electronics

Article

Physical Unclonable Function and Machine Learning Based Group Authentication and Data Masking for In-Hospital Segments

Pintu Kumar Sadhu [1,*], Venkata P. Yanambaka [2] and Ahmed Abdelgawad [1]

[1] College of Science and Engineering, Central Michigan University, Mount Pleasant, MI 48858, USA
[2] Department of Mathematics and Computer Science, Texas Woman's University, Denton, TX 76204, USA
* Correspondence: sadhu1pk@cmich.edu

Abstract: The involvement of the Internet of things (IoT) in the development of technology makes systems automated and peoples' lives easier. The IoT is taking part in many applications, from smart homes to smart industries, in order to make a city smart. One of the major applications of the IoT is the Internet of medical things (IoMT) which deals with patients' sensitive information. This confidential information needs to be properly transferred and securely authenticated. For successful data protection and preserving privacy, this paper proposes multidevice authentication for the in-hospital segment using a physical unclonable function (PUF) and machine learning (ML). The proposed method authenticates multiple devices using a single message. Most of the protocols require PUF keys to be stored at the server, which is not required in the proposed framework. Moreover, authentication, as well as data, is sent to the server in the same message, which results in faster processing. Furthermore, a single ML model authenticates a group of devices at the same time. The proposed method shows 99.54% accuracy in identifying the group of devices. Moreover, the proposed method takes 2.6 ms and 104 bytes to complete the authentication of a device and takes less time with the increment of devices in the group. The proposed algorithm is analyzed using a formal analysis to show its resistance against various vulnerabilities.

Keywords: Internet of Things; Internet of Medical Things; smart city; security and privacy; authentication framework; group device authentication; physical unclonable function; machine learning

Citation: Sadhu, P.K.; Yanambaka, V.P.; Abdelgawad, A. Physical Unclonable Function and Machine Learning Based Group Authentication and Data Masking for In-Hospital Segments. *Electronics* **2022**, *11*, 4155. https://doi.org/10.3390/electronics11244155

Academic Editors: Sabrine Kheriji, Olfa Kanoun, Faouzi Derbel and Suleiman Yerima

Received: 31 October 2022
Accepted: 12 December 2022
Published: 13 December 2022

1. Introduction

Industry 4.0, or the fourth industrial revolution, was coined in 2011 by the German Federal Government to emphasize its high-tech approach [1]. Industry 4.0 aimed to integrate the physical components or devices for manufacturing (i.e., various machines, sensors, complex tools) and communication parts (i.e., advanced software) through wired/wireless networks to predict, control, maintain, and integrate the manufacturing process [2]. IoT, cyberphysical systems, cloud computing, edge computing, Big Data analytics, robotics, virtual reality, etc., are the categories of industry 4.0 [3]. The smart city is a major application to make industry 4.0 successful. Many countries are adopting smart cities to provide a better quality of life [4]. Since the smart city generates a network where all connected devices can communicate with one another, enabling the creation of a device-to-device or machine-to-machine network, it is imperative to develop an omnipresent computing system [5]. To provide seamless support and better health service, the IoMT system is developing in the smart city. The IoMT is a system where people with both wearable or implantable medical devices (MD) suffering from diseases, for example, blood pressure, cardiac, and diabetes, are connected to a network to transfer health data to health experts. Experts check the data and prescribe the patients accordingly [6]. The environment of the IoMT can be classified into five categories.

1. On-body segment: consumer health wearables (used for fitness or health data monitoring such as fitness bands, smartwatches, smart shoes, smart clothes, etc.) and clinical-grade wearables (for example, elders can wear smart belts for identifying any risks and providing safety support, Halo Sport headset to activate specific brain regions, etc.) [7].
2. In-home segment: For making a healthy home, wearable devices for monitoring patients at home use health data collection using sensors [8]. For example, a personal emergency response system can be used by older people to get live service; telemedicine and digital tests are part of the in-home IoMT segment.
3. Community segment: The MDs and health stations of a particular area develop this segment [9]. This segment considers mobility, emergency response intelligence, kiosks, point-of-care devices, and logistics as components.
4. In-clinic segment: the MDs that help to gather necessary data and device suggestions regarding administrative and clinical operations build this segment [10].
5. In-hospital segment: This segment manages the system of a hospital using MDs' data. This can provide solutions in the area of patient management, personnel management, environment, etc., in the hospital [11].

Figure 1 shows the IoMT ecosystem in a hospital environment in a smart city [12]. In the IoMT system, all the MDs are connected to the cloud server through the internet (wired or wireless). MDs collect data from the human body, and the data are passed to the cloud server. Doctors or experts access the data to take further action. Doctors do not need to visit in person to check the condition of patients. The advantages of the IoMT are attracting the healthcare sector to adopt it.

Figure 1. IoMT ecosystem of in-hospital segment in smart city.

According to Deloitte's report, the market value of the IoMT will reach around USD 158 billion by 2022 [13]. The IoMT is bringing advantages not only to the doctors and patients' management but is also becoming a blessing economically. It is expected that the healthcare industry will be able to save approximately USD 300 billion per year. Nevertheless, the IoMT is also bringing severe challenges in the area of security and privacy.

1.1. Security and Privacy Concerns in the IoMT

Security and privacy concerns are rising in the IoMT system. Different adversarial impacts, such as data breach, identity theft, data modification, denial of service, etc., could pose threats to the IoMT network [14]. A report from Cynerio in August 2022 stated that in 88% of cases of cyberattacks, an MD was involved. Moreover, healthcare had to pay around

USD 1 million for average data breach cases, and it increased the mortality rate by 24% [15]. A total of 2.9 million subscribers were affected in the cyberattack on Health South-East RHF in Norway. Approximately 19,000 appointments were canceled in the WannaCry ransomware attack on England's National Health Service. Moreover, they had to spend 92 million to recover from the threat. Since the IoMT deals with sensitive information, it is required to preserve the security of the system and preserve user privacy to avoid data loss, unauthorized access, or malicious attackers.

IoMT systems need to implement a robust and secure authentication and data communication mechanism [16]. Authentication systems can be developed using a decentralized system, such as the blockchain, and also using a centralized server system. In the blockchain, a number of transactions make a block and connect with the previous blocks to build a chain. In the centralized server system, the server is responsible for making decisions. Different encryption methods, such as advanced encryption standard (AES), message authentication code (MAC), attribute-based encryption (ABE), elliptic curve cryptography (ECC), etc., are used to develop authentication mechanisms. A PUF is another method to develop an authentication method that does not need to store a cryptographic key in the system. To build a robust and secure authentication framework as well as data masking, this paper used a PUF and ML.

A PUF is a one-way function that provides a unique Boolean mapping between the input of the PUF and the output of the PUF. The manufacturing process differences of the chips create statistical process variations that define the PUF working principle [17]. PUF can also be called a chip fingerprint. It is a useful tool for applications such as authentication, validation, identification, etc. Unlike conventional methods to store secret keys, such as encryption and password, a PUF generates secret keys instantly using a semiconductor chip's complex characteristics [18]. Due to imperfections in the manufacturing process, a PUF generates different secret keys when a different input is provided. The combination of input (challenge) and output (response) of the PUF is the signature of the PUF, which is referred to as a challenge–response pair (CRP) [19]. Figure 2 shows the CRP relations among different PUFs. It can be seen that n sets of challenges are sent to n PUFs where each PUF generates different responses due to different process variations of the chips. Recently, mathematical models was used against PUFs to exploit it. The attack was called a modeling attack. A successful tool for performing modeling attacks is ML [20]. By protecting CRP interfaces, it is possible to resist modeling attacks.

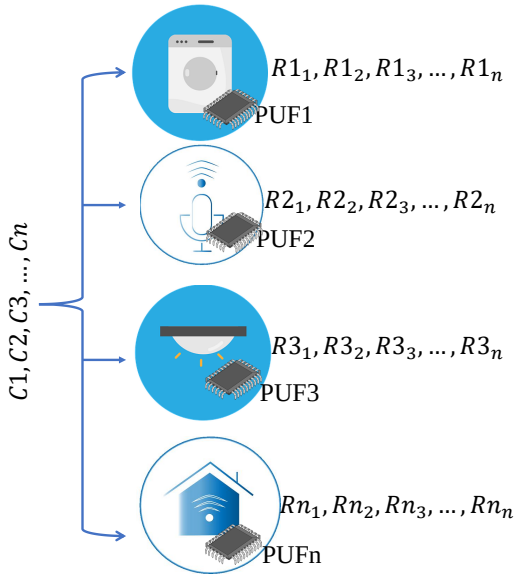

Figure 2. CRP characteristics of PUFs.

On the other hand, ML is mainly driven by data [21]. ML is a powerful tool that is making life easier and providing faster solutions to complex systems. In healthcare, ML is a blessing by identifying diseases using images and data, scheduling appointments, and creating a better relationship between patients and doctors [22]. ML is a method that tries to learn outputs using input data through mathematical models. There could be a number of input features and a number of output features. Supervised, unsupervised, and reinforcement learning are the three types of ML [23]. In supervised learning, a training dataset is required. During training, a decision rule is trained, and a classifier is constructed. The learning process runs till the desired goal is achieved with a proper optimization of weights and other parameters [24]. Supervised ML is divided into segments such as binary classification, multiclass classification, regression, etc. In binary classification, there are two kinds of output classes, and a sigmoid is used in the last layer of the modeling. In multiclass classification, there are a number of classes in the output feature. There are three types of layers which are the input layer, the hidden layer, and the output layer. The input features map with the output class to learn properly. Figure 3 shows an example of multiclass classification. The first layer is the input layer, where the size of the layer depends on the features of the dataset. The next layer is the hidden layer, which is a layer of mathematical functions, each designed to produce an output specific to an intended result. There could be more than one hidden layer. The last layer is the output layer, where the size is the classes/labels of the dataset. In this kind of supervised learning, a softmax function with the number of classes as unit size is in the last layer of the model. An explanation of the relationships among input features and one or more responses or target output features is provided by a regression model.

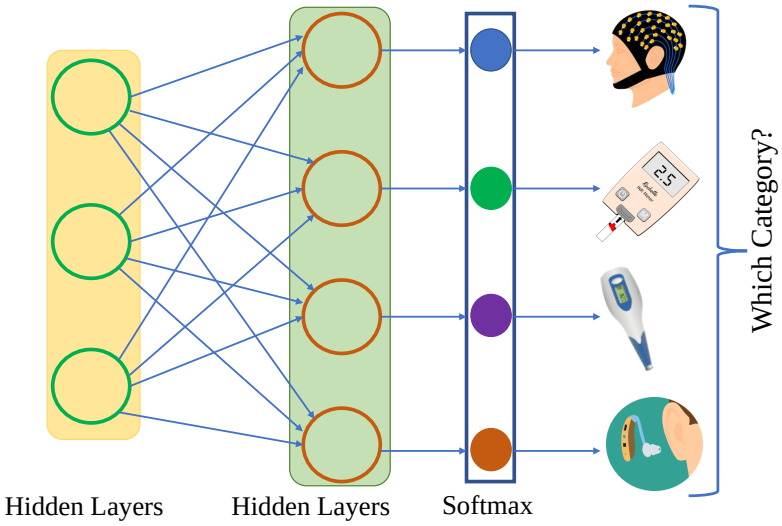

Figure 3. Multiclass classification model.

1.2. Contributions

Researchers in both academia and the industry are working continuously to develop authentication methods to secure IoMT applications. The developed methods need to be robust enough to provide security against known security threats. Moreover, it is required to avoid complex calculations and make the framework lightweight as the IoMT devices are resource-constrained and cannot allocate a lot of resources for authentication purposes. Furthermore, the authentication process should not raise a burden on the communication network. Most of the existing methods use single device authentication and need to perform several steps of communication to validate the devices. This type of method consumes the transmission resources and device resources to perform operations at different times. Moreover, the device needs to send data separately after completing the authentication. To resist known security attacks, validate multiple devices, and send

data in the authentication request, simple and lightweight operations along with a low communication cost are required; this paper proposes an authentication framework using a PUF and ML to validate a group of medical devices and send data at the same time. The contributions of the proposed framework are as follows.

- It is the general norm for PUF-based methods to transmit challenges to the device and/or the cloud. The proposed framework incorporates machine learning to control the PUF. It eliminates the requirement of transmitting challenges from the cloud server.
- Usually, CRPs are stored on the cloud server to verify the devices. The proposed method removes the requirement of CRP storage in the cloud server, which reduces the storage cost in the secure database.
- A group of devices are authenticated at a time instead of a single device.
- A single message transfer is adequate to complete the authentication of the group of medical devices.
- A single machine learning model identifies a group of medical devices. The proposed method eliminates the requirement of storing multiple models for multiple devices.
- No need to transfer data separately after authentication. Health data are sent at the same time with the authentication request.
- The secret encryption key in the edge router is updated periodically, which removes prevents key guessing attacks.
- The method involves each device's authentication separately, it follows a linear relation for communication overhead. Communication overhead is decreased with the increment of devices in the proposed method.
- Less computation cost is involved. The cost is also decreased with the number of devices.

The rest of the paper consists of related work in Section 2, the proposed framework in Section 3, results in Section 4, a security proof in Section 5, and the conclusion and future work in Section 6.

2. Related Work

Li et al. [25] proposed a lightweight authentication mechanism using a secret key. The method required a trusted gateway to register the node using the secret key. The secret key helped to establish the session key. The method was lightweight; however, the storage cost was high in the developed scheme. Amintoosi et al. [26] developed another lightweight scheme using identity (ID), a password, and a smart card. This scheme used operations such as hash, XOR, concatenation, etc., to make it lightweight. A MAC-based authentication scheme was developed by Siddiqi et al. [27]. In the method, the server used the key where K-bit was missing, and the sensor device needed to identify the hash of the missing bits. The method saved energy but did not preserve user anonymity. Hwang et al. [28] developed a CP-ABE-based framework for the IoMT environment. The ciphertext length was the same and it was independent of other parameters. The method identified the root cause by finding the first receiver of the key. The proposed method needed significant computations to identify devices. If an attacker received the delegated key, there could be a chance of health data leakage. Liu et al. [29] proposed a multikeyword searchable method using the ABE mechanism. The proposed method used ABE to encrypt the symmetric key. The method performed most of the decryption calculations on the server, which reduced the calculations on the user side. A combination of boolean attributes and a weighted comparison of attributes was used to develop a new method called ciphertext policy weighted-attribute-based encryption (CP-WABE) by Li et al. [30]. The storage cost was comparatively high due to the storage of a set of weights. An ECG- based IoMT device authentication scheme was developed by Huang et al. [31]. The method removed the noise from the ECG signal to operate. Depending on the saved attributes and patients' movement, noise interference was changed. Angular distances dues to different movements, such as running and walking, could impact the proposed method. The method needed to improve to preserve user anonymity and reduce computation cost. Another method developed by Ying et al. used ECG templates as user biometrics [32].

The framework used ECC and a smart card. Different nonces and hash functions were written in the smart card for verification purposes. The method needed to cater to ECG-related problems and server impersonation attacks. Moreover, the method needed to optimize communication overhead. Ryu et al. also proposed an ECC based authentication framework [33]. The proposed method also used the user's biometrics, but it kept the features separate to avoid attacks due to theft parameters. The proposed method could optimize storage costs due to unused parameters in the authentication process. Moreover, the cost of the framework was high comparatively. Al-Zubaidie et al. [34] developed an authentication method using ECC and the PHOTON hash function. The method used information from the user and device for the successful completion of the authentication. Moreover, information related to physical address, a one-time password, etc., was part of the authentication framework. Gopalan et al. [35] proposed an ECC based mechanism for secure data transfer when the initial authentication is completed. The method used log-of-round-value-based elliptic curve cryptography (LR-ECC) for securing the data transfer. De-Marcos et al. [36] developed a continuous authentication method to avoid the security issue of one-time authentication. They used seven different classifiers where the ensemble algorithm showed better performance for classification. Wazid et al. developed an authentication mechanism for secure communication between a device and a personal server using a secret key [37]. They also used artificial intelligence for data analysis. The method used a decision tree, a support vector machine, and logistic regression, and the accuracy was below 88%. The above methods used different cryptographic keys, which are required to be stored in the device.

To avoid key storage, a PUF-based authentication mechanism was developed by Alladi et al. [38]. The method used two factors to complete the authentication. Moreover, instead of using a complete response, the method used a segmented response. However, the method required complex computations. Moreover, the method needed to store the challenge, and after completing authentication, the device needed to select the challenge in random order. If the PUF of the device could not produce a stable response for the challenge, there was authentication failure. Gope et al. [39] also developed a PUF-based authentication framework for the IoMT system. Both server and client devices verified one CRP to identify each other. Two CRPs were required to transfer over the communication channel. The process of updating reserve CRPs was not mentioned in the method. Lee et al. [40] also used a PUF to develop an authentication mechanism for a dynamic group key agreement. A key agreement between two devices was initiated by a register center. After that, CRPs from both devices, stored parameters, and different operations were used for mutual agreement. Both devices broadcast the keys to update the group key by other devices of the group. One of the devices for mutual agreement was required to be an existing device for a successful update. Moreover, the method needed to state the procedure for getting the existing group key for the new devices.

To avoid using a centralized server, blockchain-based authentication mechanisms have been used. Abdellatif et al. [41] developed a blockchain-based authentication mechanism which integrated edge computing to process health data. The method created three different channels for different kinds of data. The work focused on prioritizing different levels of data. Another blockchain-based authentication system was developed by Lin et al. [42]. They used reinforcement learning to allocate resources. In the method, a 360° video was captured using virtual reality. The method could identify the common view of similar patients, but a comparatively high computing capacity was required. Egala et al. [43] developed a novel blockchain-based system. The method introduced selective ring-based access control and ECC to authenticate the device. The method also maintained a gray list to record unauthorized devices. Wang et al. [44] developed an authentication mechanism using a PUF. They applied fuzzy extraction to handle users' biometric data. Table 1 shows a comparison of existing authentication schemes.

Table 1. Comparative analysis of related works.

Author	Objective	Technique Used	Pros	Cons
Li et al. [25]	Reduce complexity and secure communication	PKI	Lightweight scheme	Much time and storage required
Siddiqi et al. [27]	Security protocol for IMD ecosystem	MAC	7% energy consumption	No user anonymity
Hwang et al. [28]	Improve CP-ABE-based scheme	CP-ABE	Resolves key abuse problem	PHI leakage
Liu et al. [29]	Achieve data SNP preservation	ABE	Major decryption on server side	Complex
Huang et al. [31]	Protection from unauthorized entity	ECG	Remove noise, light algorithm	No anonymous identity
Ying et al. [32]	Secure communication	ECC	Low computational time	High communication overhead
Ryu et al. [33]	Robust authentication	ECC	Used biometrics along with stored parameter	Unused parameters
Wazid et al. [37]	Secure communication among devices, personal server, and cloud server	AI	Low end-to-end delay	Low accuracy
Alladi et al. [38]	To achieve physical security	PUF	Low computation time	Unstable CRP can cause failure
Gope et al. [39]	Secure and efficient authentication	PUF	Less computation at server	Two CRPs per transaction
Lee et al. [40]	Establish group key agreement	PUF	Simple	Two new devices cannot take part at a time
Abdellatif et al. [41]	Process large quantities of medical data	Blockchain	Remote monitoring, different actions for different data	Security is not focused
Egala et al. [43]	Efficient secure exchange for decentralized network	Blockchain and ECC	Low energy, fast response	Ring tamper resistance instead of device
Wang et al. [44]	To build a reliable communication channel for healthcare	Blockchain (PoW) and PUF	Low cost	Storage cost

3. Proposed Group Medical Devices Authentication and Data Masking Framework

In the proposed method, a group of MDs are authenticated, and their data are masked. The cloud server (CS) authenticates and retrieves the original data with the help of an edge router (ER). A group of MDs are connected to the ER through a wired/wireless medium. The below elements are used in the proposed framework. Figure 4 shows the overview of the proposed mechanism. Table 2 shows the acronyms used in the proposed section.

- Medical Devices: The MDs are wearable devices used by patients on hospital premises. These MDs collect data from patients' bodies for further analysis by doctors or other experts. Each MD is equipped with a PUF and also stores a unique challenge. The MDs are connected to the hospital network.
- Edge Router: The ER is the gateway of the hospital, which has enough processing power to handle multiple requests and perform many tasks at the same time. It is not a limited-resources device like an MD. It is responsible for network handling, maintaining MDs' authentication processes, data masking, etc. Like MDs, it is also equipped with a PUF. Moreover, an ML model is stored in the ER to control the PUF.

- Cloud Server: The CS is the central element for making the decision and storing any kind of data. The CS stores the authentication parameters of the MDs. It is the only trusted element in the network. It is the most powerful device in the IoMT network. It is responsible for authenticating MDs and retrieving data. After the extraction of data, it stores the data in a secure database (SDB) through a secure channel. Moreover, the SDB stores the pseudoidentity (PID) of each MD and ERs. Furthermore, two ML models are stored in the SDB for authentication purposes.
  1. PUF controlling model: The PUF used in the method is a controlled PUF or MC-PUF [45]. PID and timestamp (T_{st}) are used as the input features, and the partial challenge is the output feature of the model. This ML model is stored in both the EG and the SDB. The model is called MC_{model}.
  2. Device prediction model: This model is responsible for identifying the MDs. The CRPs of the group of MDs are collected and trained. The CRPs are the input features of the model and the MDs identity are the output feature. This model is stored in the SDB for identifying the PIDs of the MDs. The model is named ML_{model}.

Table 2. Acronyms used in the proposed section.

Acronym	Full Form	Acronym	Full Form
MD	IoMT device	PUF	Physical unclonable function
ER	Edge router	CS	Cloud server
SDB	Secure database	PID, PID_1, PID_2, PID_n	Pseudoidentity
MC_{model}	PUF controlling model	ML_{model}	Device prediction model
C, C_1, C_2, C_3, C_n	Challenge of MDs	R, R_1, R_2, R_3, R_n	Response of MDs
$C1_n, C2_n, C3_n, Cn_n$	Stored challenge in MDs	$R1_n, R2_n, R3_n, Rn_n$	Stored response in MDs
C_{ER}	Stored challenge in ER	R_{ER}	Stored response in ER
$C1(p), C2(p), Cn(p)$	Partial challenge	Res_1, Res_2, Res_n	MD's generated response
T_{st}	Timestamp	DT_1, DT_2, DT_n	Health data
$(Y)_E$	Y encrypted using E	$E(Y)$	Y decrypted using E
$\rightarrow$	CRP generation	$\longrightarrow$	Data transfer
$\mapsto$	ML model prediction	$\models$	Model training
H	Hash operation	$\oplus$	XOR operation
$\in$	Store operation	?	Validation checking
$\|\|$	Concatenation operation	$\times$	Delete operation

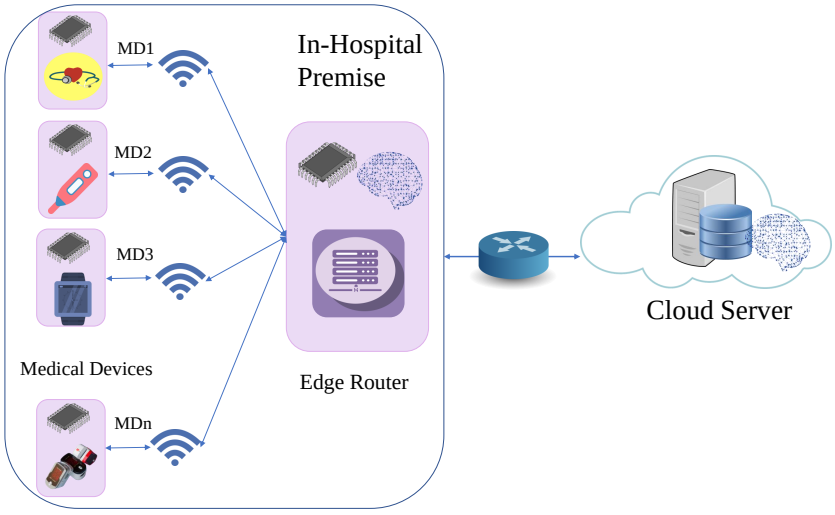

Figure 4. Overview of the proposed framework.

3.1. Assumptions

The following assumptions need to be considered for the successful deployment of the proposed group device authentication and data masking/retrieving operations.

- Each MD and the ER of the group need to be incorporated in the PUF module.
- The model which is stored in the ER for partially controlling the group of MDs is not modified.
- The ER is already authenticated prior to the MD group validation process.
- The stored challenge in the ER is updated periodically.
- The PUFs used are strong and reliable PUFs. The PUFs are not affected by noise and external parameters.
- The secure connection between the CS and the SDB is uninterrupted.
- No impact on CS and SDB is considered in the method.
- IDs, $PIDs$, CRPs, and models are stored in the trusted SDB only.
- The group of MDs are enrolled at the same time. If any modification in the group is required, the model of the CS is updated. Moreover, the ER is updated with the required data of the new MD. After successful updating, the MD is included in the group.

The group of MDs' authentication process is divided into two phases which are the enrollment phase and the authentication with data masking/retrieving phase. In the enrollment phase, both the group of MDs and the ER are enrolled in the network and necessary information is stored in the MDs, ER, and SDB. The successful training of ML models is done in the enrollment phase.

3.2. Edge Router Enrollment

Before placing the ER in the network of the hospital, it is registered to the IoMT network. The enrollment process of the ER is shown in Figure 5. The enrollment process is completed in a secure environment. The total enrollment phase is divided into two steps as shown in Algorithm 1.

1. MC_{model} training: The CS trains the MC_{model} stored in the SDB. Moreover, it shares the model with the ER. After that, the ER receives the PID and a secret response for each MD.
2. Encryption of credentials: The ER first selects a secret challenge C_{ER} and stores it. The incorporated PUF of the ER generates R_{ER} using the challenge C_{ER}. The R_{ER} is used as the encryption key to encrypt the $PIDs$ and the secret responses of the MDs. After encryption, the ER transfers the R_{ER} to the CS for storing purposes in the SDB.

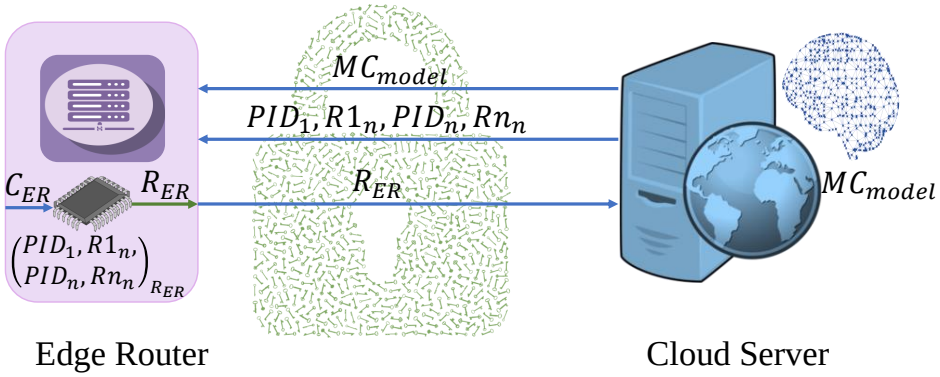

Figure 5. Edge router enrollment for a group of devices.

Algorithm 1 Edge router secure registration process.

Step 1: MC_{model} **training**
Server:
$PID, T_{st}, C \models MC_{model}$
Server $\longrightarrow$ edge router $\{(PID_1, PID_2, \ldots, PID_n), (R1_n, R2_n, \ldots, Rn_n), MC_{model}\}$
Step 2: Encryption of credentials
Edge router:
$\in MC_{model}$
PUF:
$C_{ER} \rightarrow R_{ER}$
$(PID_1, R1_n, PID_2, R2_n, \ldots, PID_n, Rn_n)_{R_{ER}}$=$SecKy_{ER}$
$\in SecKy_{ER}$
Edge router $\longrightarrow$ server $\{R_{ER}\}$
Server $\longrightarrow$ database $\{R_{ER}\}$
Database:
$\in R_{ER}$

3.3. Group of Medical Devices Enrollment

The group of MDs are enrolled at the same time and in a secure environment. Like the ER enrollment process, the group of MDs' registration is also divided into two steps as shown in Algorithm 2. The whole enrollment process is presented in Figure 6.

1. MC_{model} Prediction and challenge collection: The stored MC_{model} uses different timestamps and PIDs to generate challenges for each MD. The different sets of challenges are sent to the different MDs.
2. PUF response generation and ML_{model} training: Each MD generates responses using the received set of challenges using the incorporated PUF in each MD. Each MD shares the set of CRPs with the CS. After receiving all the CRPs from each MD, the CS trains ML_{model} and stores it in the SDB.

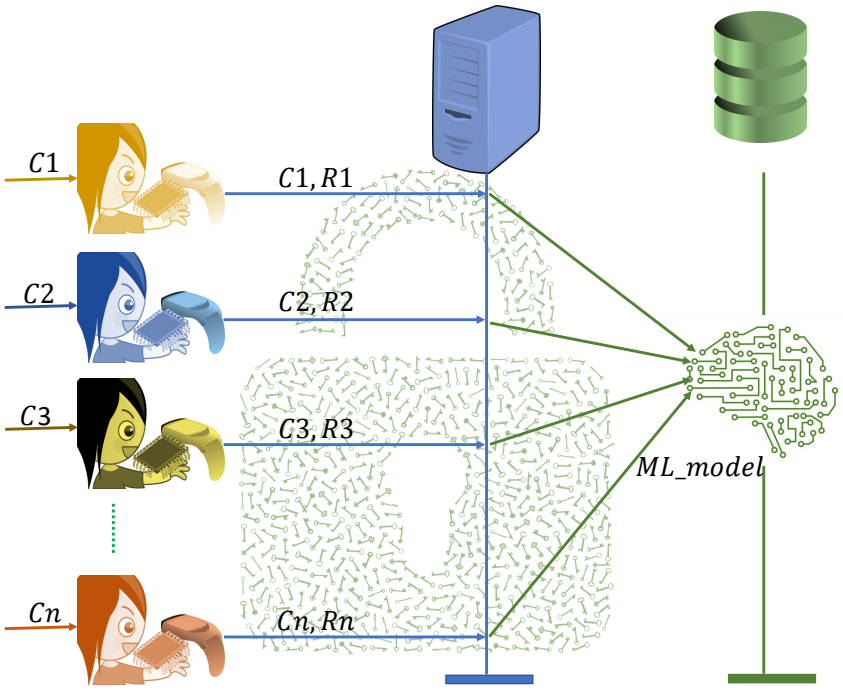

Figure 6. A group of medical devices enrollment.

Algorithm 2 Secure registration process of a group of medical devices.

Step 1: MC_{model} **prediction and challenge collection**
Server:
MC_{model}:
$PID, T_{st} \mapsto C$
Server $\longrightarrow$ device $\{C\}$
Step 2: PUF response generation and ML_{model} **training**
IoMT device:
PUF:
$C \rightarrow R$
Device $\longrightarrow$ server $\{C, R\}$
Server:
$C, R, PID \models ML_{model}$
Server $\longrightarrow$ database $\{ML_{model}\}$
Database:
$\in ML_{model}$

3.4. Proposed Group Devices Authentication and Data Masking

Figure 7 shows the proposed method regarding group of MDs' authentication and data masking/retrieving process. The total mechanism is divided into four steps as shown in Algorithm 3.

1. Request for response generation: Authentication starts in the ER with the secret response generation by sending the stored challenge to the incorporated PUF in the ER. The ER uses the response to decrypt the stored secret message to find out the PIDs of the group of MDs and the secret response of each MD. After this, the ER uses MC_{model} to generate a partial challenge. PID and T_{st} act as input features of the model. A random nonce is concatenated with the partial challenge, then an XOR operation is performed with each secret response of each MD. The output of each XOR operation is shared with the corresponding MD.
2. Response sharing by MDs: Each MD generates its secret response using the stored challenge. By performing an XOR operation, MDs get the partial challenge and random nonce. Each MD finds out the complete challenge by combining the received partial challenge from the ER and its own PID. The MDs generate the response using the PUF to validate the identity of the MD. The collected data and the random nonce are concatenated and an XOR operation is performed.
3. Authentication request and data masking: The ER separates the responses and data using concatenation and an XOR operation using the nonce. The ER calculates DT using PIDs, completed challenges, and data. Here, XOR operations are performed to mask all the information. HD is calculated by performing hash operations of partial challenges, the response of the ER, and data. Moreover, responses are concatenated to define Res and the PIDs of all the MDs and the ER are calculated and a hash is made to find out H_{PID}. DT, HD, Res, and H_{PID} are sent to the CS.
4. Device authentication and retrieving data: After receiving a request from the ER, the CS runs the MC_{model} to predict partial challenges like the ER and also calculates the complete challenges. Both challenges and responses act as input features of the ML_{model} to predict the PIDs to verify the identity. H_{PID} is verified to complete the initial verification of the MDs and the ER. Using the challenges, PIDs, and DT, the data of the MDs are extracted. The retrieved data are verified if the calculated $HD^{'}$ matches the received HD. This completes the group of MDs' authentication and data retrieving process.

Algorithm 3 Group of devices' authentication and secure data transfer.

Step 1: Request for response generation
Edge router:
PUF:
$C_{ER} \rightarrow R_{ER}$
$(PID_1, R1_n, PID_2, R2_n, \ldots, PID_n, Rn_n)$=$R_{ER}(SecKy_{ER})$
MC_{model}:
$PID, T_{st} \mapsto C1(p), C2(p), \ldots, Cn(p)$
$\{(R1_n \oplus (C1(p)||N_1)), (R2_n \oplus (C2(p)||N_1)), \ldots, (Rn_n \oplus (Cn(p)||N_1))\}$
Edge router $\longrightarrow$ IoMT device 1 $\{(R1_n \oplus (C1(p)||N_1))\}$
Edge router $\longrightarrow$ IoMT device 2 $\{(R2_n \oplus (C2(p)||N_1))\}$
.....
Edge router $\longrightarrow$ IoMT device n $\{(Rn_n \oplus (Cn(p)||N_1))\}$
Step 2: Response sharing by MDs
IoMT Device 1:
PUF:
$C1_n \rightarrow R1_n$
$(R1_n \oplus (C1(p)||N_1)) \oplus R1_n = C1(p)||N_1$
$C1(p)||PID_1 = C_{n1}$
PUF:
$C_{n1} \rightarrow Res_1$
IoMT device 1 $\longrightarrow$ edge router $\{(Res_1||DT_1) \oplus N_1\}$
Step-3: Authentication request and data masking
Edge router:
$((PID_1||PID_2||\ldots||PID_n) \oplus (C_{n1}||C_{n2}||\ldots||C_{nn}) \oplus (DT_1||DT_2||\ldots||DT_n)) = DT$
$H(C1(p)||C2(p)||\ldots||Cn(p)||R_{ER}||DT_1||DT_2||\ldots||DT_n) = HD$
$(Res_1||Res_2||\ldots||Res_n) \oplus R_{ER} = Res$
$H(PID_{ER}||PID_1||PID_2||\ldots||PID_n) = H_{PID}$
Edge router $\longrightarrow$ Server $\{DT, HD, Res, H_{PID}\}$
Step 4: Device authentication and retrieving data
Server:
MC_{model}:
$PID, T_{st} \mapsto C1(p), C2(p), \ldots, Cn(p)$
ML_{model}:
$C, Res \mapsto PID$
if $H(PID_{ER}||PID_1||PID_2||\ldots||PID_n) == H_{PID}$ **then**
Initial verification of IoMT devices and edge router
else
Invalid IoMT device and edge router
end if
$((PID_1||PID_2||\ldots||PID_n) \oplus (C_{n1}||C_{n2}||\ldots||C_{nn}) \oplus DT) = (DT_1||DT_2||\ldots||DT_n)$
$H(C1(p)||C2(p)||\ldots||Cn(p)||R_{ER}||DT_1||DT_2||\ldots||DT_n) = HD^{'}$
if $HD == HD^{'}$ **then**
$(DT_1||DT_2||\ldots||DT_n)$ is actual IoMT data
else
Invalid IoMT data
end if

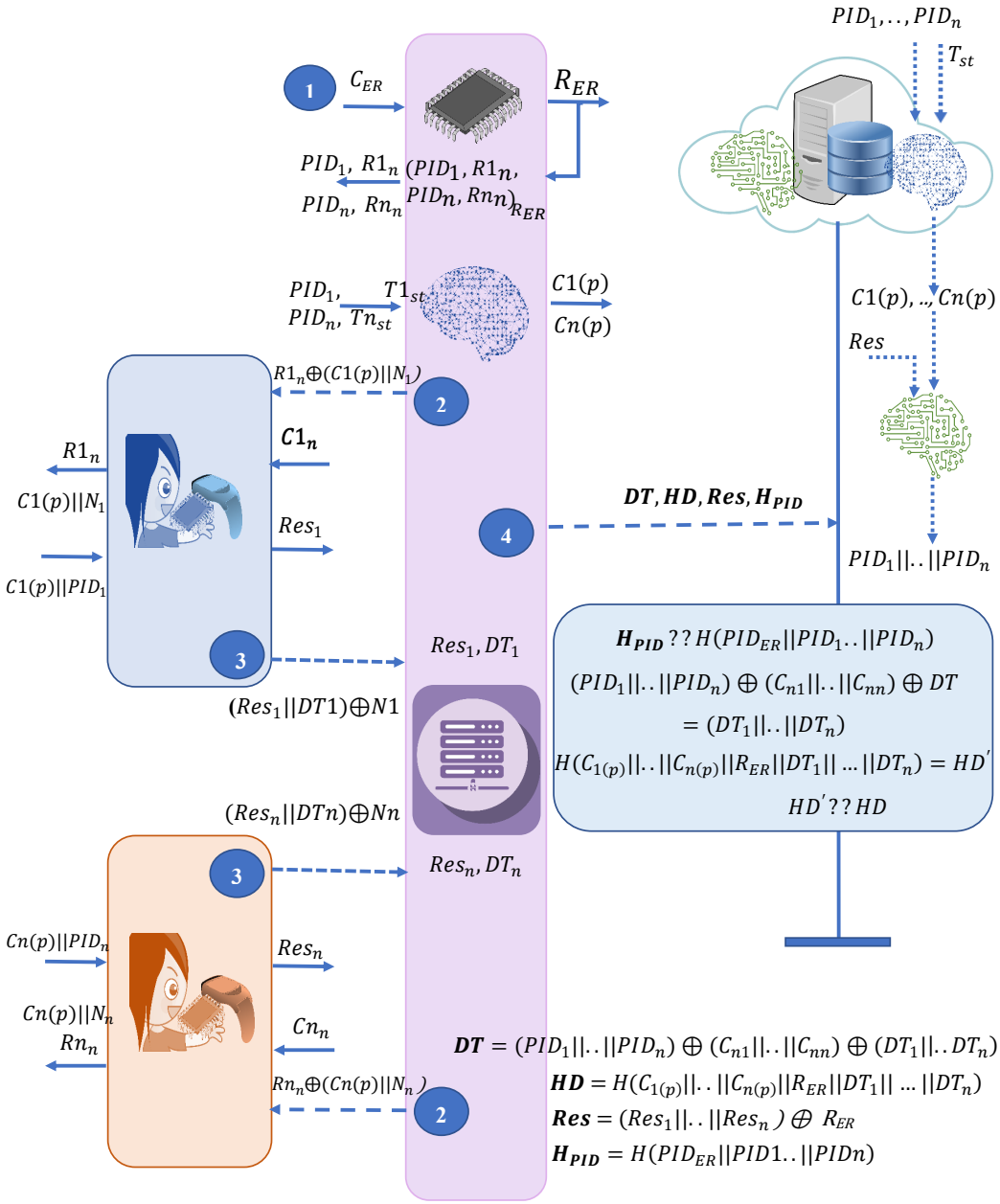

Figure 7. The proposed authentication and data masking framework.

3.5. Encryption Key Update Process of the Edge Router

To avoid exposure to the secret response, the proposed framework uses periodic updates of the secret key. The encryption key update process is shown in Figure 8. The key-changing process is divided into two steps which are presented in Algorithm 4.

1. New secret key generation: The ER selects a new challenge $C_{ER(new)}$ and generates response $R_{ER(new)}$ using the PUF. It decrypts the stored message using the previous secret response and encrypts it again using the new generated response.
2. Updating secret key: The ER uses the XOR operation between $R_{ER(new)}$ and R_{ER} and transfers the result to the CS using a public channel. After receiving the message, the CS uses the XOR operation to get the new secret key of the ER and stores it in the SDB.

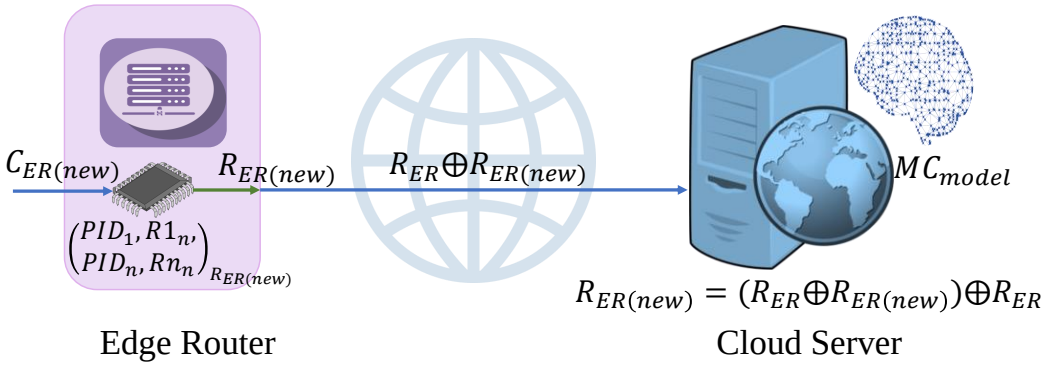

Figure 8. Edge router CRP update process.

Algorithm 4 Edge router encryption key update process.

Step 1: New secret key generation
Edge router:
$C_{ER(new)} \rightarrow R_{ER(new)}$
$(PID_1, R1_n, PID_2, R2_n, \ldots, PID_n, Rn_n) = R_{ER}(SecKy_{ER})$
$(PID_1, R1_n, PID_2, R2_n, \ldots, PID_n, Rn_n)_{R_{ER(new)}} = SecKy_{ER(new)}$
$\in SecKy_{ER(new)}$
Step 2: Updating secret key
Edge router:
$R_{ER} \oplus R_{ER(new)} = R_{new}$
Edge router $\longrightarrow$ server $\{R_{new}\}$
Server:
$R_{new} \oplus R_{ER} = R_{ER(new)}$
Server $\longrightarrow$ database $\{R_{ER(new)}\}$
Database:
$\times R_{ER}$
$\in R_{ER(new)}$

4. Results

The proposed framework used a Raspberry Pi 4 model B, Xilinx PYNQ Z2 FPGA, Xilinx BASYS 3 FPGA, and a Jupyter notebook. The scheme used a 64-bit arbiter PUF which was deployed in both FPGAs. Figure 9 shows the experimental setup of the proposed framework. In the experimental setup, one FPGA was connected to a Raspberry Pi to make an MD or an ER. On the other hand, a standalone Raspberry Pi acted as a CS. At first, the model MC_{model} was trained and predicted the challenges of PUFs of MDs [45]. The challenges were used in the PUFs of the FPGAs and the connected Raspberry Pis collected the responses. For example, if the predicted challenges were received for MD1, the challenges were placed in the Raspberry Pi. Tx/Rx ports were connected to the Rx/Tx ports of the FPGA, which were defined in the constraints file of the PUF. The baud rate in the Raspberry Pi should be selected as the baud rate used in the PUF to define clocks per bit. By sending challenges to the FPGA from the Raspberry Pi, the PUF generated responses and the Raspberry Pi collected the responses from the Rx port. Following these steps, the responses from the group of MDs and the ER were collected. Using the CRPs, the model ML_{model} was generated.

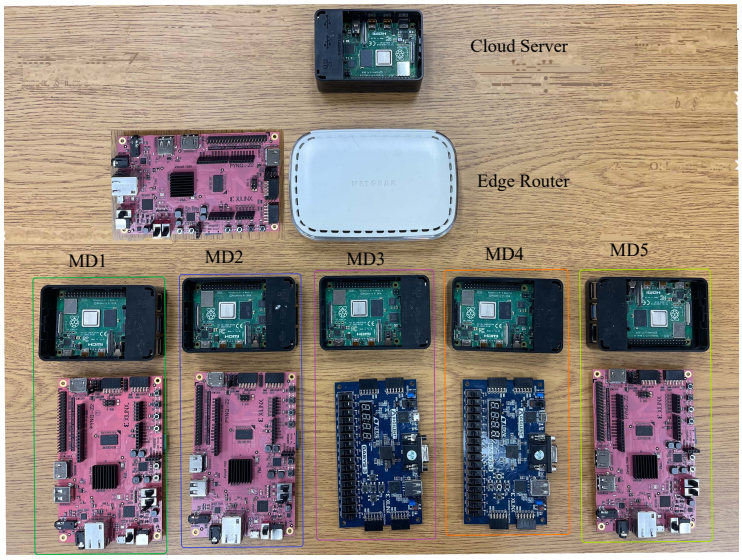

Figure 9. Experimental setup of the proposed framework.

4.1. Machine Learning Performance

In the proposed framework, five MDs were used in the group. From each MD's PUF, 487,940 CRPs were collected. All the CRPs from five MDs were used to prepare the dataset. In the dataset, a total of 2,439,700 CRPs were present. To train the model, 80% of the dataset was used as training data, and 20% of the data were used for validation. No separate data were kept for testing as there were no novel data for testing. The whole dataset was used for testing when the training was done. In the training, 16 input features were considered and one output feature. Each byte of the CRP acted as an input feature, and the PID was used as the output feature. It was a multiclass classification as there were five categories (PID) in the output feature; categorical cross-entropy was the loss function, and the regularizer L2 (0.1) was also applied. For the optimizers AdaDelta and AdaGrad, the model did not show good performance. Optimizer Nadam showed better results than AdaDelta, but the accuracy was not as good as the Adam optimizer. Table 3 shows the performance of the ML_{model} for different combinations. Each model was run for 50 epochs with batch size 5000. A few models also showed some fluctuations in their accuracy.

Table 3. ML_{model} performance of different models.

Units	Z-Score	Activation Function	Optimizer	Validation Accuracy
512-4096-4096-2048-1204	✗	ReLU	Adam	81.39
512-4096-4096-2048-1204	✓	ReLU	RMSProp	97.35
512-1024-1024-512-248	✗	ReLU	RMSProp	96.55
512-1024-1024-512-248	✓	ReLU	RMSProp	96.7
512-1024-1024-512-248	✓	ReLU	Adam	98.2
512-1024-1024-512-248	✓	ReLU	Nadam	96.65
512-4096-4096-2048-1204	✓	tanh	RMSProp	97.35
512-2048-1024-512	✓	ReLU	RMSProp	97.58
512-2048-1024-512	✓	ReLU	Adam	94.97
512-2048-1024-512	✓	tanh	Adam	97.9
512-2048-1024-512	✓	tanh	RMSProp	97.48

The best performance was found for 512-4096-4096-2048-1204 units, a ReLU activation function, and the Adam optimizer as shown in Figure 10. Moreover, a Z-score was applied to the model. The training accuracy was 99.46%, and the validation accuracy was 98.55%. While testing the whole dataset using the model, it showed 99.54% accuracy.

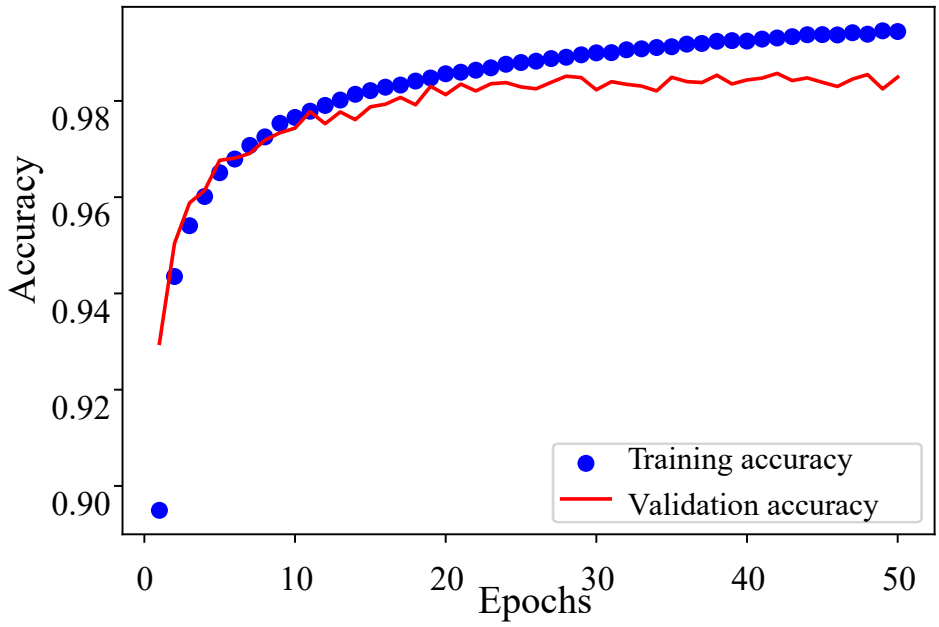

Figure 10. Performance of ML_{model}.

4.2. Computation Cost

The proposed framework is for medical devices which are resource-constrained devices. Complexity or much time consumed in operations will raise the burden on such devices. The operations performed during the authentication are lightweight and not resource hungry. The time required for completing XOR and concatenation operations is negligible and can be avoided [32]. The standard time for performing operations that are required in the proposed framework is presented in Table 4. In the proposed scheme, four hash operations, four PUF responses, one message decryption, and three model predictions are required. Among these operations, the MD is required to perform the PUF response generation twice, which is negligible, as shown in Table 4. If there is a single MD in a group, the computation time to complete the process is 2.6 ms ($4 * T_h + 4 * T_R + 1 * T_D + 3 * T_M$).

Table 4. Computation cost of the proposed framework for single MD in a group.

Notation	Description	Computation Time	MD Times	ER Times	CS Times	Total	Total Time
T_h	Hash operation [46]	0.0234 ms	0	2	2	4	0.0936 ms
T_R	PUF response [46]	0.4 µs	2	2	0	4	0.0016 ms
T_D	Decryption [47]	0.14 ms	0	1	0	1	0.14 ms
T_M	Model prediction [48]	0.75 ms	0	1	2	3	2.25 ms

In Figure 11, it can be seen that the computational cost decreases comparatively if the number of devices is increased in the group. For the addition of an MD in a group, only the server needs to additionally predict ML_{model} one more time. Each MD's addition only adds 0.75 ms of computation time and the equation of the computation cost for n number of MDs is $T_{Computation} = (1.85 + n \times 0.75)$ ms.

In the CRP update process in the ER, the computation cost is approximately 0.37 ms $(1 \times T_E + 1 \times T_D)$ [47].

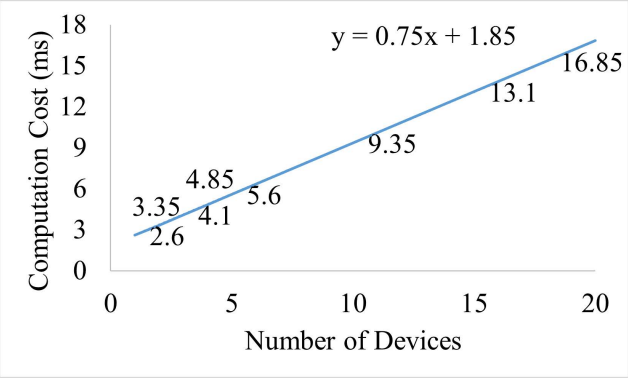

Figure 11. Computation cost trend of a group.

4.3. Communication Overhead

A single message to the CS is required to complete the authentication and data-sharing process. The CS needs not send any information in the proposed framework. Figure 12 shows the total communication cost of the proposed framework for an MD in a group. The communication overhead of an MD in a group is 104 bytes. For calculating the communication overhead, C, R, and PID are considered as having 64 bits. The hash output has 256 bits. The communication overhead is changed based on the selection of the lengths of the parameters. It can be seen that step 3 is having more overhead compared to the other steps as the step is sending multiple data after performing hash operations.

Figure 13 shows the trend of communication overhead with the increment of MDs in the group. It can be seen that only 40 bytes are required to transmit for the addition of each MD, and the equation of the communication cost is $B_{Communication} = (64 + 40 \times n)$.

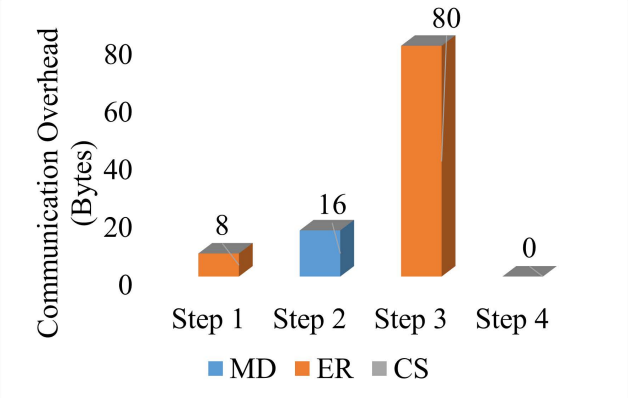

Figure 12. Communication overhead for single MD in a group.

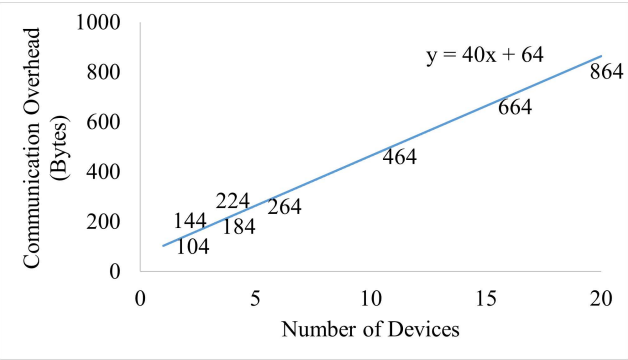

Figure 13. Communication overhead trend of a group.

The communication overhead in the CRP update process in the ER is eight bytes, as the ER only sends the response after performing an XOR operation.

4.4. Performance Comparison

Figure 14 shows the performance comparison of the proposed framework with respect to computation cost. The figure shows that the proposed method needs less computational resources than other methods. Moreover, other authentication frameworks follow a linear pattern to complete the authentication of more than one device. However, the proposed framework takes less time onward.

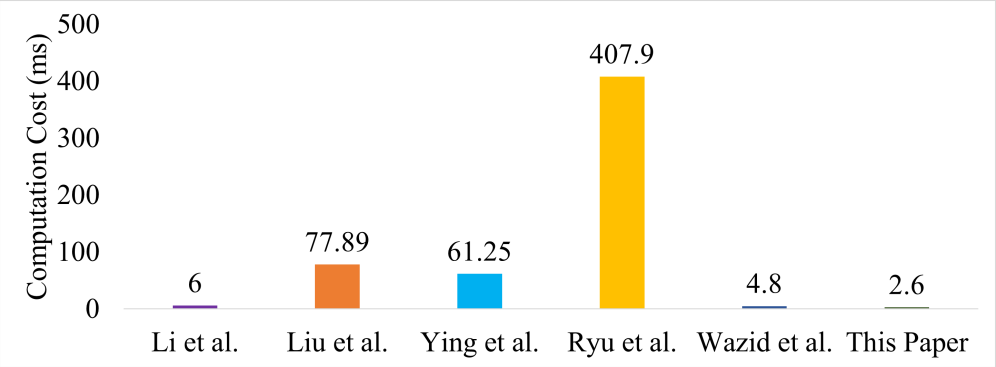

Figure 14. Computation cost comparison [25,29,32,33,37] of the proposed framework for a single device in a group.

Communication overhead comparison is presented by Figure 15. The figure depicts that the proposed framework puts less burden to the transmission medium than other existing methods. Moreover, the proposed mechanism will not follow a linear pattern like others for multiple devices authentication.

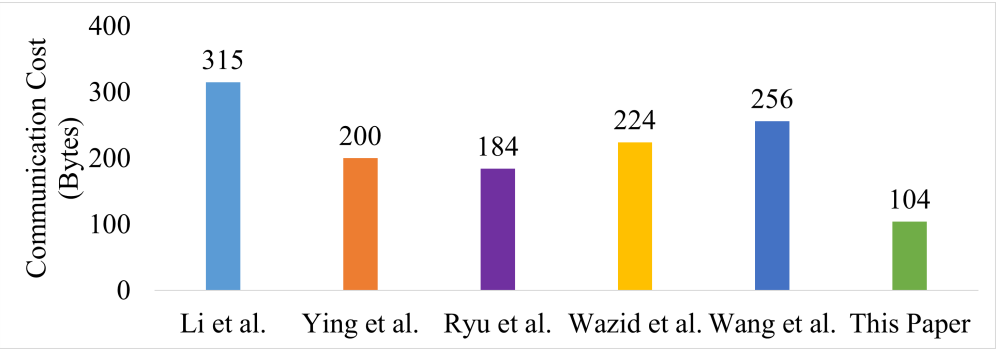

Figure 15. Communication cost comparison [25,29,32,33,37,44] of the proposed framework for a single device in a group.

5. Security Proof

In this section, how the proposed authentication is comparatively more feasible than other existing methods for an IoMT application is presented. Security resistance against known attacks is discussed using both formal and informal ways.

5.1. Formal Security Proof

In this part, the Burrows–Abadi–Needham (BAN) logic is used to show the proposed group MDs' authentication method's formal security proof [49].

5.1.1. Notations

Each interference presented by the BAN logic is organized in accordance with its relevance using the basic notations and corresponding descriptions. The following expressions are used:

- *P believes X (P* $|\equiv X$ *)*: P either believes or has the ability to think that the formula X is true.
- *P sees X (P* $\triangleleft X$*)*: P either already believes or has a substantial basis for believing that the phrase X is true.
- *P once sent X (P* $|\sim X$*)*: Although object P has already sent a message containing statement X, it is unclear if the information was sent there at the time of the process or in the past. However, in this instance, it is clear that P believes X.
- *Fresh X (#(X))*: communication X is regarded as new because it has not been addressed before the current transmission period.
- *P has complete control over X (P* $\overset{\triangleleft}{\Rightarrow} X$*)*: this happens when P has entire authority over function X and it is used in accordance with the authority's instructions.
- *Secret key between P and Q (P* $\overset{X}{\rightleftharpoons}$ *Q)*: this implies that only P and Q are aware of the secret code or methods X.

5.1.2. Inference Rules

In the BAN logic, there are several sets of inference rules with the preceding remarks:

- IR_1: <Nonce-Verification Rule>

$$\frac{P| \equiv \#(X), P| \equiv Q| \equiv | \sim X}{P| \equiv Q| \equiv X}$$

- IR_2: <Jurisdiction Rule>

$$\frac{P| \equiv P \Rightarrow X, P| \equiv Q| \equiv X}{P| \equiv X}$$

- IR_3: <Key Freshness Rule>

$$\frac{P| \equiv \#(X)}{P| \equiv \#(X, Y)}$$

- IR_4: <Shared Key Rule>

$$\frac{P| \equiv Q \overset{K}{\leftrightarrow} P, P \triangleleft \{X\}_K}{P| \equiv Q| \sim X}$$

- IR_5: <Secret Key Sharing Rule>

$$\frac{P| \equiv Q| \equiv R \overset{K}{\rightleftharpoons} R'}{P| \equiv Q| \equiv R' \overset{K}{\rightleftharpoons} R}$$

5.1.3. Initial Assumptions

The following propositions are taken into consideration when evaluating the security attribute of mutual authentication:

- A_1: CS $|\equiv$ CS $\overset{MC_{model}}{\rightleftharpoons}$ ER
- A_2: MD1 $|\equiv$ CS $\overset{MC_{model}}{\rightleftharpoons}$ ER
- A_3: CS $|\equiv$ CS $\overset{ML_{model}}{\rightleftharpoons}$ MD1
- A_4: MD1 $|\equiv$ CS $\overset{ML_{model}}{\rightleftharpoons}$ MD1

- A_5: ER $|\equiv$ ER $\overset{R1_n}{\rightleftharpoons}$ MD1
- A_6: MD1 $|\equiv$ ER $\overset{R1_n}{\rightleftharpoons}$ MD1
- A_7: CS $|\equiv$ CS $\overset{R_{ER}}{\rightleftharpoons}$ ER
- A_8: ER $|\equiv$ CS $\overset{R_{ER}}{\rightleftharpoons}$ ER

5.1.4. Idealized Form

The idealized structure of the presented framework's messages is as follows:

- I_1: ER $\rightarrow$ MD1: $\{R1_n, C1(p), N_1 \#(N_1, C1(p))\}$
- I_2: MD1 $\rightarrow$ ER: $\{Res_1, DT_1, N_1 \#(Res_1, DT_1)\}$
- I_1: ER $\rightarrow$ CS: $\{Res_1, R_{ER}, PID_1, PID_{ER}, C1(p), C_{n1}, DT_1, \#(R_{ER}, Res_1, C_{n1}, C1(p))\}$

5.1.5. Goals of Proposed Framework

In order to have successful authentication, the following requirements must be met:

- G_1: ER $|\equiv$ MD1 $|\equiv <$ER $\overset{R1_n}{\longleftrightarrow}$ MD1$>$
- G_2: CS $|\equiv$ MD1 $|\equiv <$CS $\overset{Res_1}{\longleftrightarrow}$ MD1$>$

5.1.6. Formal Verification Proof

We now use the preceding inference techniques, idealized form, and objectives to evaluate the framework's MD authentication procedure.

- FV_1: from I_1 and by practicing IR_1, IR_2 and IR_3, it is desired to obtain (1) and achieve goal G_1:

$$\frac{MD1| \equiv \# < N_1, C1(p) >, MD1| \equiv ER| \equiv \#(N_1, C1(p)), MD1| \equiv ER \Rightarrow (R1_n), MD1| \equiv ER| \equiv (R1_n)}{MD1| \equiv \#(N_1, C1(p), R1_n), MD1| \equiv (R1_n)} \quad (1)$$

- FV_2: from I_1 and by practicing IR_1, IR_3 and IR_5, it is desired to obtain (2):

$$\frac{ER| \equiv \#(Res_1, DT_1), ER| \equiv MD1 \overset{Res_1}{\rightleftharpoons} CS, ER| \equiv \sim (N_1)}{ER| \equiv \#(Res_1, DT_1, N_1), ER| \equiv CS| \sim Res_1} \quad (2)$$

- FV_3: from I_3 and by practicing IR_1, IR_3, and IR_5, it is desired to obtain (3) and achieve goal G_2:

$$\frac{CS| \equiv \#(R_{ER}, C_{n1}, DT_1), CS| \equiv MD \overset{Res_1}{\rightleftharpoons} CS, CS| \equiv MD \overset{R_{ER}}{\rightleftharpoons} CS, CS| \equiv ER| \equiv \sim (PID_1, PID_{ER})}{CS| \equiv \#(PID_1, PID_{ER}, DT_1, R_{ER}, Res_1, C_{n1}, C1(p)), CS| \equiv MD \overset{R_{ER}}{\rightleftharpoons} CS, CS| \equiv MD \overset{Res_1}{\rightleftharpoons} CS} \quad (3)$$

5.2. Informal Security Proof

This section demonstrates how the developed authentication system can implement security measures when taking into account the capacity of an adversary to alter and listen in on the sent data through public networks.

5.2.1. Impersonation Attacks

The proposed framework can resist MD, ER, and CS impersonation attacks. An adversary can try to act as an MD to alter or provide false data. To act as a valid MD, an attacker needs to generate the correct response of a challenge. However, the ER is responsible for providing a partial challenge. Therefore, it is not possible to impersonate an MD. To impersonate the ER, an attacker needs to get the secret responses of the MDs. An attacker can only get the secret response if they can get the decryption key. Moreover, the secret encryption key is updated periodically, which defends against the theft of the encryption key. Furthermore, the information of the MDs is not stored in plaintext but rather formed in a model. Moreover, the server is not requested to provide data in the framework. Therefore, it can be said that the proposed framework can resist impersonation attacks.

5.2.2. Side-Channel Attacks

Side-channel attacks are usually performed by measuring computation time, power analysis, etc., operations. Moreover, secret keys stored in the memory of the devices raise the chance to be affected by side-channel attacks [27]. To avoid this, a PUF was used to generate the responses. In the proposed method, an XOR operation was used to fabricate the response using random health data and a random nonce, which was not stored in the memory of the device nor used further by the MD.

5.2.3. Modeling Attacks

In modeling attacks, an adversary tries to grab the pattern of secret keys to build a model to predict the next keys/responses of the MD to disrupt the system. It is possible to resist modeling attacks to protect the CRP interface. By providing an additional block, hash function, etc., it is also possible to mask the interface [50]. The proposed method used MC_{model} to generate a partial response which was completed by the MD itself and the challenge was not going out. Furthermore, the generated response was also masked before sending to the ER. Thus, the proposed scheme is able to resist modeling attacks. Figure 16 shows the process of hiding the challenge.

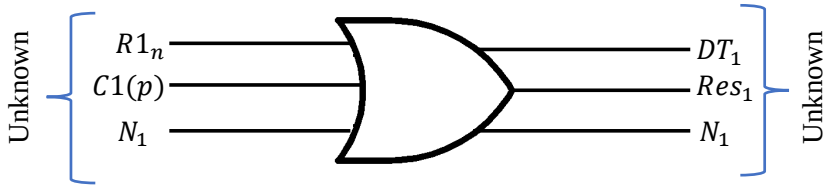

Figure 16. Hiding CRP interfaces to resist modeling attack.

5.2.4. Physical Attacks

Physical attacks can be conducted by accessing the secret keys from the devices' memory. If there is any attempt by an adversary to tamper with the PUF-based device, then the device will be damaged, and it will generate incorrect responses [39]. Moreover, neither challenges nor responses are stored in the devices. Thus, it can be said the framework is not affected by physical attacks.

5.2.5. Dos Attacks

An attacker can try to perform a DoS attack so that the regular service of the node is interrupted, and the device could go out of service. In this attack, the adversary tries to consume the limited resources of the device. To avoid this attack, the MD is able to understand whether the authentication request is from the legitimate ER by checking $R1_n$. If it does not match, it does not process the authentication request, and the PUF response generation is not a time- or resource-consuming operation. The proposed method manages a gray list for such kinds of requests to avoid DoS attacks.

5.2.6. Replay Attack

A replay attack is when an attacker sends repeated or falsely delayed lawful data transmission. Replay attacks can be resisted by using clock synchronization or a random nonce method. Li et al. [51] stated that clock synchronization is still an open research area for communication in a wireless sensor network. This paper adopted a random nonce N_1, a random challenge, a random response, etc., in each message to defend against replay attacks. When all of these factors are taken into account, it can be said that the proposed solution can block replay attacks by employing random numbers, and its effectiveness is unaffected by clock synchronization issues.

5.2.7. Eavesdropping Attack

Due to the distinctive characteristics of each PUF, the CRP ensures that only the legitimate device and server can interact with one another. Because the server stores only the model throughout the enrollment process, and the server has no information about the responses, eavesdroppers cannot replicate it without having access to the model and CRPs [52].

5.2.8. Man-in-the-Middle Attack

Each time, different responses are generated based on masked or partial challenges from the ER. The generated response is also masked using DT and nonce. Moreover, the ER uses hash operations to mask all the collected responses and data from all the MDs of the group. Both the MDs and the ER generate random messages to prevent man-in-the-middle (MITM) attacks. Therefore, inside intruders do not have a chance to execute a MITM assault unless they are familiar with CRPs, XOR functions, and random nonces [52].

5.2.9. Anonymous Identity

To avoid exposing the real identity of a node, pseudoidentity is used instead of the original ID. The uses of PID are resistant to attackers who try to identify the original owner of the device. Furthermore, it is not shared in plaintext. The proposed framework can preserve privacy by maintaining an anonymous identity.

5.2.10. Forward Secrecy

Forward secrecy's primary goal is to guarantee that previously established transaction keys remain safe in the event that the keys are compromised. In the proposed framework, MDs do not use any prestored key to authenticate the device. The MDs generate responses where challenges are decided by the partial challenge and PIDs by steps 1 and 2. Moreover, the response is masked using nonce and MD data, which are random as shown in step 2. Therefore, the secret key cannot be identified without the random numbers. Furthermore, for the next time authentication and data masking, different challenges, responses, nonce, data, etc., will be used, which keeps the forward secrecy.

The discussion has illustrated that the proposed group of medical devices' authentication framework is resilient to known security threats.

6. Conclusions and Future Directions

Patients in a hospital pass a crucial time in their life. The diagnosis and treatment should be provided in an accurate manner as per the direction of doctors and experts. The IoMT is making the treatment process timelier, which helps patients, doctors, nurses, etc. However, the process could backfire and even could cause the death of patients if the IoMT data were fabricated. To ensure patients' safety, secure communication is a must for the IoMT system. To protect the security of the IoMT system in a hospital environment in a smart city, the paper proposed an authentication framework where a group of devices were authenticated in a single message transmission. Moreover, the devices did not need to send health data separately; the data could be sent in the authentication request. The proposed framework showed a better resistance against security attacks compared to the state-of-the-art methods. The proposed framework is lightweight, and a low communication cost is required with respect to other works. In the future, federated learning along with blockchain will be incorporated to introduce more security features. Furthermore, group key agreement will be considered to make the proposed framework feasible for mobile applications.

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

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Bajic, B.; Rikalovic, A.; Suzic, N.; Piuri, V. Industry 4.0 implementation challenges and opportunities: A managerial perspective. *IEEE Syst. J.* **2021**, *15*, 546–559. [CrossRef]
2. Rikalovic, A.; Suzic, N.; Bajic, B.; Piuri, V. Industry 4.0 implementation challenges and opportunities: A technological perspective. *IEEE Syst. J.* **2022**, *16*, 2797–2810. [CrossRef]
3. Sadhu, P.K.; Yanambaka, V.P.; Mohanty, S.P.; Kougianos, E. Easy-Sec: PUF-based rapid and robust authentication framework for the internet of vehicles. *arXiv* **2022**, arXiv:2204.07709.
4. Khan, M.A.; Siddiqui, M.S.; Rahmani, M.K.I.; Husain, S. Investigation of big data analytics for sustainable smart city development: An emerging country. *IEEE Access* **2022**, *10*, 16028–16036. [CrossRef]
5. Khalil, U.; Mueen-Uddin; Malik, O.A.; Hussain, S. A blockchain footprint for authentication of IoT-enabled smart devices in smart cities: State-of-the-art advancements, challenges and future research directions. *IEEE Access* **2022**, *10*, 76805–76823. [CrossRef]
6. Sadhu, P.; Yanambaka, V.P.; Abdelgawad, A.; Yelamarthi, K. NAHAP: PUF-based three factor authentication system for internet of medical things. *IEEE Consum. Electron. Mag.* **2022**. [CrossRef]
7. Hernandez, S.; Raison, M.; Torres, A.; Gaudet, G.; Achiche, S. From on-body sensors to in-body data for health monitoring and medical robotics: A survey. In Proceedings of the Global Information Infrastructure and Networking Symposium (GIIS), Montreal, QC, Canada, 15–19 September 2014; pp. 1–5.
8. Noguchi, H.; Mori, T.; Sato, T. Framework for search application based on time segment of sensor data in home environment. In Proceedings of the Seventh International Conference on Networked Sensing Systems (INSS), Kassel, Germany, 15–18 June 2010; pp. 261–264.
9. Internet of Medical Things (IoMT) Market by Component, Platform, Connectivity Devices, Application and Is Expected to Reach USD 1,84,592.31 Million by 2028. Available online: https://www.marketwatch.com/press-release/internet-of-medical-things-iomt-market-by-component-platform-connectivity-devices-application-and-is-expected-to-reach-usd-18459231-million-by-2028-2022-04-26 (accessed on 22 June 2022).
10. Internet of Medical Things Revolutionizing Healthcare. Available online: https://aabme.asme.org/posts/internet-of-medical-things-revolutionizing-healthcare/ (accessed on 1 April 2021).
11. What Is the Internet of Medical Things (IoMT)? Available online: https://mobius.md/2019/03/06/what-is-the-iomt/ (accessed on 22 June 2022).
12. Sadhu, P.K.; Yanambaka, V.P.; Abdelgawad, A.; Yelamarthi, K. Prospect of internet of medical things: A review on security requirements and solutions. *Sensors* **2022**, *22*, 5517. [CrossRef]
13. Meng, W.; Cai, Y.; Yang, L.T.; Chiu, W.Y. Hybrid emotion-aware monitoring system based on brainwaves for internet of medical things. *IEEE Internet Things J.* **2021**, *8*, 16014–16022. [CrossRef]
14. Masud, M.; Gaba, G.S.; Alqahtani, S.; Muhammad, G.; Gupta, B.B.; Kumar, P.; Ghoneim, A. A lightweight and robust secure key establishment protocol for internet of medical things in COVID-19 patients care. *IEEE Internet Things J.* **2021**, *8*, 15694–15703. [CrossRef]
15. Healthcare IT sEcurity Budgets Aren'T Keeping Pace with IoMT Threats. Available online: https://www.ivanti.com/blog/healthcare-it-security-budgets-aren-t-keeping-pace-with-iomt-threats (accessed on 10 October 2022).
16. Chen, C.M.; Chen, Z.; Kumari, S.; Lin, M.C. LAP-IoHT: A lightweight authentication protocol for the internet of health things. *Sensors* **2022**, *22*, 5401. [CrossRef]
17. Elmitwalli, E.; Ni, K.; Köse, S. Machine learning attack resistant area-efficient reconfigurable Ising-PUF. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2022**, *30*, 526–538. [CrossRef]
18. Wang, A.; Tan, W.; Wen, Y.; Lao, Y. NoPUF: A novel PUF design framework toward modeling attack resistant PUFs. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2021**, *68*, 2508–2521. [CrossRef]
19. Kroeger, T.; Cheng, W.; Guilley, S.; Danger, J.L.; Karimi, N. Assessment and mitigation of power side-channel-based cross-PUF attacks on arbiter-PUFs and their derivatives. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2022**, *30*, 187–200. [CrossRef]

20. Wisiol, N.; Thapaliya, B.; Mursi, K.T.; Seifert, J.P.; Zhuang, Y. Neural network modeling attacks on arbiter-PUF-based designs. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 2719–2731. [CrossRef]
21. Olowononi, F.O.; Rawat, D.B.; Liu, C. Resilient machine learning for networked cyber physical systems: A survey for machine learning security to securing machine learning for CPS. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 524–552. [CrossRef]
22. Al-Dhief, F.T.; Latiff, N.M.A.; Malik, N.N.N.A.; Salim, N.S.; Baki, M.M.; Albadr, M.A.A.; Mohammed, M.A. A survey of voice pathology surveillance systems based on internet of things and machine learning algorithms. *IEEE Access* **2020**, *8*, 64514–64533. [CrossRef]
23. Habib, M.; Wang, Z.; Qiu, S.; Zhao, H.; Murthy, A.S. Machine learning based healthcare system for investigating the association between depression and quality of life. *IEEE J. Biomed. Health Inform.* **2022**, *26*, 2008–2019. [CrossRef]
24. Guezzaz, A.; Asimi, Y.; Azrour, M.; Asimi, A. Mathematical validation of proposed machine learning classifier for heterogeneous traffic and anomaly detection. *Big Data Min. Anal.* **2021**, *4*, 18–24. [CrossRef]
25. Li, J.; Su, Z.; Guo, D.; Choo, K.K.R.; Ji, Y. PSL-MAAKA: Provably secure and lightweight mutual authentication and key agreement protocol for fully public channels in internet of medical things. *IEEE Internet Things J.* **2021**, *8*, 13183–13195. [CrossRef]
26. Amintoosi, H.; Nikooghadam, M.; Shojafar, M.; Kumari, S.; Alazab, M. Slight: A lightweight authentication scheme for smart healthcare services. *Comput. Electr. Eng.* **2022**, *99*, 107803. [CrossRef]
27. Siddiqi, M.A.; Doerr, C.; Strydis, C. IMDfence: Architecting a secure protocol for implantable medical devices. *IEEE Access* **2020**, *8*, 147948–147964. [CrossRef]
28. Hwang, Y.W.; Lee, I.Y. A study on CP-ABE-based medical data sharing system with key abuse prevention and verifiable outsourcing in the IoMT environment. *Sensors* **2020**, *20*, 4934. [CrossRef] [PubMed]
29. Liu, X.; Yang, X.; Luo, Y.; Zhang, Q. Verifiable multi-keyword Search encryption scheme with anonymous key generation for medical internet of things. *IEEE Internet Things J.* **2021**, *9*, 22315–22326. [CrossRef]
30. Li, H.; Yu, K.; Liu, B.; Feng, C.; Qin, Z.; Srivastava, G. An efficient ciphertext-policy weighted attribute-based encryption for the internet of health things. *IEEE J. Biomed. Health Inform.* **2022**, *26*, 1949–1960. [CrossRef] [PubMed]
31. Huang, P.; Guo, L.; Li, M.; Fang, Y. Practical privacy-preserving ECG-based authentication for IoT-based healthcare. *IEEE Internet Things J.* **2019**, *6*, 9200–9210. [CrossRef]
32. Ying, B.; Mohsen, N.R.; Nayak, A.A. Efficient authentication protocol for continuous monitoring in medical sensor networks. *IEEE Open J. Comput. Soc.* **2021**, *2*, 130–138. [CrossRef]
33. Ryu, J.; Oh, J.; Kwon, D.; Son, S.; Lee, J.; Park, Y.; Park, Y. Secure ECC-based three-factor mutual authentication protocol for telecare medical information system. *IEEE Access* **2022**, *10*, 11511–11526. [CrossRef]
34. Al-Zubaidie, M.; Zhang, Z.; Zhang, J. RAMHU: A new robust lightweight scheme for mutual users authentication in healthcare applications. *Secur. Commun. Netw.* **2019**, *2019*, 3263902. [CrossRef]
35. Padinjappurathu Gopalan, S.; Chowdhary, C.L.; Iwendi, C.; Farid, M.A.; Ramasamy, L.K. An efficient and privacy-preserving scheme for disease prediction in modern healthcare systems. *Sensors* **2022**, *22*, 5574. [CrossRef]
36. de Marcos, L.; Martínez-Herráiz, J.J.; Junquera-Sánchez, J.; Cilleruelo, C.; Pages-Arévalo, C. Comparing machine learning classifiers for continuous authentication on mobile devices by keystroke dynamics. *Electronics* **2021**, *10*, 1622. [CrossRef]
37. Wazid, M.; Singh, J.; Das, A.K.; Shetty, S.; Khan, M.K.; Rodrigues, J.J. ASCP-IoMT: AI-enabled lightweight secure communication protocol for internet of medical things. *IEEE Access* **2022**, *10*, 57990–58004. [CrossRef]
38. Alladi, T.; Chamola, V.; Naren. HARCI: A two-way authentication protocol for three entity healthcare IoT networks. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 361–369. [CrossRef]
39. Gope, P.; Gheraibia, Y.; Kabir, S.; Sikdar, B. A secure IoT-based modern healthcare system with fault-tolerant decision making process. *IEEE J. Biomed. Health Inform.* **2021**, *25*, 862–873. [CrossRef] [PubMed]
40. Lee, T.F.; Ye, X.; Lin, S.H. Anonymous dynamic group authenticated key agreements using physical unclonable functions for internet of medical things. *IEEE Internet Things J.* **2022**, *9*, 15336–15348. [CrossRef]
41. Awad Abdellatif, A.; Samara, L.; Mohamed, A.; Erbad, A.; Chiasserini, C.F.; Guizani, M.; O'Connor, M.D.; Laughton, J. MEdge-Chain: Leveraging edge computing and blockchain for efficient medical data exchange. *IEEE Internet Things J.* **2021**, *8*, 15762–15775. [CrossRef]
42. Lin, P.; Song, Q.; Yu, F.R.; Wang, D.; Guo, L. Task offloading for wireless VR-enabled medical treatment with blockchain security using collective reinforcement learning. *IEEE Internet Things J.* **2021**, *8*, 15749–15761. [CrossRef]
43. Egala, B.S.; Pradhan, A.K.; Badarla, V.R.; Mohanty, S.P. Fortified-chain: A blockchain-based framework for security and privacy-assured internet of medical things with effective access control. *IEEE Internet Things J.* **2021**, *8*, 11717–11731. [CrossRef]
44. Wang, W.; Chen, Q.; Yin, Z.; Srivastava, G.; Gadekallu, T.R.; Alsolami, F.; Su, C. Blockchain and PUF-based lightweight authentication protocol for wireless medical sensor networks. *IEEE Internet Things J.* **2022**, *9*, 8883–8891. [CrossRef]
45. Sadhu, P.K.; Yanambaka, V.P. MC-PUF: A robust lightweight controlled physical unclonable function for resource constrained environments. In Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Nicosia, Cyprus, 4–6 July 2022; pp. 452–453. [CrossRef]
46. Alladi, T.; Chakravarty, S.; Chamola, V.; Guizani, M. A lightweight authentication and attestation scheme for in-transit vehicles in IoV scenario. *IEEE Trans. Veh. Technol.* **2020**, *69*, 14188–14197. [CrossRef]

47. Pravinchandra, M.M.; Diwanji, H.M.; Shah, J.S.; Kotak, H. Performace analysis of encryption and decryption using genetic based cancelable non-invertible fingerprint based key in MANET. In Proceedings of the International Conference on Communication Systems and Network Technologies, Rajkot, India, 11–13 May 2012; pp. 357–361. [CrossRef]
48. Sadhu, P.K.; Yanambaka, V.P.; Abdelgawad, A. MC-Multi PUF based lightweight authentication framework for internet of medical things. In Proceedings of the IEEE 8th World Forum on Internet of Things (WF-IoT), Yokohama, Japan, 26 October–11 November 2022; pp. XX–YY.
49. Burrows, M.; Abadi, M.; Needham, R. A logic of authentication. *ACM Trans. Comput. Syst.* **1990**, *8*, 18–36. [CrossRef]
50. Yao, J.; Pang, L.; Su, Y.; Zhang, Z.; Yang, W.; Fu, A.; Gao, Y. Design and evaluate recomposited OR-AND-XOR-PUF. *IEEE Trans. Emerg. Top. Comput.* **2022**, *10*, 662–677. [CrossRef]
51. 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.* **2020**, *14*, 39–50. [CrossRef]
52. Yıldız, H.; Cenk, M.; Onur, E. PLGAKD: A PUF-based lightweight group authentication and key distribution protocol. *IEEE Internet Things J.* **2021**, *8*, 5682–5696. [CrossRef]