middle, denial-of-service, and internal privilege.

Zhang et al. [21] propose a password-based lightweight security authentication scheme that can flexibly achieve mutual authentication between the user and sensor. Unfortunately, studies have demonstrated that this authentication scheme based on only a single factor can be easily compromised and therefore cannot withstand attacks such as password guessing. To address these problems, Nandy et al. [22] and Singh et al. [23] have proposed security schemes based on multifactor privacy protection. However, Chaudhry et al. [24] point out that the public key of the sensor in the scheme of Nandy et al. [22] is invalid, due to the inability of the device to generate its own private key, and susceptible to clogging attacks. Moreover, the above schemes also require high communication costs.

Nyangaresi et al. [25] propose a lightweight key management and mutual authentication protocol based on Elliptic Curve Cryptography (ECC) for smart home environments. Li et al. [26] design a robust two-factor user authentication protocol based on ECC and prove that the construction of the proposed scheme can achieve user anonymity, forward secrecy of the session key, etc. However, since the above schemes use the ECC algorithm, this significantly increases the communication and computational costs to verify the protocol. Furthermore, Xie et al. [27] proposed a blockchain-based vehicle-to-infrastructure (V2I) authentication protocol using lightweight cryptographic primitives that guarantee sensor anonymity and untraceability. Son et al. [28] design a lightweight mutual authentication protocol for IoT sensors, in which the node performs cryptographic computation only when switching in order to improve the network transmission efficiency. Yang et al. [29] propose a mutual authentication scheme based on decentralized edge collaboration to provide continuous protection for zero-trust IoT and enable flexible updating for the sensor.

By reading and summarizing the above existing studies, we found that existing authentication protocols have low utility in IoMT, e.g., susceptibility to various attacks, high overhead algorithmic application, access control of user authority, high maintenance cost of protocol, and so on. Therefore, we intended to design a lightweight secure and reliable authentication protocol for IoMT to solve the above problems, and some of these research results have been published in the form of a conference [30]. Please note that 3ECAP is an extended version of the published conference paper. Compared to the previous version, 3ECAP contains more comprehensive authentication schemes, security analyses, simulations, graphs, results and utilities. The relevant changes are indicated in the text. Table 1 summarizes the relevant work described above.

Table 1. Related works.

Reference	Method	Advantage (+)	Limitation (−)
Wang et al. [13]	ECC, hash, fuzzy extractor	+three-factor authentication +forward secrecy	−high computational cost −user privacy −lack of access control
Masud et al. [14]	hash, password	+lightweight authentication +node anonymity	−impersonation attack −replay attack −lack of access control
Sutrala et al. [15]	ECC, hash	+impersonation attack protection +MITM attack protection	−privilege-insider attack −high computational cost −lack of access control
Iqbal et al. [16]	hash, symmetric encryption	+privacy-preserving +node anonymous	−impersonation attack −replay attack −lack of access control
Wei et al. [17]	ECC, hash, pseudo random function	+privacy-preserving +system secret key update	−impersonation attack −lack of access control
Zhang et al. [21]	hash, password, homomorphic encryption	+key leakage protection +anonymity and untraceability	−password guessing attack −lack of access control −high resource cost
Nandy et al. [22]	hash, ECC, RSA or DSA	+privacy-preserving +forward secrecy −insider attack protection	−clogging attack −high resource cost −lack of access control
Singh et al. [23]	hash, fuzzy extractor, PUF	+two-factor authentication +physical layer security	−MITM attack −replay attack −high resource cost −privilege escalation attack
Nyangaresi et al. [25]	hash, ECC, password	+replay attack protection +impersonation attack protection +MITM attack protection	−anonymity and untraceability −device capture attack −high resource cost −lack of access control
Li et al. [26]	hash, ECC	+three-factor authentication +forward secrecy +device capture attack protection +impersonation attack protection	−high resource cost −untraceability −lack of access control
Xie et al. [27]	hash, ECC, PUF	+device capture attack protection +MITM attack protection +impersonation attack protection	−privilege-insider attack −forward secrecy −lack of access control
Son et al. [28]	hash, ECC, password	+anonymity and untraceability +ephemeral key leakage protection	−device capture attack −high calculation cost −lack of access control
Yang et al. [29]	hash, ECC, bilinear pairing	+device update +token forgery attack protection	−device capture attack −privacy disclosure −high resource cost −lack of access control

3. Preliminaries

3.1. One-Way Hash Function

A one-way hash function can transform an input message string of arbitrary length into a fixed-length output. It is widely used in areas such as the generation of message digests and message authentication codes, key encryption, and data integrity tests. Collision resistance is the main property and is defined as follows.

Definition 1. *Suppose a one-way hash function can be expressed as* $h : \{0,1\}^* \rightarrow \{0,1\}^n$*. Specifically, the hash function outputs a fixed-length binary string* $h(m) \in \{0,1\}^n$ *for an arbitrary-length input binary string* $m \in \{0,1\}^*$*. Assume* $Adv_{\mathcal{A}}^{HASH}(t)$ *is defined as the probability of an adversary obtaining* *a hash collision in execution time t, then* $Adv_{\mathcal{A}}^{HASH}(t) = \Pr[(m,n) \in_R \mathcal{A} : m \neq n, h(m) = h(n)]$, *where* $P_r[X]$ *refers to the probability of a random event X occurring, and* $(m,n) \in_R \mathcal{A}$ *means that both input strings m and n are randomly selected by* $\mathcal{A}$*. If an* (θ, t)*-adversary* $\mathcal{A}$ *attempts to attack the collision resistance of* $h(\cdot)$*, it means that the maximum execution time of* $\mathcal{A}$ *is t and that* $\mathrm{Ad}\,v_{(A)}^{\mathrm{HASH}}(t) \leq \theta$.

3.2. Fuzzy Extractor for Biometric Verification

The secret value in an encryption mechanism is a random string that requires uniform distribution and can be copied exactly. However, in the real world, it is difficult for the secret value to satisfy this. For example, biometric features, such as fingerprints, brain prints, etc., cannot be accurately copied due to a non-uniform distribution of random values. Thus, we select the fuzzy extraction method for the collection of biometric features [31].

Recently, the fuzzy extractor method has been widely used to extract biometric keys from user biometric input. This method can allow the input to have a certain amount of noise (or error), and as long as the input is similar, the same uniform random string can be extracted. The general structure is as follows.

(1) Gen: Given that the user inputs biometrics BIO_i, the gen process will generate a biometric key r_i of l bits and the corresponding auxiliary public parameter p_i; that is, $Gen(BIO_i) = (r_i, p_i)$.

(2) Rep: Given a noisy user input biometric BIO_i', Rep will return the original biometric key r_i with the help of the auxiliary public data p_i when the Hamming distance between the current biometric input BIO_i' and the original biometric input BIO_i is less than a specific error tolerance threshold t; that is, $HamDis(BIO_i', BIO_i) \leq t$. Thus, $Rep(BIO_i', p_i) = (r_i)$.

Considering the false-positive and false-negative events of biometric authentication, we make a note of BIO_i and BIO_i'. If both BIO_i and BIO_i' originate from the same person, then the Hamming distance between the two will converge to 0. We assume that $Pr[HamDis(BIO_i', BIO_i) \leq t] \geq 1 - \lambda n$, where λn means the false negative probability. If BIO_i and BIO_i' originate from different people, then the Hamming distance between the two may be significant. We assume that $Pr[HamDis(BIO_1, BIO_2) \geq t'] \geq 1 - \lambda p$, $t' \gg t$, where λp means the false positive probability.

4. System Model

4.1. Authentication Model

The IoMT-based authentication model is shown in Figure 1. In this model, patients suffering from different diseases are being treated in the hospital. Each hospital bed is equipped with a number of sensors to monitor and sense the real-time status of the patient (e.g., blood pressure, heart rate, etc.). Since the hospital contains different departments, such as brain, orthopedics, etc., as well as different types of medical staff in each department, such as doctors and nurses, they are all concerned with monitoring the patient's physical condition. Specifically, only the nurse is required to handle a patient who needs a medication change, while the doctor is required to take quick emergency measures when the patient is in a life-threatening situation. Therefore, it is necessary to set the corresponding accessible sensor cluster for different user levels.

Four different departments C_1, C_2, C_3, and C_4 exist in the hospital, as shown on the left side of Figure 1, and some sensor devices are deployed in them. For example, in C_1, seven sensors $\{D_1, D_2, ..., D_7\}$ are deployed to detect real-time data of patients, where $\{D_1, D_2, D_3, D_4\}$ represents the accessible sensor cluster by a particular member of the medical staff U_1. Before authentication, both U_1 and D_j need to complete registration with the help of GW, where U_1 also sends a sensor cluster to GW. Then, U_1 can authenticate with D_j through GW. Once authenticated, U_1 can securely access the real-time data from D_j. Specifically, U_1 first sends a login request to GW. Then, GW validates the login request and sends the access request to the accessible D_j. Finally, once the authentication is complete, D_j sends a reply message to U_1 and generates a session key shared between the two. It is worth noting that the registration phase of 3ECAP is performed in a secure environment, whereas information is transmitted via a public channel in the authentication phase, which makes it vulnerable to anonymous attackers.

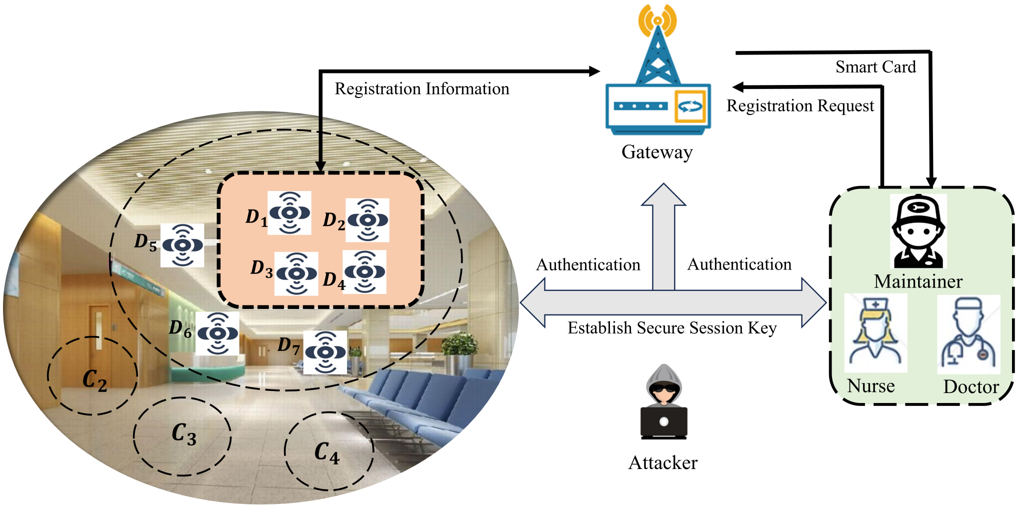

Figure 1. Authentication model for IoMT.

4.2. Threat Model

The protocol we designed uses the Dolev–Yao [32] threat model (DY model) for security analysis, where an adversary can not only intercept messages transmitted between participants but also perform deletion and modification operations. In addition, we consider the widely accepted RoR model [33], which is used to secure the session key generated by medical staff and sensors. Note that in the authentication model, suppose that the GW is fully trusted and is deployed in a fixed location that is physically protected so that the likelihood of the GW being captured is extremely low compared to that of the sensor device. In contrast, for some physically captured sensor devices, the corresponding secret information stored in these devices can be extracted by the adversary using power analysis attacks.

5. Proposed Scheme

In this section, we elaborate on a new protocol called 3ECAP for IoMT deployments. The protocol requires the following phases: (1) setup; (2) medical staff registration; (3) sensor registration; (4) login and authentication; (5) password and biometric update; and (6) new smart-device addition phase. In the setup phase, the public parameters of the protocol are selected by the fully trusted GW. Once the setup is complete, the medical staff and the sensor need to complete the registration in the system. In the login and authentication phase, a user (i.e., legal medical staff) U_i and a sensor device SD_j, with the help of the GW, establish a shared key between U_i and SD_j for future communication. The proposed protocol also enables U_i to change the password and biometric information without the need for GW. In addition, the protocol can support the addition of new sensor devices. The notations and their abbreviations are presented in Table 2 [30] for the analysis of 3ECAP.

5.1. Setup Phase

During the system setup phase, some public parameters are initialized by GW. Specifically, GW chooses a one-way hash function $h(\cdot)$, a biometric key generation function $Gen(\cdot)$ and a biometric key replication function $Rep(\cdot)$, where $Gen(\cdot)$ and $Rep(\cdot)$ are used for bio-information extraction and recovery of medical staff, respectively. Then, GW generates a unique master key x, an identity ID_{GW}, and also calculates the corresponding pseudo-identity $RID_{GW} = h(ID_{GW} \| x)$.

Table 2. Notations and abbreviations.

Noation	Description
U_i, GW, SD_j	i_{th} user, j_{th} sensor and gateway
ID_i, ID_{GW}, SID_j	Identities of U_i, GW and SD_j
RID_i, RID_{GW}, $RSID_j$	Pseudo-identities of U_i, GW, and SD_j
SC_i, BIO_i	Smart card and biometrics of user
AL	User's access list
$Gen(\cdot)$, $Rep(\cdot)$	Functions of the fuzzy extractor
r_i, p_i	Secret parameter and public parameter of U_i
$HamDis(BIO_i', BIO_i)$	Hamming distance between BIO_i' and BIO_i
t	Fault tolerance threshold applied in $Rep(\cdot)$
λ_n, λ_p	False negative probability and false positive probability
$h(\cdot)$	One-way collision-resistant hash function
$\oplus$, $\|$	Bitwise XOR and concatenation operations
T_1, T_2, T_3	Current timestamps
ΔT	Maximum transmission delay
α_j, β_i	Random numbers applied in the registration phase
a, b, c	Random numbers applied in the login and authentication phase
x	Master key for GW
k_{GWj}, k_{jGW}	Shared keys for GW and SD_j
$\mathcal{A}$	Adversary
$P \rightarrow Q : M$	P sends the message M to Q

5.2. Sensor Addition Phase

During the sensor addition phase, GW generates a unique SID_j for the medical sensor SD_j, a random number α_j, and then calculates the pseudo-identity $RSID_j = h(SID_j \| ID_{GW} \| \alpha_j)$. In addition, a secret pairwise key is established between GW and SD_j by means of the master key x of GW, where $k_{GWj} = h(ID_{GW} \| SID_j \| x)$, which will be used for mutual authentication and message encryption between nodes in the subsequent login phase. Finally, GW stores $\{RSID_j, k_{GWj}\}$ into the database. Meanwhile, SID_j also saves $\{RSID_j, k_{GWj}\}$ into the memory.

5.3. Medical Staff Registration Phase

In general, there are many disease departments in the medical system, such as neurology, orthopedics, brain and cardiovascular, etc. Each department is composed of many medical sensor devices that contain sensitive patient information. Therefore, in order to protect patient privacy, medical staff can only access patient information based on access permission for a specific cluster of sensor devices. The registration process for medical staff can also be divided into two phases, as follows.

(1) Fine-Grained Access Control: The purpose of fine-grained access control [30] is to restrict the access permission of the medical staff. For example, medical staff in a neurology department can only access information from sensors relevant to their department, where these sensors are connected to the patient to monitor individual status.

Our sensor cluster model can be simplified to a merkle tree, which consists of multiple leaf nodes $\{D_1, D_2, ..., D_n\}$ and a single root node Ver_i, where $\{D_1, D_2, ..., D_n\}$ represents the sensor device nodes accessible to a particular medical staff. Assume that the number of leaf nodes $n = 2^m$ for $m \geq 1$, Ver_i can be computed as follows.

Procedure 1: Denote the leaf nodes $D_1, D_2, ..., D_n$ as $H_{(log_2 n)(0)}, H_{(log_2 n)(1)}, ..., H_{(log_2 n)(n-1)}$, respectively.

Procedure 2: $Ver_i = H_{00} = h(H_{10} \| H_{11})$, where $H_{xy} = h(H_{(x+1)(2y)} \| H_{(x+1)(2y+1)})$ for $x = 0, 1, 2, ..., (\log_2 n) - 1$ and $y = 0, 1, 2, ..., n - 1$.

Ver_i can also be calculated using auxiliary and leaf nodes, which can effectively reduce the computational complexity. As shown in Figure 2 [30], $Ver_i = h(h(H_{21} \| H_{20}) \| H_{11})$, utilizing auxiliary nodes H_{20} and H_{11}, instead of H_{22} and H_{23}.

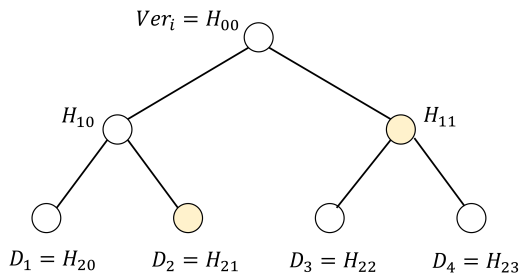

Figure 2. Merkle tree-based access list.

(2) *Personal Information Registration*

Step 1: U_i chooses his or her identity ID_i, password PW_i, access list (generated from the cluster of accessible sensors) $AL = \{RSID_1, RSID_2, ..., RSID_j\}$ and imprints biological information BIO_i on the specific acquisition device. The device then extracts the secret parameter r_i and the public parameter p_i with the help of the generating function $Gen(\cdot)$, namely $Gen(BIO_i) = (r_i, p_i)$. Next, U_i computes $RPW_i = h(PW_i \| r_i)$ and sends $\{ID_i, RPW_i, AL\}$ to GW via a secure channel.

Step 2: After receiving the message $\{ID_i, PW_i, AL\}$, GW computes Ver_i using the above *markle* tree and auxiliary nodes and also computes the personal information $P_i = h(ID_i \| RPW_i)$. Moreover, GW generates the current registration timestamp T_{re}, a random number β_i, and computes the pseudo-identity $RID_i = h(ID_i \| ID_{GW} \| \beta_i)$ and the sensor device list $SDL_i = h(Ver_i \| T_{re} \| RID_i)$. Next, GW stores $\{RID_i, SDL_i\}$ in its memory. GW also returns the message $\{P_i, RID_i, SDL_i, RID_{GW}\}$ to U_i via a secure channel.

Step 3: Once the message is received from GW, U_i calculates $HP_i = h(ID_i \| PW_i \| r_i)$, $\beta_i^* = \beta_i \oplus HP_i$, $SDL_i^* = SDL_i \oplus h(HP_i \| \beta_i)$. Finally, U_i stores the verifiable information $\{P_i, \beta_i^*, SDL_i^* RID_{GW}, RID_i, p_i\}$ into its own smart card SC_i; note that SDL_i^* represents the cluster of sensor devices accessible to the particular user. Figure 3 illustrates the complete process of 3ECAP registration.

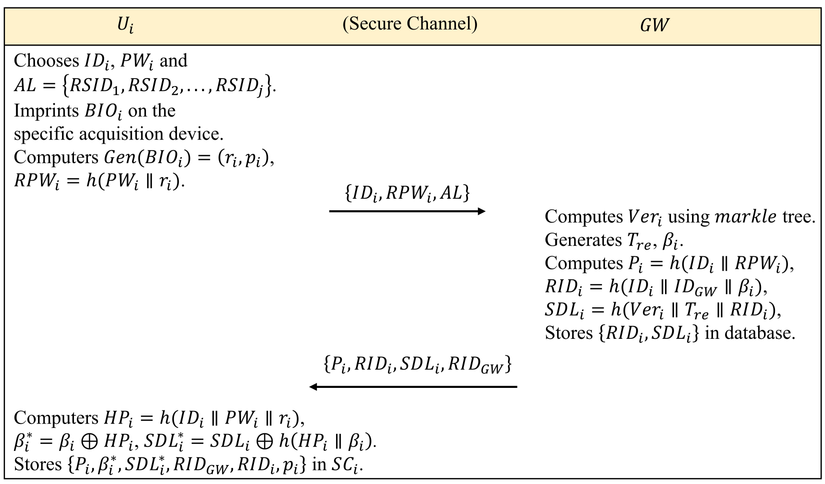

Figure 3. Summary of medical staff registration phase.

5.4. Login and Authentication Phase

When U_i wants to access the data of SD_j, he/she needs to login and authenticate to the GW first. After the authentication process is complete, a secure session key is established between U_i and SD_j for subsequent communication. The following steps are essential under the proposed protocol.

Step 1: $U_i \rightarrow GW : \{A_i, B_i, C_i, RSID_j, T_1\}$

Step 1.1: U_i inputs ID_i', PW_i', and imprints BIO_i' at a biometric acquisition device. SC_i then extracts the public parameter p_i and recovers $r_i = Rep(BIO_i', p_i)$ if $HamDis(BIO_i', BIO_i) \leq t$

is satisfied. Next, SC_i computes $P_i' = h(ID_i' \| h(PW_i' \| r_i))$ and verifies $P_i' \stackrel{?}{=} P_i$. The login request is terminated if $P_i' \neq P_i$.

Step 1.2: SC_i then generates the current timestamp T_1, a random number a, and calculates $HP_i = h(ID_i' \| PW_i' \| r_i)$, $\beta_i' = \beta_i^* \oplus HP_i$, $SDL_i' = SDL_i^* \oplus h(HP_i \| \beta_i')$, $A_i = RID_i \oplus h(RID_{GW} \| T_1)$, $b_i = h(RID_i \| T_1 \| a)$, $B_i = h(RID_i \| RID_{GW} \| T_1) \oplus b_i$, $C_i = h(b_i \| SDL_i' \| RID_{GW} \| RSID_j \| T_1)$. Finally, SC_i sends the message $M_1 = \{A_i, B_i, C_i, RSID_j, T_1\}$ to GW via a common channel, where $RSID_j$ contains the information U_i wants to obtain.

Step 2: $GW \rightarrow SD_j : \{D_i, E_i, F_i, T_2\}$

Step 2.1: Once M_1 is received from U_i, GW first verifies the validity of T_1 under the condition of $|T_1^* - T_1| \leq \Delta T$, where T_1^* is the receive timestamp, and ΔT is the maximum time delay. The entire session is aborted if the condition is not met. Otherwise, GW computes $RID_i = A_i \oplus h(RID_{GW} \| T_1)$ and finds the corresponding SDL_i from the memory. Meanwhile, GW also calculates $b_i = B_i \oplus h(RID_i \| RID_{GW} \| T_1)$, $C_i' = h(b_i \| SDL_i \| RID_{GW} \| RSID_j \| T_1)$ and verifies $C_i' \stackrel{?}{=} C_i$. If $C_i' \neq C_i$, it indicates two possibilities, *case 1:* U_i is an external attacker who does not have the key for the registration phase of the personnel information, and *case 2:* U_i is an internal attacker who wants to access sensors beyond his/her own permission, i.e., the sensor cluster $SDL_i' \neq SDL_i$.

Step 2.2: If $C_i' = C_i$, it indicates that the identity of U_i is confirmed. Then, GW generates the current timestamp T_2, a random number b and computes $D_i = b \oplus h(k_{GWj} \| T_2)$, $E_i = b_i \oplus h(b \| T_2)$, $F_i = h(RSID_j \| k_{GWj} \| b_i \| b \| T_2)$, where k_{GWj} is the symmetric key for SD_j and is stored in the memory of GW. At last, GW sends the message $M_2 = \{D_i, E_i, F_i, T_2\}$ publicly to SD_j.

Step 3: $SD_j \rightarrow U_i : \{G_i, H_i, J_i, T_3\}$

Step 3.1: Once SD_j receives message M_2 from GW, SD_j verifies that timestamp T_2 matches condition $|T_2^* - T_2| \leq \Delta T$. If the condition does not match, it indicates that the timeliness of M_2 is not guaranteed and the session will be closed. Otherwise, SD_j calculates $b = D_i \oplus h(k_{GWj} \| T_2)$, $b_i = E_i \oplus h(b \| T_2)$ and $F_i' = h(RSID_j \| k_{GWj} \| b_i \| b \| T_2)$ using the stored symmetric key k_{GWj}. SD_j then verifies that $F_i' \stackrel{?}{=} F_i$.

Step 3.2: If $F_i' \neq F_i$, the access request from GW is terminated. Otherwise, SD_j authenticates GW successfully. Then, SD_j generates the current timestamp T_3, a random number c and calculates $G_i = c \oplus b_i$, $H_i = c \oplus h(k_{GWj} \| T_2) = c \oplus d_i$, $J_i = h(RSID_j \| d_i \| c \| T_3)$. Meanwhile, SD_j also computes the secure session key $sk_{ji} = h(b_i \| d_i \| RSID_j \| c \| T_3)$. Finally, SD_j sends the message $M_3 = \{G_i, H_i, J_i, T_3\}$ to U_i via a public channel.

Step 4: Once M_3 is received at time T_3^* by U_i, SC_i verifies the validity of T_3 in this message with the condition of $|T_3^* - T_3| \leq \Delta T$. If the condition fails, the session is immediately terminated by U_i. Otherwise, SC_i calculates $c = G_i \oplus b_i$, $d_i = c \oplus H_i$ and $J_i' = h(RSID_j \| d_i \| c \| T_3)$ and verifies $J_i' \stackrel{?}{=} J_i$. If $T_i' = J_i$, it means that the identity of SD_j is confirmed. Eventually, SC_i computes the session key $sk_{ij} = h(b_i \| d_i \| RSID_j \| c \| T_3) (= sk_{ji})$ shared with SD_j, which will be used to encrypt the data transmitted between U_i and SD_j. Figure 4 illustrates the complete process of 3ECAP login and authentication.

5.5. Password and Bio-Information Update Phase

Usually, human biological characteristics change over time, for example, the characteristics of brain waves are completely different at different ages. Therefore, 3ECAP supports the modification of biological information for medical staff. In addition, we recommend that medical staff change their passwords regularly to ensure the security of their privacy (this part was not considered in the previous conference). The specific steps for password and bio-information modification are as follows.

Step 1: U_i input ID_i and PW_i^{old} and imprint the old bio-information BIO_i^{old} on the specific collection device. Meanwhile, U_i inserts its smart card SC_i in the system terminal. Then, SC_i computes $r_i^{old} = Rep(BIO_i^{old}, p_i)$ with the condition $HamDis(BIO_i^{old}, BIO_i) \leq t$, where BIO_i is the biological information previously registered by U_i. Next, SC_i computes $P_i^{old} = h(ID_i \| h(PW_i^{old} \| r_i^{old}))$ and verifies $P_i^{old} \stackrel{?}{=} P_i$. If $P_i^{old} = P_i$, SC_i authenticates U_i

successfully. Otherwise, password and bio-information change requests are terminated by SC_i.

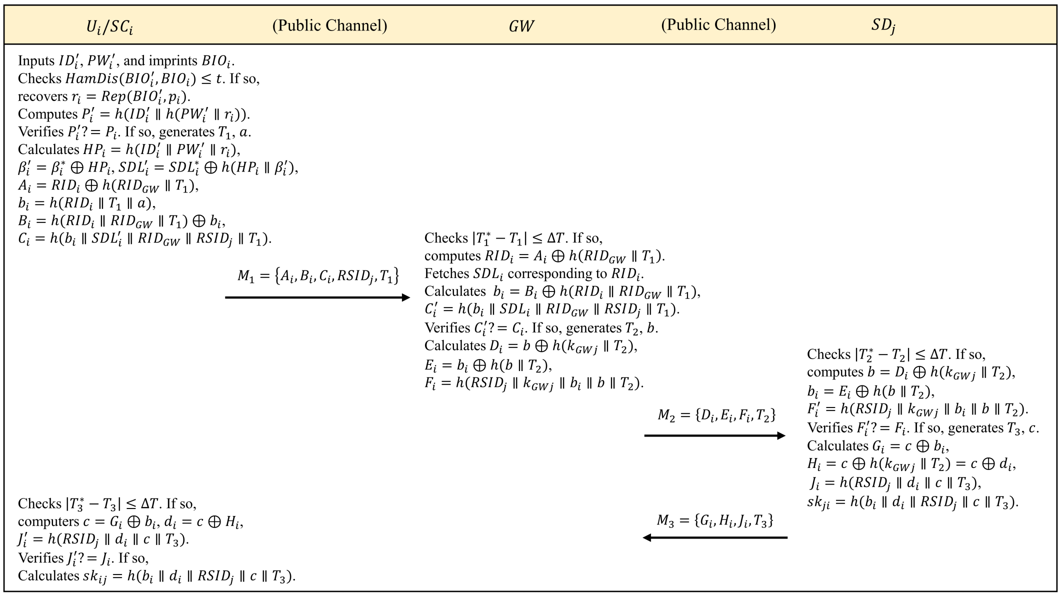

Figure 4. Summary of login and authentication phase.

Step 2: After successful authentication, U_i enters a new password PW_i^{new} and imprints the new bio-information BIO_i^{new} at the acquisition device. The device then extracts the corresponding secret parameter r_i^{new} and public parameter p_i^{new} using $Gen(\cdot)$. Next, SC_i calculates the old secret information $HP_i^{old} = h(ID_i \| PW_i^{old} \| r_i^{old})$, $\beta_i = \beta_i^* \oplus HP_i^{old}$ and $SDL_i = SDL_i^* \oplus h(HP_i^{old} \| \beta_i)$. SC_i also calculates the new secret information $P_i^{new} = h(ID_i \| h(PW_i^{new} \| r_i^{new}))$, $HP_i^{new} = h(ID_i \| PW_i^{new} \| r_i^{new})$, $\beta_i^{new*} = \beta_i \oplus HP_i^{new}$, $SDL_i^{new*} = SDL_i \oplus h(HP_i^{new} \| \beta_i)$. SC_i finally replaces $\{P_i, \beta_i^*, SDL_i^*, p_i\}$ with $\{P_i^{new}, \beta_i^{new*}, SDL_i^{new*}, p_i^{new}\}$ in its memory.

5.6. New Smart Device Addition Phase

Usually, a sensor is installed in each sickbed to capture the real-time status of the patient (e.g., blood pressure, temperature, heartbeat, etc.). Hence, the number of sensors is generally fixed. However, when emergencies arise (for example, the outbreak of COVID-19), the original number of beds cannot meet the demand of patients. Therefore, 3ECAP can support the bulk addition of new sensors with the following steps (this part was not considered in the previous conference).

Step 1: Before new sensors are deployed, they need to register with the gateway. Specifically, GW selects an identity SID_j^{new} and a random number α_j^{new} for SD_j^{new} and computes the pseudo-identity $RSID_j^{new} = h(SID_j^{new} \| ID_{GW} \| \alpha_j^{new})$. Meanwhile, GW computes $k_{GWj}^{new} = h(ID_{GW} \| SID_j^{new} \| x)$. Then, GW and SD_j^{new} store $\left\{SID_j^{new}, RSID_j^{new}, k_{GWj}^{new}\right\}$ and $\left\{RSID_j^{new}, k_{GWj}^{new}\right\}$ into their own databases, respectively. Furthermore, after the registration of the sensor is complete, GW needs to broadcast the addition regarding SD_j^{new} so that U_i can access the data therein.

Step 2: U_i needs to update sensor device list SDL_i with the help of GW before accessing the bulk-added $\left\{RSID_1^{new}, RSID_2^{new}, \ldots, RSID_j^{new}\right\}$. U_i first inputs ID_i', PW_i', BIO_i' and inserts SC_i. Then, SC_i calculates $r_i = Rep(BIO_i', p_i)$ if condition $HamDis(BIO_i', BIO_i) \leq t$ is satisfied. Next, SC_i computes $P_i' = h(ID_i' \| h(PW_i' \| r_i))$ and verifies $P_i' \stackrel{?}{=} P_i$. If $P_i' = P_i$, SC_i sends $\{RID_i, AL\}$ to GW via a secure channel, where AL consists of the old devices $\{RSID_1, RSID_2, \ldots, RSID_j\}$ and the newly added devices $\left\{RSID_1^{new}, RSID_2^{new}, \ldots, RSID_j^{new}\right\}$.

Step 3: Once the message $\{RID_i, AL\}$ is received from U_i, GW generates a new current registration timestamp T_{re}^{new} and computes $SDL_i^{new} = h(Ver_i^{new} \| T_{re}^{new} \| RID_i)$, where Ver_i^{new} is calculated based on AL using the *merkle* tree. Then, GW sends SDL_i^{new} to U_i via a secure channel.

Step 4: When SDL_i^{new} is received from GW, SC_i computes $HP_i = h(ID_i' \| PW_i' \| r_i)$, $\beta_i = \beta_i^* \oplus HP_i$, $SDL_i^{new*} = SDL_i^{new} \oplus hHP_i \| \beta_i)$. Finally, SC_i replaces SDL_i^* with SDL_i^{new*} in its memory.

6. Security Analysis

In this section, we verify the security reliability of 3ECAP using both formal and informal security analysis. Specifically, we first prove the security of session keys in the proposed protocol based on the ROR model. Then, we use informal security analysis to demonstrate that 3ECAP is secure in the face of access privilege escalation as well as other known attacks. In addition, we perform formal security verification using the popular automated verification tool Proverif.

6.1. ROR Model-Based Formal Security Analysis

We consider random oracles under the formal security model, where the adversary/attacker $\mathcal{A}$ can make multiple oracle queries (this part was not considered in the previous conference).

(1) ROR model:

In the login and authentication phase of 3ECAP, three participants U_i, GW and SD_j are involved in this process. The model considers the following.

Participants: The instances i, k, and j corresponding to the participants U_i, GW, and SD_j can be denoted as $\omega_{U_i}^i$, ω_{GW}^k, and $\omega_{SD_j}^j$, respectively, which are called oracles.

Accepted state: An instance ω^i is in the accept state, indicating that it has received the last message. Once the messages sent and received by ω^i are sequentially ordered, it forms the session identifier side of ω^i for the running session.

Partnering: Two instances, called ω^i and ω^j, are partners if the following conditions are met: (1) ω^i and ω^j are in the accepted state; (2) ω^i and ω^j have the same session identification (sid), i.e., $sid_\omega^i = sid_\omega^j$; and (3) ω^i's partner identification (pid) is ω^j and vice versa.

Freshness: Two instances, called ω^i and ω^j, are fresh if the key sk_{ij} $(= sk_{ji})$ established between U_i and SD_j is not disclosed by adversary $\mathcal{A}$ through reveal query.

Adversary: Since the ROR model is based on the DY threat model, adversary $\mathcal{A}$ can fully control all messages transmitted in the network, which means that $\mathcal{A}$ can eavesdrop, modify, delete, forge, or inject messages between two entities.

Execute (ω^i,ω^k,ω^j): The passive attack is modeled under this query, which allows $\mathcal{A}$ to intercept all communication records between participants U_i, GW, and SD_j.

Send (ω^i,m): This query is considered an active attack, where $\mathcal{A}$ can send a message m to an instance ω^i and also receive a response message.

Reveal (ω^i): When this query is executed, the session key sk_{ij} (=sk_{ji}) established between ω^i and its partner is leaked to $\mathcal{A}$.

CorruptSC ($\omega_{U_i}^i$): Once such a query is executed, the information stored in the smart card SC_i of U_i is disclosed to $\mathcal{A}$.

CorruptSD ($\omega_{SD_j}^j$): Under this query, $\mathcal{A}$ can extract all the sensitive information stored in a sensor by a power analysis attack. Therefore, this query is modeled as an active attack. In addition, we also assume that both $CorruptSC$ and $CorruptSD$ provide a weak corruption model where the temporary keys and internal data of the instance are not corrupted.

Test (ω^i): The semantic security of the session key sk (i.e., sk_{ij} or sk_{ji}) established between instances can be modeled with this query. Once this query is executed, a coin c is tossed and the result is returned to $\mathcal{A}$. If $c = 1$, the instance returns sk or a random number of the same length as sk if $c = 0$; otherwise, it returns a null value.

It is worth noting that, according to [34], we perform a limit on the number of queries for *CorruptSC* and *CorruptSD* queries. However, $\mathcal{A}$ is allowed to execute multiple *Test* queries. Furthermore, since GW is absolutely secure in the network, $\mathcal{A}$ cannot obtain any information from GW by *Corrupt* query. All participants and $\mathcal{A}$ have access to a one-way collision-resistant hash function $h(\cdot)$, which is modeled as a random oracle.

(2) Security Proof: The semantic security (or AKE security) of the session key SK in 3ECAP is given in Theorem 1. Furthermore, similar proofs [35] and [34] follow Theorem 1.

Theorem 1. *If $\mathcal{A}$ is the adversary in polynomial time against 3ECAP in the RoR model, and q_h, q_s, and q_e denote the number of Hash queries, Send queries, and Execute queries, respectively, then*

$$Adv_{3ECAP,\mathcal{D}}^{\mathrm{AKE}} \leq \frac{q_h^2}{2^{l_h}} + \frac{(q_s+q_e)^2}{2^{l_r}} + 2\max\left(C'q_s^{s'}, \frac{q_s}{2^{l_b}}, \lambda_p q_s\right)$$

where l_h, l_r, and l_b refer to the length of the hash output, the length of the random number, and the length of the user bio-secret parameter r_i, respectively. $\mathcal{D}$ is denoted as the password space and obeys the Zipf distribution, and C' and s' are the parameters of Zipf.

Proof. The security proof of the proposed protocol (3ECAP) is composed of a series of games: G_0, G_1, G_2, G_3. Suppose $Succ_{\mathcal{A}}^{G_j}$ $(j = 0,1,2,3)$ represents an event in which $\mathcal{A}$ successfully guesses the random bit c of a tossed coin in the game G_j and the corresponding probability of occurrence is denoted as $Pr[Succ_j]$. □

Game G_0: This is the initial game where $\mathcal{A}$ performs a real attack simulation on 3ECAP in the ROR model. Thus, according to the definition of semantic security, we have

$$Adv_{3ECAP,\mathcal{D}}^{\mathrm{AKE}} = |2Pr[Succ_0] - 1|. \tag{1}$$

Game G_1: It corresponds to a passive attack implemented by $\mathcal{A}$, where $\mathcal{A}$ can perform an *Execute* query and intercept all messages $M_1 = \{A_i, B_i, C_i, RSID_j, T_1\}$, $M_2 = \{D_i, E_i, F_i, T_2\}$ and $M_3 = \{G_i, H_i, J_i, T_3\}$ transmitted in the public channel during the login and authentication phases of U_i. Once the game is over, $\mathcal{A}$ executes a *Test* query and discriminates the genuine sk from a random number based on the results returned by the query, where $sk = h(b_i \| d_i \| RSID_j \| c \| T_3)$, $b_i = h(RID_i \| T_1 \| a)$ and $d_i = h(k_{GWj} \| T_2)$. Therefore, $\mathcal{A}$ needs the secret information RID_i, k_{GWj} and a to calculate the session key sk. However, this secret information cannot be obtained by $\mathcal{A}$ by eavesdropping on messages M_1, M_2 and M_3. Therefore, the probability of adversary $\mathcal{A}$ winning the game G_1 does not increase. Due to the indistinguishability of games G_0 and G_1, we have

$$Pr[Succ_1] = Pr[Succ_0]. \tag{2}$$

Game G_2: Game G_2 is modeled as an active attack where the primary goal of $\mathcal{A}$ is to attempt to convince participating nodes that the forged message is legitimate. Suppose that $\mathcal{A}$ performs q_h number of Hash queries with the help of q_s number of the *Send* queries. Based on the results of the birthday paradox, the collision probability of the Hash query is at most $\frac{q_h^2}{2^{l_h}}$. Since the random numbers a, b and c exist in messages M_1, M_2 and M_3, respectively, the collision probability of the random numbers is at most $\frac{(q_s+q_e)^2}{2^{l_r}}$. Hence, we obtain

$$|Pr[Succ_2] - Pr[Succ_1]| \leq \frac{q_h^2}{2^{l_h}} + \frac{(q_s+q_e)^2}{2^{l_r}}. \tag{3}$$

Game G_3: This is the last game, where $\mathcal{A}$ executes *CorruptSC* and *CorruptSD* queries. Specifically, the information $\{P_i, \beta_i^*, SDL_i^*, RID_{GW}, RID_i, p_i\}$ stored in SC_i and the information $\{RSID_j, k_{GWj}\}$ stored in SD_j are obtained by $\mathcal{A}$ using *CorruptSC* and *CorruptSD*, respectively. Note that the pseudo-identity $RSID_j$ and k_{GWj} of all sensors are different from each other. In 3ECAP, U_i uses both password PW_i and bio-information BIO_i for authentication, which can be divided into two cases.

Case 1: Suppose that $\mathcal{A}$ attempts to guess the low entropy password using q_s number of the send queries. Since the user's password follows Zipf's law [36,37], the probability of this case is $C'q_s^{s'}$.

Case 2: Assume that $\mathcal{A}$ tries to extract the biological key r_i of U_i from the obtained information. Since 3ECAP adopts the fuzzy extractor technique, $\mathcal{A}$ can only extract at most l_b random bits, and the corresponding probability of guessing r_i is approximately 2^{-l_b}. In addition, we consider the probability of false positive λ_p that occurs for biometric feature extraction. In general, for fingerprints, $\lambda_p \approx 2^{-14}$ [34].

Therefore, based on case 1 and case 2, it follows that

$$|Pr[Succ_3] - Pr[Succ_2]| \leq \max\left(C'q_s^{s'}, \frac{q_s}{2^{l_b}}, \lambda_p q_s\right). \tag{4}$$

Since all queries are executed, $\mathcal{A}$ can only win the game by guessing bit c. This means that

$$Pr[Succ_3] = \frac{1}{2}. \tag{5}$$

From (1) and (2), it is given that

$$\frac{1}{2} Adv^{\text{AKE}}_{3ECAP,\mathcal{D}} = \left| Pr[Succ_1] - \frac{1}{2} \right|. \tag{6}$$

From (5) and (6), we have

$$\frac{1}{2} Adv^{\text{AKE}}_{3ECAP,\mathcal{D}} = |Pr[Succ_1] - Pr[Succ_3]|. \tag{7}$$

Using the trigonometric inequality, we can obtain

$$\begin{aligned} |Pr[Succ_1] - Pr[Succ_3]| \leq\ & |Pr[Succ_1] - Pr[Succ_2]| \\ & + |Pr[Succ_2] - Pr[Succ_3]|. \end{aligned} \tag{8}$$

Finally, from (3), (4), (7) and (8), we have

$$Adv^{\text{AKE}}_{3ECAP,\mathcal{D}} \leq \frac{q_h^2}{2^{l_h}} + \frac{(q_s + q_e)^2}{2^{l_r}} + 2\max\left(C'q_s^{s'}, \frac{q_s}{2^{l_b}}, \lambda_p q_s\right).$$

6.2. Informal Security Analysis

(1) Medical Staff Impersonation Attack: The adversary/attacker $\mathcal{A}$ who attempts to impersonate a legitimate medical staff needs to create a valid message $M_1 = \{A_i, B_i, C_i, RSID_j, T_1\}$, where $A_i = RID_i \oplus h(RID_{GW} \| T_1)$, $B_i = h(RID_i \| RID_{GW} \| T_1) \oplus b_i$, $C_i = h(b_i \| SDL_i \| RID_{GW} \| RSID_j \| T_1)$. Even if $\mathcal{A}$ can generate the timestamp T_1' and the random number a', $\mathcal{A}$ cannot recover M_1 due to the lack of the key secret information RID_i, RID_{GW}, b_i and SDL_i'. This indicates that 3ECAP is secure against a user impersonation attack.

(2) Gateway Impersonation Attack: In order to become a legitimate node by impersonating GW, adversary $\mathcal{A}$ needs to create a message $M_2 = \{D_i, E_i, F_i, T_2\}$ to send to SD_j, where $D_i = b \oplus h(k_{GWj} \| T_2)$, $E_i = b_i \oplus h(b \| T_2)$, $F_i = h(RSID_j \| k_{GWj} \| b_i \| b \| T_2)$. Even if $\mathcal{A}$ can generate the timestamp T_2' and the random number b', $\mathcal{A}$ will be unable to recover M_2 as the calculations of $\{D_i, E_i, F_i\}$ need the secret information k_{GWj}, b_i and $RSID_j$. Thus, 3ECAP is protected in a gateway impersonation attack.

(3) Sensor Impersonation Attack: Suppose that $\mathcal{A}$ attempts to generate a message $M_3 = \{G_i, H_i, J_i, T_3\}$ on behalf of SD_j to become a legitimate device node, where $G_i = c \oplus b_i$, $H_i = c \oplus d_i$, $J_i = h(RSID_j \| d_i \| c \| T_3)$. Although $\mathcal{A}$ can generate timestamp T_3' and random number c' due to the absence of secret information b_i and d_i, $\mathcal{A}$ also cannot recover M_3.

(4) Stolen Verifier Attack: Assume that $\mathcal{A}$ has stolen the medical staff's smart card SC_i and obtains the secret information $\{P_i, \beta_i^*, SDL_i^*, RID_{GW}, RID_i, p_i\}$ stored in SC_i using the power

analysis attack, where $P_i = h(ID_i \| h(PW_i \| r_i))$, $\beta_i^* = \beta_i \oplus HP_i$, $SDL_i^* = SDL_i \oplus h(HP_i \| \beta_i)$, $RID_i = h(ID_i \| ID_{GW} \| \beta_i)$, $RID_{GW} = h(ID_{GW} \| x)$. Suppose $\mathcal{A}$ guesses a password PW_i' and attempts to verify its authenticity using known information. However, verifying PW_i' requires guessing both the identity ID_i and the secret information r_i of U_i, which is computationally difficult to achieve due to the collision-resistant property of $h(\cdot)$ (see Definition 1). Similarly, A cannot guess the bio-information r_i correctly without ID_i and PW_i. Moreover, it is not possible for $\mathcal{A}$ to compute other information, such as β_i and SDL_i, in the absence of HP_i. Hence, 3ECAP is secure against a stolen smart card attack.

(5) Replay Attack: Suppose that adversary $\mathcal{A}$ intercepts messages M_1, M_2 and M_3 in a session and replays them after some time. The replay attack makes the participating nodes unable to recognize the authenticity of the messages and may lead to system breakdown as the number of replayed messages increases. However, due to the presence of timestamp T in M_1, M_2 and M_3, when a node receives a message, the first task for it is to verify the validity of T under the condition $|T^* - T| \le \Delta T$, where T^* represents the reception timestamp. Therefore, 3ECAP is secure against a replay attack.

(6) Denial-of-Service Attack: In the login and authentication phase of medical staff U_i, U_i first inserts the smart card SC_i and imprints his or her bio-information BIO_i' on the acquisition device, and also enters the corresponding identity ID_i' and password PW_i'. If the condition $HamDis(BIO_i', BIO_i) \le t$ is not satisfied, the whole session is terminated. Otherwise, SC_i computes $r_i = Rep(BIO_i', p_i)$, $P_i' = h(ID_i' \| h(PW_i' \| r_i))$, and verifies $P_i' \stackrel{?}{=} P_i$. The session is also aborted if the equation does not hold. Therefore, it is clear that 3ECAP is capable of dealing with denial-of-service attacks.

(7) Sensor Device Capture Attack: Assume that $\mathcal{A}$ has captured SD_j and obtained information $\{RSID_j, k_{GWj}\}$ from it and attempts to compute the session key between U_i and other uncaptured sensors SD_j' based on $\{RSID_j, k_{GWj}\}$, where $RSID_j = h(SID_j \| ID_{GW} \| \alpha_j)$, $k_{GWj} = h(ID_{GW} \| SID_j \| x)$. However, it is difficult for $\mathcal{A}$ to accomplish this task as these calculations require SID_j and α_j, which are randomly generated by GW. Hence, 3ECAP is secure in the face of a sensor device capture attack.

(8) Man-in-the-Middle Attack: In this attack, adversary $\mathcal{A}$ intercepts the messages M_1, M_2 and M_3 in a particular session and attempts to modify them into another form, which can make it impossible for participating nodes, such as U_i, GW, and SD_j, to determine whether they are communicating with a legitimate node. Suppose $\mathcal{A}$ intercepts message $M_1 = \{A_i, B_i, C_i, RSID_j, T_1\}$ and forges a new message M_1' using the information in it, where $A_i = RID_i \oplus h(RID_{GW} \| T_1)$, $B_i = h(RID_i \| RID_{GW} \| T_1) \oplus b_i$, $C_i = h(b_i \| SDL_i' \| RID_{GW} \| RSID_j \| T_1)$. Even if $\mathcal{A}$ has the ability to generate timestamp T_1' and random number a', $\mathcal{A}$ cannot forge message M_1', which can be recognized by participating nodes, due to the fact that these calculations require secret information RID_i, RID_{GW}, b_i and SDL_i'. Similarly, adversary $\mathcal{A}$ cannot forge M_2' and M_3'. Therefore, 3ECAP is safe in responding to a man-in-the-middle attack.

(9) Insider Privilege Attack: There may be a scenario in which a privileged internal personnel of the trusted GW serves as an internal attacker $\mathcal{A}$. This attack can be divided into two cases as follows.

Case 1: Assume that $\mathcal{A}$ obtains RID_i of U_i during the medical staff registration phase, where $RID_i = h(ID_i \| ID_{GW} \| \beta_i)$. Without knowing the identity ID_i of U_i and the random number β_i, it is difficult for A to guess one of them correctly from RID_i due to the collision resistance property of $h(\cdot)$.

Case 2: Suppose that $\mathcal{A}$ intercepts message $\{P_i, RID_i, SDL_i, RID_{GW}\}$ at the time of medical staff registration , which is initially sent by GW to U_i via a secure channel, where $P_i = h(ID_i \| RPW_i)$, $SDL_i = h(Ver_i \| T_{re} \| RID_i)$, $RID_{GW} = h(ID_{GW} \| x)$. However, $\mathcal{A}$ cannot obtain any information from the message, due to the lack of ID_i, RPW_i, Ver_i, T_{re}, ID_{GW}, x and the collision resistance property of $h(\cdot)$. Hence, 3ECAP has the capability to cope with a privileged-insider attack.

(10) Privilege Escalation Attack: In this attack, medical staff U_i, authorized by GW, wants to gain data from other devices, which are out of U_i's access list, by upgrading

his/her access privilege. For this purpose, the access list AL for U_i needs to be changed from $AL = \{RSID_1, RSID_2, ..., RSID_j\}$ to $AL' = \left\{RSID'_1, RSID'_2, ..., RSID'_j\right\}$, such that $AL' \neq AL$ and $Ver'_i = Ver_i$, when U_i's sensor device list is $SDL_i = h(Ver_i \| T_{re} \| RID_i)$. Although U_i gains T_{re}, he or she cannot upgrade AL while keeping SDL_i unchanged, as explained below.

Note: Let $f_j(\cdot)$ be a function for the calculation of the root hash of a *merkle* tree consisting of j leaf nodes. Also let $\{RSID_1, RSID_2, ..., RSID_j\}$ be denoted as $\{D_1, D_2, ..., D_j\}$. Then, we prove that $f_j(\cdot)$ has the property of collision resistance by mathematical induction, same as $h(\cdot)$. In order not to lose generality, assume that $j = 2^m$ for $m \geq 1$. Given $AL = \{D_1, D_2\}$ for $m = 1$, we have

$$f_2(D_1, D_2) = h(H_{10} \| H_{11}) \tag{9}$$

where $H_{10} = D_1$ and $H_{11} = D_2$. It is obvious that $f_2(\cdot)$ is a collision-resistant function, same as $h(\cdot)$. Suppose the same is true when $m = k$, that is, there is no

$$f_{2^k}(D_1, D_2, ..., D_{2^k}) = f_{2^k}(D'_1, D'_2, ..., D'_{2^k}) \tag{10}$$

where $\{D_1, D_2, ..., D_j\} \neq \left\{D'_1, D'_2, ..., D'_j\right\}$. Then, when $m = k + 1$, we have

$$f_{2^{k+1}}(D_1, D_2, ..., D_{2^{k+1}}) = h(H_{10} \| H_{11}) \tag{11}$$

$$H_{10} = f_{2^k}(D_1, D_2, ..., D_{2^k}) \tag{12}$$

$$H_{11} = f_{2^k}(D_{2^k+1}, D_{2^k+2}, ..., D_{2^{k+1}}). \tag{13}$$

Therefore, it follows from (9), (10), (11), (12) and (13) that $f_j(\cdot)$ is as collision resistant as $h(\cdot)$. Let $AL = \{D_1, D_2, ..., D_j\}$ and $AL' = \left\{D'_1, D'_2, ..., D'_j\right\}$, where $j = 2^m$ for $m \geq 1$ and $\{D_1, D_2, ..., D_j\} \neq \left\{D'_1, D'_2, ..., D'_j\right\}$. Next, $Ver_i = f_j(D_1, D_2, ..., D_j)$ and $Ver'_i = f_j(D'_1, D'_2, ..., D'_j)$. Due to the collision-resistant nature of $f_j(\cdot)$, it is not feasible to find AL', where $AL' \neq AL$, such that $Ver_i = Ver'_i$ is satisfied.

Suppose U_i obtains the registration timestamp T_{re} and extracts the pseudo-identity RID_i by power analysis attack, which is stored in SC_i. Then, U_i expands the permissions to $AL' = \left\{D'_1, D'_2, ..., D'_j\right\}$ and computes $Ver'_i = f_j(D'_1, D'_2, ..., D'_j)$ and $SDL'_i = h(Ver'_i \| T_{re} \| RID_i)$. However, in step 2.1 of the authentication phase, GW uses SDL_i stored in the database to compute $C'_i = h(b_i \| SDL_i \| RID_{GW} \| RSID_j \| T_1)$ and verify $C'_i \stackrel{?}{=} C_i$, where C_i belongs to M_1 and is sent by U_i to GW. It is clear that $C'_i \neq C_i$ because of the collision-resistant property of $f_j(\cdot)$ and $h(\cdot)$ such that the whole session is terminated. Thus, 3ECAP is protected against a privilege escalation attack.

(11) Anonymity and Untraceability: In 3ECAP, all messages $M_1 = \{A_i, B_i, C_i, RSID_j, T_1\}$, $M_2 = \{D_i, E_i, F_i, T_2\}$ and $M_3 = \{G_i, H_i, J_i, T_3\}$ of a particular session are set with timestamps T_1, T_2 and T_3, respectively, and also with random numbers a, b and c, which ensure that the participants U_i, GW and SD_j in the session are not tracked by the adversary. Furthermore, 3ECAP uses pseudo-identities RID_i, RID_{GW} and $RSID_j$ to transmit information in the public channel instead of the original identities ID_i, ID_{GW} and SID_j, respectively, of the participating nodes in the session. Therefore, the anonymity of all participants in 3ECAP can be guaranteed.

6.3. Formal Verification with Proverif

Proverif is a formal automatic verification cryptographic protocol tool based on the Dolev–Yao model developed by Bruno Blanchet, which is able to describe various cryptographic primitives such as shared key cryptography and public key cryptography (encryption and digital signatures), Hash functions and Diffie–Hellman key exchange protocols. In addition, Proverif can handle an infinite session concurrent protocol and infinite message space, which overcomes the problem of state space explosion. When applying the Proverif tool to verify a cryptographic protocol, the tool gives a sequence of attacks if the protocol is vulnerable. All details about the usage of Proverif are in [38].

Four different channels, $sch1$, $sch2$, $ch1$ and $ch2$, are defined in Proverif, where $sch1$ and $sch2$ are secure channels for node registration and $ch1$ and $ch2$ are public channels for medical staff login and authentication. In addition, we define three processes for U_i, GW, and SD_j, respectively, and use $process!User|!GW|!Device$ to implement the parallel operation of the three entities.

The results of the Proverif execution are shown in Table 3 [30] and Figure 5. The first two rows demonstrate that both weak ID_i and PW_i can cope with guessing attacks. The last two rows imply that the generated session keys between U_i and SD_j are robust against common attacks. Therefore, 3ECAP is secure under formal verification.

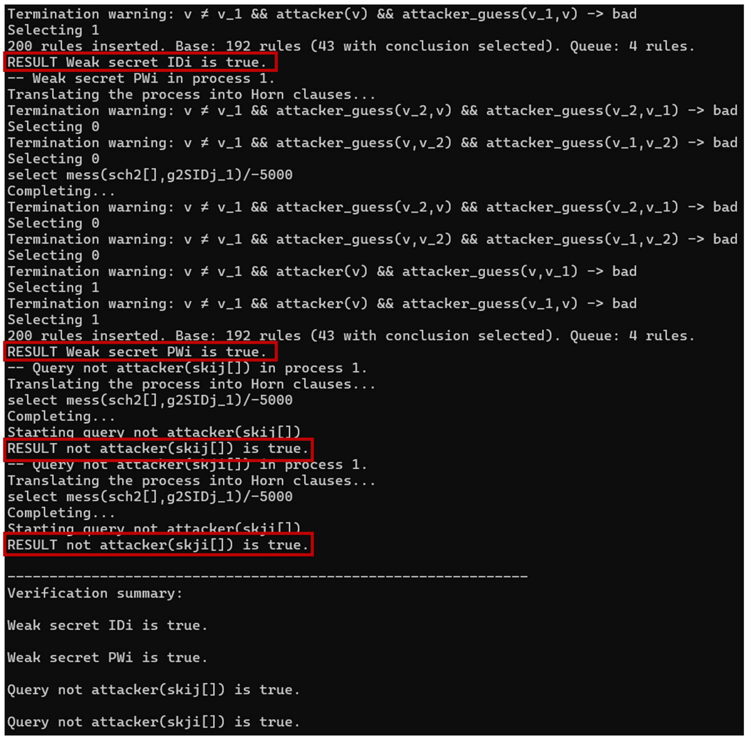

Figure 5. Results of executing Proverif.

Table 3. Results for code.

Secure channel	$sch1$, $sch2$
Public channel	$ch1$, $ch2$
Process	$User$, GW, $Device$
RESULT Weak secret IDi is true (bad not derivable).	
RESULT Weak secret PWi is true (bad not derivable).	
RESULT not attacker(skij[]) is true.	
RESULT not attacker(skji[]) is true.	

7. Comparative Analysis

In this section, a comparative analysis of the calculation cost, communication and security features of 3ECAP and related protocols for Li et al. [26], Xie et al. [27], Son et al. [28] and Yang et al. [29] is shown.

7.1. Calculation Costs Comparison

The calculation costs required for 3ECAP and other protocols in the login and authentication phases are provided in this section. Assume that T_h, T_{as}, T_{bp}, T_{ecc} and T_f represent the time required for the hash function (SHA-256), asymmetric encryption/decryption (RSA-1024), bilinear pairing, ECC point multiplication and fuzzy extractor, respectively.

Based on the available experimental results of Challa et al. [39], the time required to use these functions are $T_h = 0.019$ ms, $T_{as} = 19.536$ ms, $T_{bp} = 44.517$ ms, $T_{ecc} = 2.61$ ms and $T_f = 1.71$ ms. Specifically, the various calculation costs required for the user, gateway, and sensor in each protocol are shown in Table 4 and Figure 6. The calculation cost of 3ECAP for the U_i, GW, and SD_j are, respectively, $T_f + 10T_h$, $6T_h$ and $6T_h$. The total calculation cost of 3ECAP is only 24.14 ms compared to other protocols, which is especially suitable for the communication requirements for IoMT.

Table 4. Calculation costs comparison.

Protocol	User	Gateway	Sensor Device	Total Cost	Rough Estimation
3ECAP	$T_f + 10T_h$	$6T_h$	$6T_h$	$T_f + 22T_h$	24.14 ms
Li et al. [26]	$T_f + 3T_{ecc} + 8T_h$	$T_{ecc} + 8T_h$	$2T_{ecc} + 4T_h$	$T_f + 6T_{ecc} + 20T_h$	33.14 ms
Xie et al. [27]	$12T_h + 5T_{ecc}$	$10T_h + 6T_{ecc}$	$7T_h + 2T_{ecc}$	$29T_h + 13T_{ecc}$	34.481 ms
Son et al. [28]	$15T_h + 3T_{ecc}$	$8T_h + 3T_{ecc} + T_{as}$	$10T_h + 2T_{as}$	$33T_h + 6T_{ecc} + 3T_{as}$	69.159 ms
Yang et al. [29]	$5T_h + 7T_{ecc} + 3T_{bp}$	$2T_h + 2T_{ecc} + 4T_{bp}$	$2T_h + 2T_{ecc} + 3T_{bp}$	$9T_h + 11T_{ecc} + 10T_{bp}$	474.051 ms

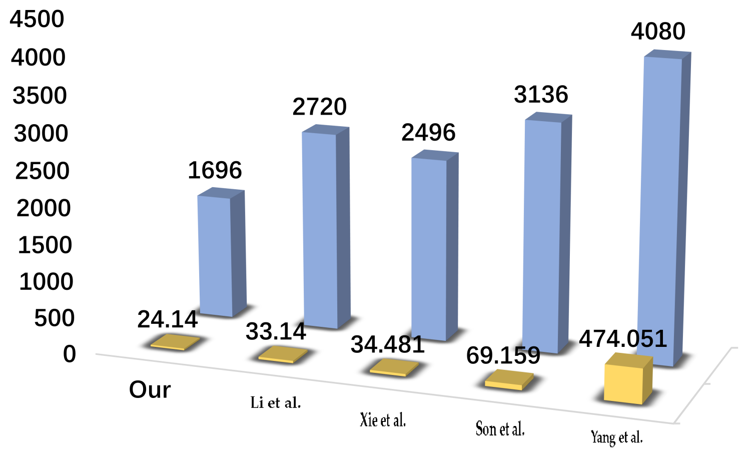

Figure 6. Comparison of calculation and communication cost [26–29].

7.2. Communication Costs Comparison

To measure the communication cost of the login and authentication phase, we assume that the identity, hash digest, random nonce, ECC point multiplication, asymmetric encryption/decryption (RSA-1024), and timestamp are 160 bits, 160 bits, 128 bits, 320 bits, 512 bits and 32 bits, respectively. Therefore, the total communication cost in 3ECAP is 1696 bits. The protocols of Li et al. [26], Xie et al. [27], Son et al. [28] and Yang et al. [29] require 2720, 2496, 3136, and 4080 bits (b) of communication cost, respectively. The details are shown in Table 5.

Table 5. Communication costs comparison.

Messages	3ECAP	[26]	[27]	[28]	[29]
$U_i \rightarrow GW$	672 b	800 b	960 b	672 b	684 b
$GW \rightarrow SD_j$	512 b	640 b	1088 b	672 b	2684 b
$SD_j \rightarrow GW$	—	640 b	—	—	—
$GW \rightarrow U_i$	—	640 b	—	—	—
$SD_j \rightarrow U_i$	512 b	—	448 b	1792 b	712 b
Total cost	1696 b	2720 b	2496 b	3136 b	4080 b

7.3. Security Features Comparison

The comparative analysis of thesecurity and functional features of 3ECAP and other related protocols is presented in Table 5. It can be observed that 3ECAP provides improved security and more functional features compared to the other four protocols. For example, the protocol by Li et al. [26] directly uses the identity of the participating nodes for information transmission, which can be easily tracked by the adversary. Moreover, in IoMT, the permission of different levels of users should be divided, which is not involved in the four other protocols. In contrast, 3ECAP divides several sensors into corresponding clusters based on the user's access list and stores SDL_i in a smart card and gateway, which not only achieves permission segmentation but also eliminates part of the subsequent database validation. Therefore, 3ECAP clearly outperforms other related protocols according to the comparison of all the features in Table 6.

Table 6. Security features comparison.

Feature	3ECAP	[26]	[27]	[28]	[29]
User impersonation attack	✓	✓	✓	×	✓
Gateway impersonation attack	✓	✓	✓	×	✓
Sensor device impersonation attack	✓	✓	✓	×	✓
Stolen verifier attack	✓	—	✓	×	✓
Replay attack	✓	✓	✓	✓	✓
Denial-of-service attack	✓	✓	×	×	×
Sensor device capture attack	✓	×	✓	×	×
Man-in-the-middle attack	✓	×	✓	✓	✓
Insider privilege attack	✓	×	✓	×	✓
Privilege escalation attack	✓	×	×	×	×
Anonymity	✓	—	×	✓	×
Untraceability	✓	✓	×	×	×
Forward secrecy	✓	✓	✓	✓	×
Mutual authentication	✓	✓	✓	✓	×
Session key agreement	✓	×	✓	✓	✓
Biometric update	✓	✓	×	×	×
Password change	✓	✓	✓	×	×
Sensor device addition	✓	—	—	—	—
Two/three factor authentication	3	3	3	2	2
Fine-grained access control	✓	×	×	×	×
Formal analysis	✓	✓	✓	×	✓
Authentication based on Proverif/AVISPA tool	✓	✓	✓	×	×

✓: The protocol securely resists a particular attack or supports a particular feature; ×: the protocol is insecure against a particular attack or does not support a particular feature; —: not applied in the protocol.

8. NS3 Simulation

In this section, we attempt to measure the performance of 3ECAP in terms of network throughput (in bytes/second) and end-to-end delay (EED, in seconds) using the widely accepted NS3 tool (this part was not considered in the previous conference).

8.1. Simulation Parameters and Scenario

Table 7 [30] lists the basic network parameters used in the NS3 simulation. We used the Ubuntu 18.04.4 LTS platform. The simulation of the user, gateway, and sensor was executed on 2.4GHz Wi-Fi media. The gateway was set at the origin. The users were permitted to move randomly in any direction at a speed of 3 m within a 150 m^2 area centered at the origin. Sensors were randomly distributed on an 80-m ring and centered on the gateway. We then set the size of the messages transmitted between the nodes, i.e., $M_1 = 84$ bytes, $M_2 = 64$ bytes, and $M_3 = 64$ bytes.

In this scenario, a complete message transfer consists of (the NS3 simulation does not involve specific cryptographic operations) (1) the user first sends an authentication request M_1 to a gateway in order to access the device; (2) the gateway receives the request and then forwards M_2 to the device; and (3) once it receives the message from gateway, the device sends the message M_3 to the user. Through 1, 2 and 3, the information interaction between a user and a device can be accomplished. Meanwhile, since there is more than one user and device in the scenario, they can all authenticate each other through the gateway. Therefore, it can be assumed that there are multiple message transfers at a given moment. And the main purpose of using NS3 is to show how the total throughput and delay change with the number of participating nodes.

We also set the simulation time for this scenario to 1200 s, which is a relatively appropriate setting that is sufficient to reflect the simulation results of 3ECAP. Finally, we configure a different number of users and devices, and the simulation parameters and results are shown in Table 7 [30] and Figure 7.

Table 7. Simulation parameters.

Parameter	Description	
Platform	NS3 3.27/Ubuntu 18.04.4 LTS	
Mobility	random (3 m/s)	
Simulation time	1200 s	
Scenarios	No. of users	No. of devices
1	10	5
2	5	10
3	8	10
4	5	15
5	5	20
6	8	50

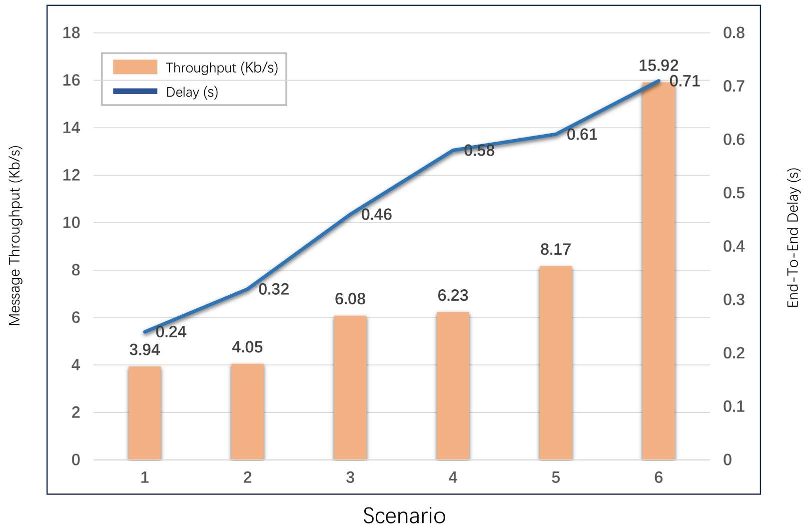

Figure 7. NS3-based 3ECAP simulation results.

8.2. Discussion of Simulation Results

(1) Impact on Network Throughput: The total throughput of 3ECAP in the six scenarios is represented by bar charts in Figure 7. The throughput is calculated as $\varrho_d/(\sigma_s - \sigma_r)$, where the total amount of data received in the simulated environment is ϱ_d, the time to send the first packet is σ_s, and the time to receive the last packet is σ_r. It is observed that as the number of participating nodes, including users and sensors, increases, the network throughput in the network also increases accordingly.

(2) Impact on End-to-End Delay: The total delay of 3ECAP in the six scenarios is represented by the discounted graph in Figure 7. EED delay can be expressed as $\sum_{i=1}^{\nu_p}(T_{si} - T_{ri})/\nu_p$, where T_{si} and T_{ri} represent the sending time and receiving time, respectively, when the ith packet is transmitted, and the total number of packets transmitted during the simulation is ν_p. It follows from the figure that when the number of participating nodes increases, the number of messages transmitted will increase, which may cause network congestion to the extent that the EED delay increases.

9. Conclusions

Considering the aspects of security, low cost, and access control for IoMT sensors, in this paper, we propose a new efficient cluster-based user authentication protocol (3ECAP).

In 3ECAP, three factors, i.e., password, biometric and smart card, are employed to resist a single-factor incidental guessing attack. In addition, 3ECAP enables user-specific privilege segmentation through fine-grained access control and can address the resulting privilege escalation attack. Furthermore, provable security based on the ROR model, formal verification based on the Proverif tool, as well as non-formal analysis are provided in this paper, and the results demonstrate the robustness of 3ECAP in the face of most attacks. Finally, the comparison and analysis with the latest related protocols indicate that 3ECAP provides higher security and lower computation and communication costs; therefore, it is very suitable for the practical deployment of the IoMT.

Future research directions related to this paper are as follows: (1) implementing and evaluating 3ECAP in real IoMT environments, (2) providing a flexible on-line sensor device addition phase, and (3) supporting dynamic updating of user-accessible lists based on sensor clusters in order to maintain forward and backward secrecy.

Author Contributions: Conceptualization, X.S. and Y.X.; methodology, X.S.; software, X.S.; validation, X.S. and Y.X.; formal analysis, X.S.; investigation, X.S.; resources, X.S. and Y.X.; data curation, X.S.; writing—original draft preparation, X.S.; writing—review and editing, X.S. and Y.X.; visualization, X.S.; supervision, X.S. and Y.X.; project administration, X.S. and Y.X.; funding acquisition, X.S. All authors have read and agreed to the published version of the manuscript.

Funding: The work of this paper was funded by the National Natural Science Foundation of China No. 62371246 and the Practice Innovation Program of Jiangsu Province No. KYCX22_0936.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors would like to thank the reviewers for their valuable feedback and suggestions on this paper, which helped to improve the quality of the paper.

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

References

1. Laghari, A.A.; Wu, K.; Laghari, R.A.; Ali, M.; Khan, A.A. A review and state of art of Internet of Things (IoT). *Arch. Comput. Methods Eng.* **2021**, *29*, 1–19.
2. Soori, M.; Arezoo, B.; Dastres, R. Internet of things for smart factories in industry 4.0, a review. *Internet Things Cyber-Phys. Syst.* **2023**, *3*, 192–204. [CrossRef]
3. Sadhu, P.K.; Yanambaka, V.P.; Abdelgawad, A. Internet of things: Security and solutions survey. *Sensors* **2022**, *22*, 7433. [CrossRef]
4. Zhang, L.; Lin, Y.; Yang, X.; Chen, T.; Cheng, X.; Cheng, W. From Sample Poverty to Rich Feature Learning: A New Metric Learning Method for Few-Shot Classification. *IEEE Access* **2024**, 2024, 124990–125002. [CrossRef]
5. Mahmoud, H.H.H.; Amer, A.A.; Ismail, T. 6G: A comprehensive survey on technologies, applications, challenges, and research problems. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4233. [CrossRef]
6. Tataria, H.; Shafi, M.; Molisch, A.F.; Dohler, M.; Sjöland, H.; Tufvesson, F. 6G wireless systems: Vision, requirements, challenges, insights, and opportunities. *Proc. IEEE* **2021**, *109*, 1166–1199. [CrossRef]
7. Razdan, S.; Sharma, S. Internet of medical things (IoMT): Overview, emerging technologies, and case studies. *IETE Tech. Rev.* **2022**, *39*, 775–788. [CrossRef]
8. Hernandez-Jaimes, M.L.; Martinez-Cruz, A.; Ramírez-Gutiérrez, K.A.; Feregrino-Uribe, C. Artificial intelligence for IoMT security: A review of intrusion detection systems, attacks, datasets and Cloud-Fog-Edge architectures. *Internet Things* **2023**, *23*, 100887. [CrossRef]
9. Garg, N.; Wazid, M.; Singh, J.; Singh, D.P.; Das, A.K. Security in IoMT-driven smart healthcare: A comprehensive review and open challenges. *Secur. Priv.* **2022**, *5*, e235. [CrossRef]
10. 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]
11. Koutras, D.; Stergiopoulos, G.; Dasaklis, T.; Kotzanikolaou, P.; Glynos, D.; Douligeris, C. Security in IoMT communications: A survey. *Sensors* **2020**, *20*, 4828. [CrossRef] [PubMed]
12. Mishra, N.; Pandya, S. Internet of things applications, security challenges, attacks, intrusion detection, and future visions: A systematic review. *IEEE Access* **2021**, *9*, 59353–59377. [CrossRef]
13. Wang, C.; Wang, D.; Duan, Y.; Tao, X. Secure and Lightweight User Authentication Scheme for Cloud-Assisted Internet of Things. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 2961–2976. [CrossRef]

14. Masud, M.; Gaba, G.S.; Choudhary, K.; Hossain, M.S.; Alhamid, M.F.; Muhammad, G. Lightweight and Anonymity-Preserving User Authentication Scheme for IoT-Based Healthcare. *IEEE Internet Things J.* **2022**, *9*, 2649–2656. [CrossRef]
15. Sutrala, A.K.; Bagga, P.; Das, A.K.; Kumar, N.; Rodrigues, J.J.; Lorenz, P. On the design of conditional privacy preserving batch verification-based authentication scheme for internet of vehicles deployment. *IEEE Trans. Veh. Technol.* **2020**, *69*, 5535–5548. [CrossRef]
16. Iqbal, W.; Abbas, H.; Deng, P.; Wan, J.; Rauf, B.; Abbas, Y.; Rashid, I. ALAM: Anonymous lightweight authentication mechanism for SDN-enabled smart homes. *IEEE Internet Things J.* **2020**, *8*, 9622–9633. [CrossRef]
17. Wei, L.; Cui, J.; Xu, Y.; Cheng, J.; Zhong, H. Secure and lightweight conditional privacy-preserving authentication for securing traffic emergency messages in VANETs. *IEEE Trans. Inf. Forensics Secur.* **2020**, *16*, 1681–1695. [CrossRef]
18. Yang, Y.; Huang, X. Comments on "On the Design of Conditional Privacy Preserving Batch Verification-Based Authentication Scheme for Internet of Vehicles Deployment". *Cryptol. ePrint Arch.* **2021**, *018*. Available online: https://eprint.iacr.org/2021/018 (accessed on 1 November 2024).
19. Yu, S.; Das, A.K.; Park, Y. Comments on "ALAM: Anonymous lightweight authentication mechanism for SDN enabled smart homes". *IEEE Access* **2021**, *9*, 49154–49159. [CrossRef]
20. Zhang, J.; Zhang, Q. Comment on "Secure and Lightweight Conditional Privacy-Preserving Authentication for Securing Traffic Emergency Messages in VANETs". *IEEE Trans. Inf. Forensics Secur.* **2021**, *18*, 1037–1038. [CrossRef]
21. Zhang, S.; Liu, Y.; Gao, T.; Xie, Y.; Zhou, C. Practical and Secure Password Authentication and Key Agreement Scheme Based Dual-Server for IoT Devices in 5G Network. *IEEE Internet Things J.* **2024**, *2024*, 34639–34651. [CrossRef]
22. Nandy, T.; Idris, M.Y.I.; Noor, R.M.; Wahab, A.W.A.; Bhattacharyya, S.; Kolandaisamy, R.; Yahuza, M. A Secure, Privacy-Preserving, and Lightweight Authentication Scheme for VANETs. *IEEE Sens. J.* **2021**, *21*, 20998–21011. [CrossRef]
23. Singh, N.; Das, A.K. TFAS: Two factor authentication scheme for blockchain enabled IoMT using PUF and fuzzy extractor. *J. Supercomput.* **2024**, *80*, 865–914. [CrossRef]
24. Chaudhry, S.A. Comments on "A Secure, Privacy-Preserving, and Lightweight Authentication Scheme for VANETs". *IEEE Sens. J.* **2022**, *22*, 13763–13766. [CrossRef]
25. Nyangaresi, V.O. ECC Based Authentication Scheme for Smart Homes. In Proceedings of the 2021 International Symposium ELMAR, Zagreb, Croatia, 13–15 September 2021; pp. 5–10. [CrossRef]
26. Li, X.; Peng, J.; Obaidat, M.S.; Wu, F.; Khan, M.K.; Chen, C. A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems. *IEEE Syst. J.* **2019**, *14*, 39–50. [CrossRef]
27. Xie, Q.; Ding, Z.; Tang, W.; He, D.; Tan, X. Provable Secure and Lightweight Blockchain-Based V2I Handover Authentication and V2V Broadcast Protocol for VANETs. *IEEE Trans. Veh. Technol.* **2023**, *72*, 15200–15212. [CrossRef]
28. Son, S.; Lee, J.; Park, Y.; Park, Y.; Das, A.K. Design of Blockchain-Based Lightweight V2I Handover Authentication Protocol for VANET. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 1346–1358. [CrossRef]
29. Yang, A.; Weng, J.; Yang, K.; Huang, C.; Shen, X. Delegating Authentication to Edge: A Decentralized Authentication Architecture for Vehicular Networks. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 1284–1298. [CrossRef]
30. Su, X.; Xu, Y.; Tong, H.; Li, T. A Cluster-based User Authentication Protocol for Internet of Medical Things Deployment. In Proceedings of the 2023 International Conference on Wireless Communications and Signal Processing (WCSP), Hangzhou, China, 2–4 November 2023; pp. 517–522.
31. Ebrahimi, S.; Bayat-Sarmadi, S. Lightweight fuzzy extractor based on LPN for device and biometric authentication in IoT. *IEEE Internet Things J.* **2021**, *8*, 10706–10713. [CrossRef]
32. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
33. Abdalla, M.; Fouque, P.A.; Pointcheval, D. Password-based authenticated key exchange in the three-party setting. In Proceedings of the International Workshop on Public Key Cryptography, Les Diablerets, Switzerland, 23–26 January 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 65–84.
34. Banerjee, S.; Odelu, V.; Das, A.K.; Srinivas, J.; Kumar, N.; Chattopadhyay, S.; Choo, K.K.R. A provably secure and lightweight anonymous user authenticated session key exchange scheme for internet of things deployment. *IEEE Internet Things J.* **2019**, *6*, 8739–8752. [CrossRef]
35. Das, A.K.; Wazid, M.; Kumar, N.; Vasilakos, A.V.; Rodrigues, J.J. Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial Internet of Things deployment. *IEEE Internet Things J.* **2018**, *5*, 4900–4913. [CrossRef]
36. Roy, S.; Das, A.K.; Chatterjee, S.; Kumar, N.; Chattopadhyay, S.; Rodrigues, J.J. Provably secure fine-grained data access control over multiple cloud servers in mobile cloud computing based healthcare applications. *IEEE Trans. Ind. Inform.* **2018**, *15*, 457–468. [CrossRef]
37. Wang, D.; Cheng, H.; Wang, P.; Huang, X.; Jian, G. Zipf's law in passwords. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2776–2791. [CrossRef]
38. Blanchet, B.; Smyth, B.; Cheval, V.; Sylvestre, M. ProVerif 2.00: Automatic cryptographic protocol verifier, user manual and tutorial. *Version* **2018**, *16*, 5–16. Available online: https://bblanche.gitlabpages.inria.fr/proverif/manual.pdf (accessed on 1 November 2024).
39. Challa, S.; Wazid, M.; Das, A.K.; Kumar, N.; Reddy, A.G.; Yoon, E.J.; Yoo, K.Y. Secure signature-based authenticated key establishment scheme for future IoT applications. *IEEE Access* **2017**, *5*, 3028–3043. [CrossRef]