A Lightweight and Robust Secure Key Establishment Protocol for Internet of Medical Things in COVID-19 Patients Care

Journal:	*IEEE Internet of Things Journal*
Manuscript ID	IoT-14289-2020
Manuscript Type:	Special Issue on Towards Intelligent Internet of Medical Things and Its COVID-19 Applications and Beyond
Date Submitted by the Author:	15-Oct-2020
Complete List of Authors:	Masud, Mehedi; Taif University, College of Computers and Information Technology Gaba, Gurjot; Electronics and Electrical Engineering Kumar, Pardeep; Swansea University, Department of Computer Science Muhammad, Ghulam; King Saud University College of Computer and Information Sciences, Dept. of Computer Engineering Gupta, Brij; National Institute of Technology Kurukshetra,
Keywords:	Cyber-Physical Systems < Sub-Area 3: Services, Applications, and Other Topics for IoT, eHealth and mHealth < Sub-Area 3: Services, Applications, and Other Topics for IoT, Security and Privacy < Sub-Area 3: Services, Applications, and Other Topics for IoT

A Lightweight and Robust Secure Key Establishment Protocol for Internet of Medical Things in COVID-19 Patients Care

Mehedi Masud, *Senior Member, IEEE,* Gurjot Singh Gaba, *Member, IEEE,* and Ghulam Muhammad, *Senior Member, IEEE,* B. B. Gupta, and Pardeep Kumar, *Member, IEEE*

***Abstract*—After the outbreak of COVID-19, the Internet of Medical Things (IoMT) has enabled the doctors to remotely perform their duties like diagnosing the patients, controlling the medical equipment, and monitoring the health of the quarantined patients in real-time through their digital devices. In IoMT, the security has paramount importance because the nodes exchange critical information over the vulnerable wireless medium. Attackers can exploit these vulnerabilities to eavesdrop on the vaccine formula, modify the antigen test results, switch off ventilators, etc. which can result in irreparable losses and fatalities. The virtual medical facilities must be protected from adversarial threats through secure sessions to ensure continuous care of COVID-19 affected patients. To address the security threat in IoMT, this paper proposes a lightweight and physically secure mutual authentication and secret key establishment protocol that makes use of Physical Unclonable Functions (PUF) to enable the network devices to verify the legitimacy of the doctor (user) and sensor node before establishing a session key. PUF also protects the sensor nodes deployed in an unattended and hostile environment from tampering, cloning, and side-channel attacks. The proposed protocol exhibits all the necessary security properties required to protect the IoMT networks like authentication, confidentiality, integrity, anonymity, untraceability, and so on. The formal (AVISPA) and informal security analysis demonstrate its robustness against attacks like impersonation, replay, a man in the middle, etc.**

***Index Terms*—Internet of Medical Things, Cyber-physical System, COVID-19, Security, Key management.**

I. Introduction

Since the outbreak of COVID-19 in December 2019 the healthcare professionals around the word giving serious efforts and putting their lives at stake to cure the patients suffering from COVID-2019. To enable the doctors to perform their duties safely, the hospital authorities are trying to transform their physical medical practices to telemedical practices. Internet of Medical Things (IoMT) has evolved from the Internet of Things (IoT), where the doctor can use the wireless media to communicate with IoT enabled sensor nodes such as smart thermometers, smart ventilators, and so on [1]. The IoT enabled sensor nodes collect, analyze, and disseminate the health reports of the patients in real-time to the doctors [2] and enable them to diagnose, treat, and monitor the COVID-19 infected patients remotely. There are many challenges in the implementation and utilization of IoMT networks. The substantial issue lies with the several adversarial threats, security and privacy of the sensitive information of patients and healthcare institutions. The possible adversarial threats are eavesdropping [3], data breach [4], and denial of service (DoS) [5]. The situation becomes more adverse in IoMT since medical users and vendors have limited awareness of security threats [6] and possible remedies [7]. As per the reports, the adversary prefers to breach the data that mainly includes patients names, bank details, medical history, and insurance information, etc.

The attackers are gaining more interest in IoMT due to the exponential growth of its market [8]. The adversaries exploits the vulnerabilities of the systems and networks to conduct cyber attacks and achieve their malicious desires. The absence of robust mutual authentication and key establishment scheme is the key factor attracting the adversaries towards IoMT networks [9]. The existing mutual authentication schemes are not directly applicable to IoMT networks as they are computation and communication expensive and can drain the precious energy reserves of IoT sensor nodes [10]. Moreover, most of the schemes do not consider the hostile environment of deployment of sensor nodes and become vulnerable to physical, cloning, and side-channel attacks [11]. Therefore, IoMT networks need a practical mutual authentication and secret key establishment approach that can provide robust security in all environments while being lightweight in computation and communication.

II. Related Work

Wolf et al. [12] emphasized on the various security issues in the Cyber-Physical systems and proposed a dynamic password-based user authentication model to restrict the unauthorized access to wireless sensor networks. Yeh et al. [13] introduced an asymmetric cryptography based authentication protocol for resource constrained networks but the scheme did not accomplish mutual authentication and is computationally

Mehedi Masud is with the College of Computers and Information Technology, Taif University, Taif, Saudi Arabia, e-mail: mmasud@tu.edu.sa

Gurjot Singh Gaba is with the Department of Electronics and Electrical Engineering, Lovely Professional University, Phagwara 144411, India, e-mail: gurjot.17023@lpu.co.in

Pardeep Kumar is with the Department of Computer Science, Swansea University SA1 8EN, U.K., e-mail: pardeep.kumar@swansea.ac.uk

Ghulam Muhammad is with the Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Saudi Arabia, email: ghulam@ksu.edu.sa

B. B. Gupta is with the Department of Computer Engineering, National Institute of Technology, India and Department of Computer Science and Information Engineering, Asia University, Taiwan, email: bbgupta@nitkkr.ac.in

Manuscript received Month dd, 20XX; revised Month dd, 20xx.

expensive due to the use of public key cryptography. Khalil et al. [14] analysed that few user authentication and key agreement schemes for IoT environment are subjected to physical, man-in-the-middle and replay attacks. Wu et al. [15] revealed that a dynamic identity oriented user authentication protocol developed by Hsieh and Leu [16] has taken inadequate security measures to protect the session key, and is prone to physical attacks on sensor nodes. [17] introduced a novel lightweight approach to securing the IoT node communications with the cloud, however, the scheme is vulnerable to MITM, replay, and impersonation attacks. Li et al. [18] designed a 3-factor user authentication scheme acquiring inputs such as user identity, password, and biometrics for its operation. As aforementioned, it is resource expensive since it computes 3 factors and exchanges 2688 bits. To reduce the computation expenditure, Esfahani et al. [19] introduced a lightweight scheme that makes use of only one way hash function and bit-wise XOR operations. But the security analysis is inadequate to declare the protocol as robust against attacks. A hybrid model of authentication and key establishment is proposed in [20]. Paliwal [21] proposed a mutual authentication and key agreement approach with the use of lightweight one-way cryptography hash operations. The scheme protects the anonymity of identities but encounters difficulties in preserving privacy.

In summary, most of the protocols are insecure against significant adversarial threats, fail to exhibit essential security properties to keep the communication secure, and are computationally and communication expensive. The IoMT networks carry very critical information, wherein a mild attack can result in fatalities. Therefore, the IoMT networks must deploy a strong mutual authentication and physically protected lightweight key establishment protocol to enable only the legitimate devices to communicate using a secured session key.

A. Contribution

Considering the importance of security and privacy of the communications in IoMT networks, this article proposes a lightweight, robust, and physically secure **M**utual **A**uthentication and **S**ecret **K**ey (MASK) establishment protocol for securing the sensitive health information of the patients. The protocol only permits the legitimate users to establish a secure connection with the IoT enabled medical equipment's (sensor nodes) for fetching the health reports of the patients. The protocol uses lightweight cryptography primitives, such as one-way hash function, nonce, and bit-wise XOR operations. The protocol exhibits all essential security properties like authentication, confidentiality, integrity, anonymity, etc. Reliability of the proposed protocol under compromised conditions is verified using formal and informal security analysis.

III. System Model

A smart healthcare facility where COVID-19 affected patients take care constitutes of sensor nodes, gateway, and the user (e.g., doctor). The sensor nodes are integrated with the medical equipment to monitor the patients' health, etc. Gateway is used to relay the information between doctor and IoT enabled sensor nodes whereas the user (e.g., doctor) is interested to receive real time information from sensor nodes to take decisive actions regarding the patients' treatment.

Table I
Symbols, Abbreviations, and Operators description

Notation	Definition
C_E^N	C: Challenge, N: Number, E: Entity
R_E^N	R: Response, N: Number, E: Entity
P_D, P_{SN}	Physically Unclonable Function
D_{ID}^H	Unique identity of doctor issued by Hospital
D_{LN}^{MCC}	Unique license no. issued by Medical Council
SN_{IEI}	International equipment identity of Sensor Node
TID_U, TID_{SN}	Temporary identity of User and Sensor Node
PW_U	User password
N_U, N_G, N_{SN}	Nonce
SK_U, SK_{SN}	Session Key
F	Strong cryptography function
h(.)	one way cryptography hash function
$\oplus$, $\|$	Bit-wise XOR and concatenation operator
$A \equiv^? B$	Is A identical to B?

A. Security and other goals

MASK protocol attains all the prominent security properties [10].The proposed protocol also introduces a mechanism to protect the devices from physical tampering resulting in prevention from aforesaid attacks. MASK protocol also complies with the essential property to claim the protocol as *efficient*. The proposed protocol use lightweight cryptography operations and minimum message exchanges while still providing the highest level of security.

B. Physically Unclonable Function (PUF)

Physically Unclonable Functions (PUF) are recommended as a solution to secure the hardware from adversarial threats. PUF is primarily endorsed for authenticating the edge devices having limited computing power, small memory, and so forth. Since the IoT devices transmit crucial data, therefore it becomes a necessity to integrate strong authentication algorithms to protect the devices from unauthorised abuses. However, being resource constrained, it becomes difficult for the edge devices to execute algorithms with acute requirements. As a result, the authentication of resource constrained edge (IoT) devices becomes a challenge.

IV. Proposed Security Framework

To prevent the IoMT networks from adversarial threats, the proposed protocol empowers the devices to perform mutual authentication followed by secret key establishment. The proposed protocol is executed in three phases, namely, *user registration*, *device registration*, and *mutual authentication and secret key establishment*. Table 1 provides the notations that have been used throughout the paper. Prior to the discussion on the three phases, assumptions are stated.

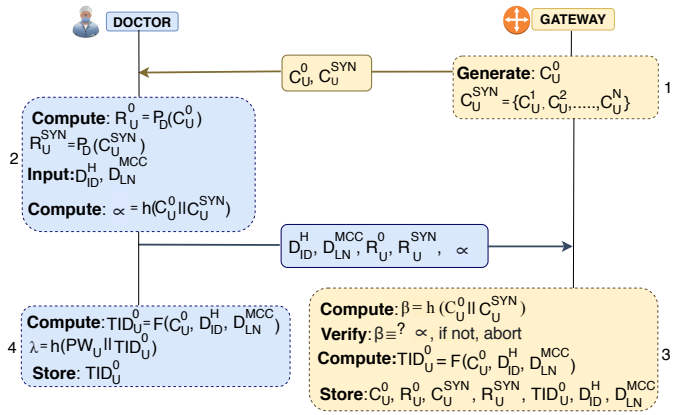

Figure 1. User Registration Phase.

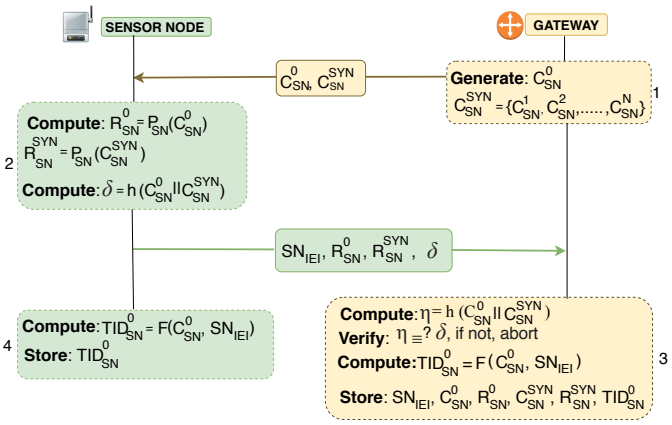

Figure 2. Sensor Node Registration Phase.

A. User Registration Phase

The user('doctor') has to register its trusted device in the gateway for obtaining real time health information of the patients. The whole registration process is demonstrated through fig. 1 as follows.

Step 1: Initially, the gateway generates a random challenge C_U^0 for the current registration process along with another set of challenges $C_U^{SYN} = \{C_U^1, C_U^2, ..., C_U^N\}$. C_U^{SYN} consists of many random challenges that will be used by the gateway in future to verify the user device. The gateway forms the message $\{C_U^0, C_U^{SYN}\}$ and sends it to doctor's device.

Step 2: Upon receiving the message, $\{C_U^0, C_U^{SYN}\}$, doctor's device triggers the PUF to generate the response, R_U^0 $(= P_D(C_U^0)$ and R_U^{SYN} $(= P_D(C_U^{SYN})$. R_U^{SYN} comprises of responses to many random challenges $(= R_U^1, R_U^2, ..., R_U^N)$ that will be used by the gateway to authenticate the doctor's device in future correspondences. Post response generation by PUF, doctor inputs his/her unique identity details, D_{ID}^H, and D_{LN}^{MCC}. Thereafter, doctor's device compute $\alpha = h(C_U^0 \| C_U^{SYN})$ to enable the gateway to verify the association between received responses $(R_U^0,\ R_U^{SYN})$ and sent challenges (C_U^0, C_U^{SYN}). Lastly, the doctor's device compose a message including D_{ID}^H, D_{LN}^{MCC}, R_U^0, R_U^{SYN}, α and sends it to gateway for requesting authorization to communicate with sensor nodes.

Step 3: After obtaining the response from user, gateway computes $\beta = h(C_U^0 \| C_U^{SYN})$ and verify the identicalness between β and α; the gateway terminates the session if result is distinct, otherwise computes $TID_U^0 = F(C_U^0, D_{ID}^H, D_{LN}^{MCC})$. At the end of the registration, gateway stores the C_U^0, R_U^0, C_U^{SYN}, R_U^{SYN}, TID_U^0, D_{ID}^H, D_{LN}^{MCC} for future communication with user.

Step 4: Likewise, doctors's device also computes and stores $TID_U^0 = F(C_U^0, D_{ID}^H D_{LN}^{MCC})$. Apart from TID_U^0, doctors' device also computes and stores user verification code λ, which is a function of user password (PW_U) and temporary identity of user (TID_U^0). The user will be prompted to enter the password (PW_U) every time he/she wishes to access the device for the particular application.

Remark 1: It is worth noting that TID_U^0 is a function of C_U^0, D_{ID}^H, and D_{LN}^{MCC}. The use of new challenge C_U^N for every session assures distinct temporal identity TID_U^N for every session, thus achieving untraceability. Moreover, instead of real identity of user $\{D_{ID}^H,\ D_{LN}^{MCC}\}$, temporary identity TID_U^0 is used to preserve user anonymity during message exchanges. Being resource constrained, the user device only executes few lightweight cryptography operations (hash and PUF) during registration to preserve resources. As no secret credentials are stored in the user device, therefore it cannot reveal any sensitive information (e.g., PW_U) despite being attacked. Besides, the protocol prevents the unauthorized access to the device by enabling the user authentication code, λ. The authentication code λ is a function of TID_U^0 and updates every session to protect the device from adversarial threats.

B. Device Registration Phase

The IoT enabled sensor node('sensor node') integrated with the medical equipment at the healthcare institution is first registered with the gateway. The registration process is very essential to discriminate between legitimate and malicious nodes. The gateway does not permit the doctor to communicate with the non-registered nodes to prevent cyber attacks. The whole process of sensor node registration is depicted in fig. 2 and discussed as follows.

Step 1: The gateway initially generates a challenge for sensor node registration C_{SN}^0 followed by another set of challenges C_{SN}^{SYN} $\{C_{SN}^1, C_{SN}^2, ..., C_{SN}^N\}$ to be used by gateway for future verification of the sensor node. The composed message $\{C_{SN}^0,\ C_{SN}^{SYN}\}$ is then sent to the sensor node through a secure channel.

Step 2: Once the sensor node receives the message $\{C_{SN}^0,\ C_{SN}^{SYN}\}$, it begins deriving the responses to the received challenges (C_{SN}^0, C_{SN}^{SYN}). The sensor node gives the challenge C_{SN}^0, C_{SN}^{SYN} as an input to the PUF wherein PUF generates the response $R_{SN}^0 = P_{SN}(C_{SN}^0)$, $R_{SN}^{SYN} = P_{SN}(C_{SN}^{SYN})$ to the challenges. The sensor node computes $\delta = \mathrm{h}(C_{SN}^0 \parallel C_{SN}^{SYN})$ to empower the gateway in verifying the association between response and challenges. Finally, the sensor node prepares a message $\{SN_{IEI}, R_{SN}^0, R_{SN}^{SYN}, \delta\}$ and delivers it to the gateway through a secure channel.

Step 3: Upon receiving the message, gateway at first computes $\eta = \mathrm{h}(C_{SN}^0 \parallel C_{SN}^{SYN})$ followed by a comparison of $\eta \equiv^? \delta$ to verify the relationship between responses and

challenges. Subsequently, the gateway derives the temporary identity of the sensor node TID^0_{SN} $\{= F(C^0_{SN}, SN_{IEI})\}$. After the computations and retrieval, the gateway stores the SN_{IEI}, C^0_{SN}, R^0_{SN}, C^{SYN}_{SN}, R^{SYN}_{SN}, and TID^0_{SN} for future interaction with the sensor node.

Step 4: Likewise, the sensor node computes and stores the temporary identity TID^0_{SN} $\{= F(C^0_{SN}, SN_{IEI})\}$. The use of temporary identity enables the sensor node to accomplish anonymity and untraceability while exchanging messages over the public channel.

Remark 2: Gateway records the real identity of the sensor node SN_{IEI} during registration for future correspondence purposes. Being resource-constrained and deployed in a hostile environment, the protocol does not store any secret credentials in the sensor node. Therefore, the attacker would not get any information despite physically capturing the sensor node. Besides, the sensor node only executes lightweight cryptography operations (hash and PUF) to prolong their active lifetime. It is noteworthy that TID^0_{SN} is a function of C^0_{SN}, and SN_{IEI}. The use of new challenge C^N_U for every session assures distinct short-term identity of sensor node TID^N_{SN} for every session, therefore accomplishing untraceability. Moreover, TID^0_{SN} does not disclose the real identity of the sensor node during message exchanges, thus preserving sensor node identity anonymity.

C. Mutual Authentication and Secret Key Establishment Phase

This phase demonstrates how the proposed protocol ensures the security and privacy of real-time communication between a doctor and a sensor node. The doctor and the sensor node prove their legitimacy to gateway succeeded by secret key establishment to prevent the unauthorized access of information to illegitimate nodes. The total process is illustrated in fig. 3 and explained as follows:

Step 1: Initially the doctor has to prove its identity to the device. At this stage, the doctor has to input the same password PW_U that was kept by him/her during the registration process. After the doctor enters the password PW_U, the doctor's device calculates $\lambda^* = h(PW_U \| TID^0_U)$ and verify the user authenticity, $\lambda^* \equiv^? \lambda$. In case of non-equivalency, the device does not permit the user to succeed in further communication, whereas successful verification indicates that the device is being used by the legitimate user and should be permitted to proceed further. Afterward, the device generates a nonce N^1_U, and compute $N^{1*}_U = N^1_U \oplus D^H_{ID}$ to protect the nonce privacy. The doctor's device creates a pseudo-identity $TID^{0*}_U = TID^0_U \oplus D^{MCC}_{LN}$ from the temporary identity TID^0_U to add a second layer of identity anonymity and untraceability protection. At last, the doctor's device prepares the message $\{N^{1*}_U, TID^{0*}_U\}$ and sends it to the gateway through a public channel.

Step 2: After receiving the identity details of user $\{N^{1*}_U, TID^{0*}_U\}$, the gateway extracts the real nonce, $N^1_U = N^{1*}_U \oplus D^H_{ID}$. Thereupon, the gateway verifies the freshness of the N^1_U. Subsequently, the gateway derives the temporary identity from pseudo-identity, $TID^0_U = TID^{0*}_U \oplus D^{MCC}_{LN}$ and matches with the database. Non-availability of TID^0_U in the database indicates a fraudulent attempt by an adversarial node. Once TID^0_U is located in the database, the gateway selects the corresponding challenge (C^0_U) and response (R^0_U). To meet the privacy requirements, the gateway encloses the real C^0_U, N^1_G within the G_1 $(= D^H_{ID} \oplus C^0_U)$ and G_2 $(= D^{MCC}_{LN} \oplus N^1_G)$. As the identity details of doctor D^H_{ID}, D^{MCC}_{LN} are not disclosed to anyone, therefore no one else can extract C^0_U and N^1_G other than gateway; the doctor identity details are already provided to the gateway during registration. Finally, the gateway computes $G_3 = h(C^0_U \| N^1_G \| R^0_U)$; G_3 can help the doctor's device to verify the authenticity of the gateway. The gateway sends the challenge C^0_U, nonce N^1_G, and authentication message enclosed in G_1, G_2, and G_3, respectively to the doctor's device through a public channel.

Step 3: Upon receiving the G_1, G_2, and G_3 from the gateway, the doctor's device begins extracting the challenge $C^{0*}_U = G_1 \oplus D^H_{ID}$, and nonce $N^{1*}_G = G_2 \oplus D^{MCC}_{LN}$. After examining the freshness of N^{1*}_G, doctor's device extracts response from PUF, $R^{0*}_U = P_D(C^{0*}_U)$. Subsequently, the doctor's device calculates $U_1 = h(C^{0*}_U \| N^{1*}_G \| R^{0*}_U)$ and compares with G_3 $\{= h(C^0_U \| N^1_G \| R^0_U)\}$. The equal values hereby proves the authenticity of the gateway as only gateway has the knowledge of R^0_U, whereas unequal values indicates a suspicious attempt by an unauthorized entity. The user device then prepares the pseudo-identity of the sensor node $SN^*_{IEI} = h(D^H_{ID} \| D^{MCC}_{LN} \| R^{0*}_U \| N^{1*}_G) \oplus SN_{IEI}$ with whom the communication link has to be established. Thereafter, the doctor device prepares device authentication value, $U_2 = h(C^{0*}_U \| N^{1*}_G \| R^{0*}_U \| TID^0_U)$. Likewise, other nonces used in the protocol, N^2_U is also shared secretly with gateway. The final message $\{U_2, U_3, SN^*_{IEI}\}$ is then sent to the gateway through a public channel.

Step 4: Firstly, the gateway derives the $N^2_U = U_3 \oplus D^{MCC}_{LN}$ and evaluates the freshness. If fresh, gateway begins the computation of $G_4 = h(C^0_U \| N^1_G \| R^0_U \| TID^0_U$ and examines the identicalness between U_2 and G_4. The message cannot be duplicated as it contains N^2_U, moreover, it cannot be prepared by anonymous entity since it contains TID^0_U which is only available with doctor's device and gateway. The mutual authentication between doctor and gateway gets accomplished if $U_2 \equiv G_4$, else fails. The gateway afterward extract the real identity of the sensor node for accomplishing mutual authentication, $SN_{IEI} = SN^*_{IEI} \oplus h(D^H_{ID} \| D^{MCC}_{LN} \| R^0_U \| N^1_G)$. After retrieving SN_{IEI}, the gateway then selects the corresponding C^0_{SN}, R^0_{SN} pair and generates the N^2_G. To ensure privacy, the challenge C^0_{SN} and the nonce N^2_G is secretly enclosed within G_5 and G_6, respectively. Gateway also prepares $G_7 = h(C^0_{SN} \| N^2_G \| R^0_{SN})$ to prove its identity to the sensor node. The gateway calculates the session key (SK) for the sensor node and enclose it secretly within SK^*_{SN} $(= h(R^0_{SN} \| TID^0_{SN}) \oplus SK)$. Correspondingly, gateway also encloses the temporary identity of doctor within μ $(= h(R^0_{SN} \| SK \| N^2_G) \oplus TID^0_U)$. Lastly, gateway selects a random new challenge C^1_{SN} from the set of challenges C^{SYN}_{SN} generated at the time of registration and computes $C^{1*}_{SN} = h(C^0_{SN} \| R^0_{SN}) \oplus C^1_{SN}$. The gateway compose a message consisting of G_5, G_6, G_7, SK^*_{SN}, μ, C^{1*}_{SN} and send it to the sensor node through a public channel.

Step 5: After receiving the message $\{G_5, G_6, G_7,$

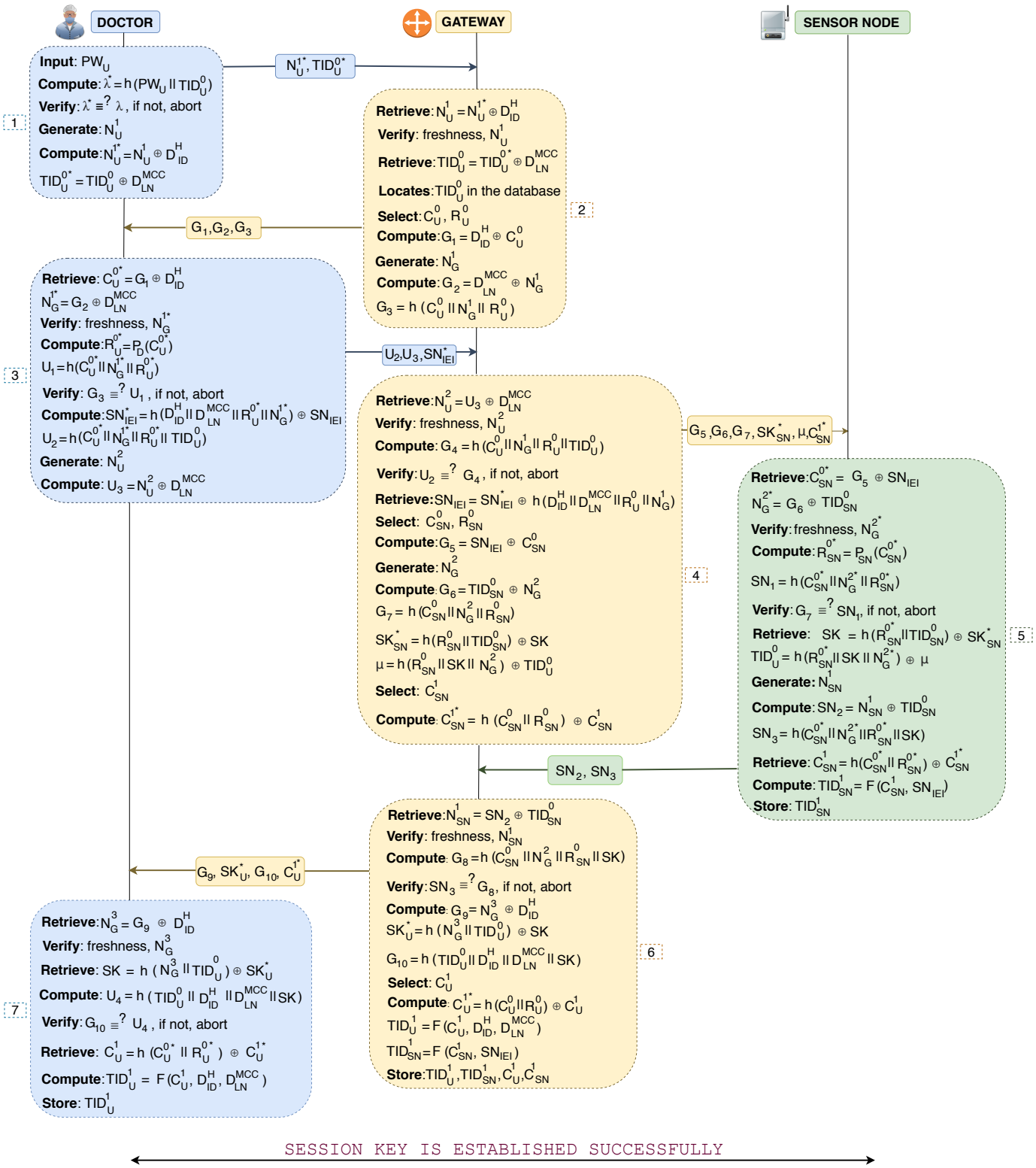

Figure 3. A lightweight and physically protected mutual authentication and secret key establishment protocol for real time data access in IoMT networks

SK_{SN}^*, μ, $C_{SN}^{1*}\}$ from gateway, the sensor node retrieves $C_{SN}^{0*} = G_5 \oplus SN_{IEI}$. Subsequently, the sensor node calculates $N_G^{2*} = G_6 \oplus TID_{SN}^0$ and examines the freshness of N_G^{2*}. Thereafter, the sensor node extracts the response from the PUF, $R_{SN}^{0*} = P_{SN}(C_{SN}^{0*})$. The sensor node derives $SN_1 = h(C_{SN}^{0*} \| N_G^{2*} \| R_{SN}^{0*})$ and compares $G_7 \equiv^? SN_1$; dissimilar values here indicate the failure of gateway authentication. If authentication is successful, the sensor node determines its session key by calculating $SK = h(R_{SN}^{0*} \| TID_{SN}^{0*}) \oplus SK_{SN}^*$. Likewise, sensor node extracts the temporary identity of doc-

tor, $TID_U^0 = h(R_{SN}^{0*} \| SK \| N_G^{2*}) \oplus \mu$. Afterward, the sensor node generates the nonce N_{SN}^1 and enclose it within SN_2 $(= N_{SN}^1 \oplus TID_{SN}^0)$ for its safe transmission. Besides, the sensor node computes $SN_3 = h(C_{SN}^{0*} \| N_G^{2*} \| R_{SN}^{0*} \| SK)$ to accomplish mutual authentication with gateway and also to assure the gateway of correct session key (SK) generation. Following the SN_3, the sensor node retrieves the new challenge $C_{SN}^1 = h(C_{SN}^{0*} \| R_{SN}^{0*}) \oplus C_{SN}^{1*}$ provided by the gateway to generate new temporary identity TID_{SN}^1 required for next future session. Once new challenge C_{SN}^1 is extracted, the sensor node calculates the $TID_{SN}^1 = F(C_{SN}^1, SN_{IEI})$ and stores the new temporary identity for future communication with gateway. At last, the sensor node composes a message $\{SN_2, SN_3\}$ and send it to gateway through a public channel.

Step 6: Upon receiving SN_2 and SN_3, the gateway starts the retrieval of N_{SN}^1 $(= SN_2 \oplus TID_{SN}^0)$ and verifies the freshness of N_{SN}^1. After the nonce verification, the gateway computes, $G_8 = h(C_{SN}^0 \| N_G^2 \| R_{SN}^0 \| SK)$ and compares, $SN_3 \equiv^? G_8$. Identical terms here indicate the successful mutual authentication along with an assurance of correct key generation by sensor node. Succeeding verification, the gateway calculates $G_9 = N_G^3 \oplus D_{ID}^H$. Thereafter, gateway encloses the session key of the user within $SK_U^* = h(N_G^3 \| TID_U^0) \oplus SK$. Subsequently, the gateway calculates $G_{10} = h(TID_U^0 \| D_{ID}^H \| D_{LN}^{MCC} \| SK)$ to let the user node verify the correct key generation. Post G_{10} computation, the gateway selects a random new challenge C_U^1 from the list of challenges C_U^{SYN} constructed during registration. Hence, the gateway encloses the new challenge within C_U^{1*} $(= h(C_U^0 \| R_U^0) \oplus C_U^1)$ to ensure confidentiality. Finally, the gateway calculates the new temporary identity of user, $TID_U^1 = F(C_U^1, D_{ID}^H, D_{LN}^{MCC})$ and sensor node, $TID_{SN}^1 = F(C_{SN}^1, SN_{IEI})$. The gateway then stores the $TID_U^1, TID_{SN}^1, C_U^1, C_{SN}^1$ into the database for future communication with the user and sensor node. The gateway compose a message $\{G_9, SK_U^*, G_{10}, C_U^{1*}\}$ and sends it to doctor's device through a public channel.

Step 7: The doctor's device receives the message $\{G_9, SK_U^*, G_{10}, C_U^{1*}\}$ from gateway. The doctor's device retrieves the nonce $N_G^3 = G_9 \oplus D_{ID}^H$ and examines its freshness. Thereafter, the doctor's device begins the extraction of session key, $SK = h(N_G^3 \| TID_U^0) \oplus SK_U^*$. Upon successful extraction, the device computes $U_4 = h(TID_U^0 \| D_{ID}^H \| D_{LN}^{MCC} \| SK)$ and verify the identicalness between U_4 and G_{10}. Identical U_4 and G_{10} results in accomplishment of mutual authentication between doctor's device and gateway whereas dissimilar U_4 and G_{10} results in connection termination. Successful verification also assures the doctor's device of correct key generation. Finally the device extracts the new challenge $C_U^1 = h(C_U^{0*} \| R_U^{0*}) \oplus C_U^{1*}$ and computes the new temporary identity, $TID_U^1 = F(C_U^1, D_{ID}^H, D_{LN}^{MCC})$. The doctor's device stores the new temporary identity TID_U^1 for future communication with the gateway.

Remark 3: The user authentication on the device ensures that only legitimate user is allowed to access the device for obtaining information from sensor nodes. Likewise user's device, the gateway also verifies the TID_U and TID_{SN} and permits only legitimate devices to access sensor nodes. In the proposed protocol, real identities of sensor node SN_{IEI} and doctor D_{ID}^H, D_{LN}^{MCC} are not used, instead temporary identities TID_U^0 and TID_{SN}^0 are used to accomplish identity anonymity and untraceability. Also, nonces are used in all messages to ensure freshness and protection against replay attacks. It is worth noting that neither the challenge $\{C_U^0, C_{SN}^0\}$ nor the nonce $\{N_U^1, N_U^2, N_G^1, N_G^2, N_G^3\}$ are disclosed on the public channel, therefore only authorized entities are entitled to retrieve this information. The verification message (U_2) for gateway is sent as a message digest, thereby not allowing the attacker to interpret and modify despite eavesdropping the G_1, G_3, and TID_U^{0*}. Moreover, the attacker cannot prepare U_2 because he does not have the knowledge of the C_U^{0*}, N_G^{1*}, R_U^{0*}, and TID_U^0. Similarly, attacker cannot interpret and alter G_7 despite eavesdropping G_5, G_6, and G_7 because he does not have the knowledge of C_{SN}^0, N_G^2, R_{SN}^0. The proposed protocol enables the gateway to exchange the session key and user identity securely with sensor node since it is enclosed using the values (R_{SN}^0, TID_{SN}^0, and N_G^2) not available to anyone except sensor node.

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/
              results/IoMT.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 1.37s
  visitedNodes: 88 nodes
  depth: 11 plies
```

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
PROTOCOL
  /home/span/span/testsuite/
              results/IoMT.if
GOAL
  As Specified
BACKEND
  CL-AtSe
STATISTICS
  Analysed   : 15 states
  Reachable  : 7 states
  Translation: 0.19 seconds
  Computation: 0.03 seconds
```

Figure 4. Results obtained from AVISPA while using OFMC and CL-AtSe backend.

V. Security Analysis

The security of the MASK protocol has been verified through formal and informal analysis. The inferences obtained from the analysis are discussed as follows.

A. Formal Analysis

To examine the strength of the MASK protocol, we have used the "Automated Validation of Internet Security Protocols and Applications (AVISPA)" tool [3], [22]. AVISPA tool is broadly adopted by the researchers for the strength investigation of mutual authentication and secret key establishment protocols. The MASK protocol has been scripted into the AVISPA understandable High-Level Protocol Specification Language (HLPSL). The MASK-HLPSL file is further transformed into Intermediate Format (IF) before being delivered to the backend of AVISPA. For the vulnerability assessment of MASK protocol, AVISPA is configured with Dolev-Yao (DY) adversary model [23]. The backend upon execution declares the scheme as either "safe/unsafe," or "inconclusive"; inconclusive mainly

results when the simulation of the protocol fails to happen either due to script error or backend incompatibility to a cryptography operation.

The MASK-HLPSL script contains the description of 3 basic roles, "user," "gateway," and "IoT node". Besides, the environment role defines the various sessions and intruder knowledge. As per the DY adversary model, the attacker has the capability to eavesdrop, alter, and replay the information. Another element in the environment role is "declaration of goals". The goals declared in the environment role of MASK protocol are data privacy, freshness, and authentication, etc. Upon execution, the OFMC backend produced the result as "safe" after visiting 88 nodes with a depth of 11 plies. Likewise, simulation of CL-AtSe took $0.19s$ to declare the protocol as "safe." The MASK-HLPSL-IF script has been fed to other 2 backends (SATMC and TA4SP) as well, but resulted inconclusive because these backends do not support bit-wise XOR operations. Therefore, it can be summarized from the results of OFMC and CL-AtSe backends that MASK protocol is safe from all prominent attacks including, replay and MITM.

B. Informal Analysis

The strength of the MASK protocol has been analyzed informally in this section for the following security properties and attacks.

Theorem 1. *Resistant to replay attacks.*

Proof. In a replay attack, the adversary captures the message and tries to resend it to the sensor node after a certain time to access unauthorized medical information of patients. Let us assume that the adversary has eavesdropped on the message $\{N_U^{1*}, TID_U^{0*}\}$ sent by the doctor to the gateway. The adversary later replays the message $\{N_U^{1*}, TID_U^{0*}\}$ to the gateway; the gateway discards the request since it contains old nonce, N_U^1. The MASK protocol equips every message with a random integer value called a nonce to accomplish message freshness that prevents replay attacks. Similarly, other exchanged messages $\{G_1, G_2, G_3\}$, $\{U_2, U_3, SN_{IEI}^*\}$, $\{G_5, G_6, G_7, SK_{SN}^*, \mu, C_{SN}^{1*}\}$, $\{SN_2, SN_3\}$, and $\{G_9, SK_U^*, G_{10}, C_U^{1*}\}$ contains the nonce N_G^1, N_U^2, N_G^2, N_{SN}^1, and N_G^3, respectively; thus ensuring the overall protection of the IoMT network from replay attacks. Moreover, the adversary cannot modify the nonce because it is enclosed secretly within other elements of the message. For example, the nonce sent by sensor node towards gateway is computed as $SN_2 = N_{SN}^1 \oplus TID_{SN}^0$; since the adversary does not know TID_{SN}^0, therefore attacker can't revive and modify the nonce, N_{SN}^1. Hence, MASK protocol accomplishes message freshness and prevents replay attacks. □

Theorem 2. *Secured from impersonation and MITM attacks.*

Proof. The adversary in impersonation attack tries to pose himself as the legitimate entity to get unauthorized access to legitimate information. The adversary needs to have some knowledge of the victim or its device credentials before launching this attack. In the MASK protocol, neither the true identities nor the secret information is shared over the public channel. Consider a scenario where attacker intercepted the message comprising $G_5, G_6, G_7, SK_{SN}^*, \mu$, and C_{SN}^{1*}. The message contains the identity of user, TID_U^0 and secret key, SK but enclosed secretly within μ $\{= h(R_{SN}^0 \| SK \| N_G^2) \oplus TID_U^0\}$ and SK_{SN}^* $\{= h(R_{SN}^0 \| TID_{SN}^0) \oplus SK\}$. For the attacker, it is computationally infeasible to retrieve TID_{SN}^0 and SK from μ and SK_{SN}^* due to collision-resistant property of hash functions [11]. Moreover, every device employing MASK protocol is integrated with unique PUF. The attacker neither has any knowledge of the responses generated by the PUF (R_U^0, R_{SN}^0) nor can predict [24]; therefore, the attacker cannot duplicate the identity of the user and sensor node. Hence, the proposed protocol is protected from impersonation attacks.

The attacker can also play MITM to compromise the communications. Let us suppose that adversary has captured the message $\{U_2, U_3, SN_{IEI}^*\}$. Subsequently, the adversary can try to modify the captured message to execute MITM. The adversary becomes unsuccessful to make any modifications in the MASK protocol messages because the information in the messages $U_2 = h(C_U^{0*} \| N_G^{1*} \| R_U^{0*} \| TID_U^0)$, $U_2 = N_U^2 \oplus D_{LN}^{MCC}$, and $SN_{IEI}^* = h(D_{ID}^H \| D_{LN}^{MCC} \| R_U^{0*} \| N_G^{1*}) \oplus SN_{IEI}$ are processed through one-way hash function and bit-wise XOR operation. The collision resistant property of hash functions [11] does not allow the attacker to either predict or revive the challenge (C_U^{0*}), response (R_U^{0*}), identity ($TID_U^0, D_{LN}^{MCC}, D_{ID}^H, SN_{IEI}$), and nonce ($N_G^{1*}, N_U^2$) values. Likewise, other messages of the MASK protocol are protected. Therefore, the MASK protocol is secured against MITM attacks. □

Theorem 3. *Preserves integrity and prevent modifications.*

Proof. The alterations by the attacker can ruin the real intent of the message and may produce unexpected outcomes. Consider an instance where adversary captured the message U_2 $\{= h(C_U^{0*} \| N_G^{1*} \| R_U^{0*} \| TID_U^0)\}$, U_3 $\{= N_U^2 \oplus D_{LN}^{MCC}\}$, SN_{IEI}^* $\{= h(D_{ID}^H \| D_{LN}^{MCC} \| R_U^{0*} \| N_G^{1*}) \oplus SN_{IEI}\}$. Despite capturing, the adversary cannot modify it since it is available as a message digest and can't be revived. The adversary has no information related to PUF (R_U^{0*}), nonce (N_U^2), and identity (D_{ID}^H), etc. required to modify the data. Moreover, the information computed through one way hash function complies with collision-resistant property [11] that does not allow the attacker to generate information from the message digest. The other messages of the MASK protocol are also composed similarly. Thus, the MASK protocol preserves the message integrity and further safeguard from modification attacks. □

Theorem 4. *Defence against DoS attacks.*

Proof. The adversary can try to disrupt the services of legitimate entities by flooding them with fraudulent requests. The consequences can be mild to severe including, exhaustion of battery reserves, temporary shutdown of network, denial of service to legitimate entities due to request overloading, etc. [25], [26]. Let us assume that the adversary eavesdropped on the message G_1 $(= D_{ID}^H \oplus C_U^0)$, G_2 $(= D_{LN}^{MCC} \oplus N_G^1)$, G_3 $(= h(C_U^0 \| N_G^1 \| R_U^0))$, and replayed it later to exhaust the battery reserves of the user device. Immediately after

reception, the user device verifies the nonce N_G^1, since the replayed message contains the old nonce, the user device terminates the connection instead of allocating new resources. Therefore, the MASK protocol can protect the network from DoS attacks to some extent. □

Theorem 5. *Promises session key security and protection from known key attacks.*

Proof. The adversary desires to acquire the session key to compromise all the future communications. In the MASK protocol, the gateway exchanges the secret key with sensor node and user device through the following messages, $\{G_5, G_6, G_7, SK_{SN}^*, \mu, C_{SN}^{1*}\}$, $\{G_9, SK_U^*, G_{10}, C_U^{1*}\}$, respectively. It is noteworthy that session key of sensor node is secretly enclosed within the SK_{SN}^* $(= h(R_{SN}^0 \| TID_{SN}^0) \oplus SK)$. Similarly, the session key for the user is also encapsulated secretly within $SK_U^* = h(N_G^3 \| TID_U^0) \oplus SK$. Therefore, the adversary who has no information about $R_{SN}^0, TID_{SN}^0, N_G^3$ cannot revive the session key (SK). Let us take a hypothetical case that adversary somehow has obtained the information after the expiry of the session, e.g., response (R_U^0, R_{SN}^0), temporary identity (TID_U^0, TID_{SN}^0), and session key (SK). These details would not help the adversary to request new keys from the gateway because the adversary does not know R_U^1, R_{SN}^1, TID_U^1, and TID_{SN}^1 which is required for authentication at gateway before establishment of new session keys. Therefore, knowing an old key does not enable the attacker to predict or compute new session keys. Therefore, the MASK protocol attains session key security and prevents known key attacks. □

Theorem 6. *Protection against physical attacks.*

Proof. Due to the tiny size, the sensor nodes deployed in a hostile environment is always subjected to physical attacks. Assume a situation where the attacker has physically captured the sensor node. The attacker's aim is either to prepare a clone or steal information from the chip of the sensor node. As aforementioned in the protocol description, the MASK protocol integrates the user device and sensor node with the PUF to protect them from physical tampering. Since the output of the PUF $\{R_U^0 = P_D(C_U^0),\ R_{SN}^0 = P_{SN}(C_{SN}^0)\}$ depends upon the intrinsic physical variations in the IC, therefore any attempt to tamper with the PUF would destroy the unique characteristics of the device and render the PUF useless [11]. Additionally, MASK protocol does not store any crucial information in the sensor node and user device. Consequently, the MASK protocol is safe against cloning and side-channel attacks. □

Theorem 7. *Exhibits data privacy.*

Proof. The adversaries captures the information and misuse it for their own benefit. It can be seen that MASK protocol does not exchange anything in the plain text considering the vulnerability of the wireless channel. Assume an instance where the adversary has captured the message $G_9, SK_U^*, G_{10}, C_U^{1*}$ to extract the useful information. The message component G_9 is composed of nonce (N_G^3) and doctor identity (D_{ID}^H); the attacker would not be able to retrieve the real nonce (N_G^3) as the doctor identity (D_{ID}^H) is never disclosed openly. Similarly $G_{10} = h(TID_U^0 \| D_{ID}^H \| D_{LN}^{MCC} \| SK)$, $SK_U^* = h(N_G^3 \| TID_U^0) \oplus SK$, and C_U^{1*} $(= h(C_U^0 \| R_U^0) \oplus C_U^1)$ are computed using secret values and one way hash function. Therefore, the information in the message remains confidential. Likewise, the remaining messages of the protocol preserves the data privacy. □

Theorem 8. *Attainment of user and sensor node identity anonymity and untraceability.*

Proof. The attackers use the identity information to conduct impersonation and MITM attacks whereas trace the origin of messages to perform physical and DoS attacks. Let us imagine that adversary has captured the message $\{N_U^{1*}, TID_U^{0*}\}$ to extract the identity details (D_{ID}^H, D_{LN}^{MCC}) of user. In spite of successful capturing, the adversary can not reveal the real identity (D_{ID}^H, D_{LN}^{MCC}) since it is never used during mutual authentication and key agreement phase. During the registration phase of MASK protocol, the gateway generates the temporary identity of the sensor node TID_{SN}^0 and user device TID_U^0 for future correspondence. Moreover, the temporary identity is further transformed into pseudo-identity during the mutual authentication for enhanced security. The user device sends the pseudo-identity of sensor node $SN_{IEI}^* = h(D_{ID}^H \| D_{LN}^{MCC} \| R_U^{0*} \| N_G^{1*}) \oplus SN_{IEI}$ and itself $TID_U^{0*} = TID_U^0 \oplus D_{LN}^{MCC}$ while communicating with the gateway. Hence the real identities of the sensor node and user device are never disclosed, thus keeping the communication anonymous. Moreover, the temporary identities $TID_U^0 = F(C_U^0, D_{ID}^H D_{LN}^{MCC})$, $TID_{SN}^0 = F(C_{SN}^0, SN_{IEI})$ changes every session due to change in input challenge $(C_U^1, C_U^2, .., C_U^N; C_{SN}^1, C_{SN}^2, .., C_{SN}^N)$, thus ensuring untraceability of user device and sensor node. □

VI. Performance and Comparative Analysis

The MASK protocol has been tested considering CM5000 TelosB mote with specifications as TI MSP430F1611 microcontroller, CC2420 RF chip, memory 1 MB, and a power source of $3V$ ($2 \times AA$ battery) [29]. The CM5000 TelosB mote is effective in operation but suffers due to limited resources such as power capacity, computing capability, storage space, and so on. Therefore, the security protocols should operate with minimum storage requirements. Besides, the security protocols should consume minimum energy while computation and communication to extend the lifetime of devices and hence networks. Considering the requirements, the MASK protocol has been evaluated in the resource-constrained environment to verify its suitability for the lightweight applications of IoMT. During the investigation, it is revealed that MASK protocol makes use of only 0.0008% of total memory space available in CM5000 TelosB mote whereas the schemes [10], [19], [21] requires 0.036%, 0.027%, and 0.015% of the storage space, respectively. It is evident from the investigation that the MASK protocol has very little storage space requirements in comparison to conventional protocols [10], [19], [21]. It is worth noting that the remaining schemes [18], [20], [22], [28] have not mentioned the storage cost requirements, hence the comparison is not possible.

The strength of the MASK protocol in a compromised environment has been examined and the results are presented

Table II
COMPARISON OF MASK PROTOCOL VS. CONVENTIONAL PROTOCOLS ON THE BASIS OF PROTECTION AGAINST ATTACKS AND ACCOMPLISHMENT OF SECURITY PROPERTIES

$\mathcal{S_G}$	[10]	[27]	[28]	[22]	[18]	[19]	[20]	[21]	$\mathcal{M}$
$\mathcal{P}_1$	✓	✓	✓	✓	✓	✓	✓	✓	✓
$\mathcal{P}_2$	✓	✓	✓	✓	✓	✓	✓	✓	✓
$\mathcal{P}_3$	✓	✓	✓	✓	✓	✓	✓	✓	✓
$\mathcal{P}_4$	✓	×	×	✓	×	×	×	✓	✓
$\mathcal{P}_5$	✓	✓	×	✓	×	✓	×	✓	✓
$\mathcal{P}_6$	✓	✓	✓	×	×	✓	✓	✓	✓
$\mathcal{P}_7$	×	✓	×	×	×	×	×	×	✓
$\mathcal{P}_8$	×	✓	×	×	×	×	×	×	✓
$\mathcal{P}_9$	✓	✓	✓	✓	✓	✓	✓	✓	✓
$\mathcal{P}_{10}$	✓	×	×	✓	×	×	×	×	✓
$\mathcal{P}_{11}$	✓	✓	✓	✓	×	×	✓	✓	✓
$\mathcal{P}_{12}$	✓	✓	✓	✓	✓	✓	✓	✓	✓
$\mathcal{P}_{13}$	✓	✓	✓	✓	✓	✓	×	✓	✓
$\mathcal{P}_{14}$	*	✓	✓	✓	✓	*	✓	✓	✓
$\mathcal{P}_{15}$	✓	✓	×	×	×	✓	×	×	✓
$\mathcal{P}_{16}$	*	✓	×	✓	✓	*	✓	✓	✓
$\mathcal{P}_{17}$	×	✓	×	×	×	×	×	×	✓

Acronyms: $\mathcal{S_G}$: Security goals, $\mathcal{M}$: MASK protocol, ✓: Secure against attack/preserves a security attribute, ×: Vulnerable/non accomplishment of security attribute, *: Not applicable, $\mathcal{P}_1$: Replay, $\mathcal{P}_2$: Impersonation, $\mathcal{P}_3$: Modification of messages, $\mathcal{P}_4$: DoS, $\mathcal{P}_5$: MITM, $\mathcal{P}_6$: Known key, $\mathcal{P}_7$: Cloning, $\mathcal{P}_8$: Side-channel, $\mathcal{P}_9$: Mutual authentication, $\mathcal{P}_{10}$: Data privacy, $\mathcal{P}_{11}$: Session key security, $\mathcal{P}_{12}$: Message integrity, $\mathcal{P}_{13}$: Message freshness, $\mathcal{P}_{14}$: User identity anonymity, $\mathcal{P}_{15}$: Sensor node identity anonymity, $\mathcal{P}_{16}$: User untraceability, $\mathcal{P}_{17}$: Sensor node untraceability

Table III
COMMUNICATION COST OF SENSOR NODE

Scheme	Cost (bits)	Cost (μJ)
[10]	720	519
[27]	1792	1405
[28]	912	703
[22]	864	653
[18]	960	748
[19]	1024	772
[20]	960	733
[21]	912	738
MASK	832	656

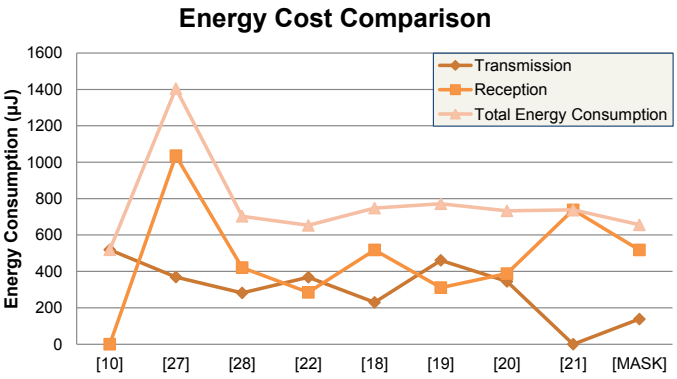

Figure 5. Energy Cost Comparison

in Table II. Table II lists the various security properties that are preserved by the protocol during its execution accompanied by the type of attacks that MASK protocol can withstand. The MASK protocol ensures data privacy, identity anonymity, untraceability, integrity, freshness, and session key security. Apart from the accomplishment of security properties, the MASK protocol is guarded against the most prominent attacks, to name a few, impersonation, modification, MITM, replay, cloning, and side-channel attacks. On the contrary, the conventional approaches [10], [18]–[22], [28] do not guarantee data privacy and untraceability. Moreover, these schemes [10], [18]–[22], [28] are also vulnerable to side-channel and cloning attacks. Also, the scheme proposed by Gope et al. [27] does not protect against DoS attacks. Since most of the schemes are neither protecting against physical attacks nor accomplishing untraceability and anonymity, therefore their deployment in a hostile environment can pose threats to the entire network. Hence, the MASK protocol is superior in contrast to other schemes [10], [18]–[22], [27], [28] in terms of preserving security properties and protection from attacks.

The protocol messages should be as minimum and as little as possible to avoid draining the battery reserves of mote. The TelosB mote [29] consumes 0.72 μJ and 0.81 μJ of energy [3], [9], [10] while transmitting and receiving, respectively. Table III provides the number of bits (T_X/R_X) and the quantity of energy consumption (μJ) by a sensor node during the mutual authentication and key establishment phase of the proposed protocol. Note that the registration phase is excluded since it executes only once during network set-up. It is apparent from the table III that MASK protocol uses the resources of the resource-deprived sensor node efficiently in comparison to the traditional approaches [18]–[21], [27], [28]. The schemes [10], [22] though consumes a little less energy than the proposed protocol, however, are incompetent to provide security to real-time IoMT applications. Moreover, these schemes [10], [22] neither prevent physical attacks nor ensure untraceability. Hence, using these approaches [10], [18]–[22], [27], [28] in IoMT networks can result in unexpected adverse consequences. Fig. 5 depicts the amount of energy spent by a sensor node for transmission and reception during the mutual authentication and key establishment phase. It is noticeable that the sensor node in the MASK protocol consumes the least energy during transmission and reasonable energy while reception. Thus, the energy-efficient characteristics of the MASK protocol makes it superior in comparison to other existing protocols [10], [18]–[22], [27], [28].

The computation cost spent by the user, gateway, and sensor node for implementing the MASK protocol have been analyzed and presented in Table IV. It can be observed that the MASK protocol is computation efficient because it employs only lightweight operations such as hash, PUF, and XOR rather than bulky cryptography operations like asymmetric and symmetric ciphers, scalar multiplications, and fuzzy extractors. Table V provides the comparison of computation cost spent by the MASK and other conventional protocols during the mutual authentication and key agreement phase. The scheme proposed by Gope et al. in [27] executes the PUF 5 times

Table IV
COMPUTATION COST OF MASK PROTOCOL

Phase	User Device	Gateway	Sensor Node	Total Cost
Registration	$2\ C_H + 2\ C_{PUF}$	$2\ C_H$	$C_H + 2\ C_{PUF}$	$5\ C_H + 4\ C_{PUF}$
Key Establishment	$7\ C_H + C_{PUF} + 9\ C_{XOR}$	$11\ C_H + 15\ C_{XOR}$	$5\ C_H + C_{PUF} + 6\ C_{XOR}$	$23\ C_H + 2\ C_{PUF} + 15\ C_{XOR}$
Total Cost	$9\ C_H + 3\ C_{PUF} + 9\ C_{XOR}$	$13\ C_H + 15\ C_{XOR}$	$6\ C_H + 3\ C_{PUF} + 6\ C_{XOR}$	$28\ C_H + 6\ C_{PUF} + 15\ C_{XOR}$

Acronyms: C - Computation, C_H - Computation of Hash, C_{PUF} - Computation of Physically Unclonable Function, C_{XOR} - Computation of Bit-wise XOR, *Integers* - defines the frequency of operation.

Table V
COMPUTATION COST COMPARISON OF MASK PROTOCOL VS. CONVENTIONAL PROTOCOLS

Scheme	Computation Cost
[10]	$C_{AE} + 3\ C_{AD} + 2\ C_H + 2\ C_M + 2\ C_{XOR}$
[27]	$22\ C_H + 5\ C_{PUF} + 16\ C_{XOR} + 3\ C_R + C_B$
[28]	$18\ C_H + 9\ C_{XOR} + 2\ C_R$
[22]	$37\ C_H + 16\ C_{XOR} + 2\ C_R + C_B$
[18]	$4\ C_{SE} + 4\ C_{SD} + 19\ C_H + 14\ C_{XOR} + 4\ C_R + C_B + 3\ C_{SM}$
[19]	$15\ C_H + 10\ C_{XOR} + 2\ C_R$
[20]	$18\ C_H + 9\ C_{XOR} + 3\ C_R + C_B + 6\ C_{SM}$
[21]	$25\ C_H + 20\ C_{XOR} + 3\ C_R + 9\ C_{MOD}$
MASK	$23\ C_H + 2\ C_{PUF} + 15\ C_{XOR} + 6\ C_R$

Acronyms: C - Computation, C_{AE} - Computation of asymmetric encryption, C_{AD} - Computation of asymmetric decryption, C_H - Computation of Hash, C_{SE} - Computation of symmetric encryption, C_{SD} - Computation of symmetric decryption, C_M - Computation of hash based MAC, C_R - Computation of random number, C_{PUF} - Computation of Physically Unclonable Function, C_B - Computation of bio-metric, C_{MOD} - Computation of modulus, C_{XOR} - Computation of Bit-wise XOR, C_{SM} - Computation of Scalar Multiplication, *Integers* - defines the frequency of operation.

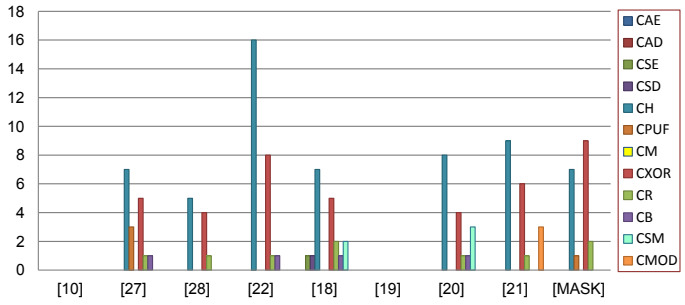

Figure 6. User Computation Cost of MASK protocol vs. Conventional Protocols

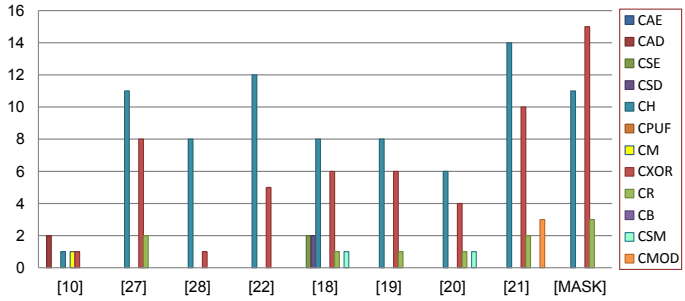

Figure 7. Gateway Computation Cost of MASK protocol vs. Conventional Protocols

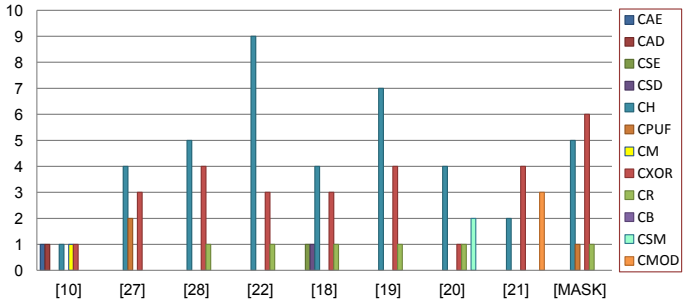

Figure 8. Sensor Node Computation Cost of MASK protocol vs. Conventional Protocols

whereas the MASK protocol only does it twice. Moreover, the scheme in [27] uses the fuzzy extractor for retrieving the bio-metrics whereas the MASK protocol does not use any bio-metrics. The approaches developed by Gaba et al. [10] and Li et al. [18] utilizes asymmetric and symmetric ciphers that overburdens the tiny processor of the sensor node. The other protocols designed by Das et al. [22], Li et al. [20], and Paliwal [21] are also computing expensive since they calculate bio-metrics, scalar multiplications, and modulus, respectively. Besides, the approaches in [22] and [21] also make excessive use of hash and XOR operations. The remaining protocols [19], [28] have reasonable computation complexity, however, it is achieved at the cost of compromised security. The schemes in [28] and [19] are vulnerable to DoS, cloning, and side-channel attacks and also fail to provide data privacy and sensor node untraceability. Additionally, these approaches [19], [28] are communication expensive as well. Fig. 6, Fig. 7, and Fig. 8 independently compares the cost of user, gateway, and sensor node, of MASK and other protocols, respectively. Based on the above analysis, it has become evident that the MASK protocol has attained all essential security properties with a very reasonable communication and computation cost.

The vertical bars in the Fig. 9 illustrates the total number of messages exchanged by the sensor node throughout the protocol whereas the diamond tag in the bar indicates the number of messages exchanged by the sensor node during mutual authentication and secret key establishment phase. It can be noticed that the resource constrained sensor node employing MASK protocol exchanges an equal number of messages as in other protocols, hence compute inexpensive. The sensor nodes in the schemes [28] and [22] exchange 3 and 4 messages, respectively, which is more than the MASK protocol. The number of message exchanges is also a performance metric

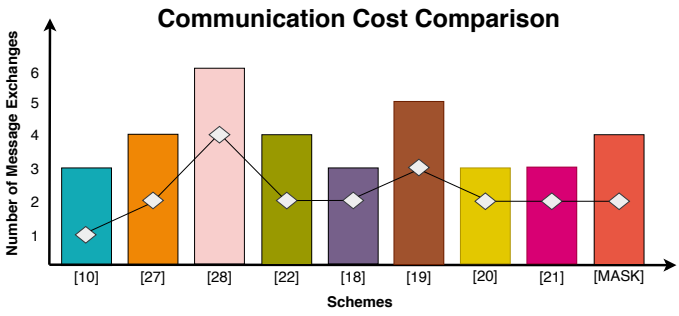

Figure 9. Communication Cost Comparison

to select a protocol for a particular application because more messages lead to more delay, overhead, and energy exhaustion. Conclusively, it can be summarized that MASK protocol is more robust and lightweight in comparison to state of the art protocols.

VII. Conclusions and Future Scope

This article have introduced a robust and lightweight security protocol to provide mutual authentication and secret key establishment between doctor and sensor node. The strength of the MASK protocol has been examined through formal and informal security analysis where it is declared safe from all the attacks mentioned in the DY adversary model. The performance analysis has proven the capability of MASK protocol to protect the sensor node from physical and other prominent attacks. The comparison reveals that MASK protocol outperforms the other conventional protocols in terms of preventing attacks, computation and communication efficiency, and so forth. In the future, the MASK protocol may be extended for those hostile environments where network devices like gateways are also subjected to physical attacks.

References

[1] M. S. Hossain and G. Muhammad, "Emotion-aware connected healthcare big data towards 5g," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2399–2406, 2017.

[2] M. S. Hossain, "Cloud-supported cyber–physical localization framework for patients monitoring," *IEEE Systems Journal*, vol. 11, no. 1, pp. 118–127, 2015.

[3] G. S. Gaba, G. Kumar, H. Monga, T.-H. Kim, and P. Kumar, "Robust and lightweight mutual authentication scheme in distributed smart environments," *IEEE Access*, vol. 8, pp. 69 722–69 733, 2020.

[4] D. Halperin, T. S. Heydt-Benjamin, K. Fu, T. Kohno, and W. H. Maisel, "Security and privacy for implantable medical devices," *IEEE pervasive computing*, vol. 7, no. 1, pp. 30–39, 2008.

[5] Y. Sun, F. P.-W. Lo, and B. Lo, "Security and privacy for the internet of medical things enabled healthcare systems: A survey," *IEEE Access*, vol. 7, pp. 183 339–183 355, 2019.

[6] A. Rahman, M. S. Hossain, N. A. Alrajeh, and F. Alsolami, "Adversarial examples–security threats to covid-19 deep learning systems in medical iot devices," *IEEE Internet of Things Journal*, 2020.

[7] G. Hatzivasilis, O. Soultatos, S. Ioannidis, C. Verikoukis, G. Demetriou, and C. Tsatsoulis, "Review of security and privacy for the internet of medical things (iomt)," in *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, 2019, pp. 457–464.

[8] "Medtech and the Internet of Medical Things," https://www2.deloitte.com/global/en/pages/life-sciences-and-healthcare/articles/medtech-internet-of-medical-things.html, 2020, online; accessed October 5, 2020.

[9] K. Choudhary, G. S. Gaba, I. Butun, and P. Kumar, "Make-it—a lightweight mutual authentication and key exchange protocol for industrial internet of things," *Sensors*, vol. 20, no. 18, p. 5166, 2020.

[10] G. S. Gaba, G. Kumar, H. Monga, T.-H. Kim, M. Liyanage, and P. Kumar, "Robust and lightweight key exchange (lke) protocol for industry 4.0," *IEEE Access*, vol. 8, pp. 132 808–132 824, 2020.

[11] P. Gope, J. Lee, and T. Q. Quek, "Lightweight and practical anonymous authentication protocol for rfid systems using physically unclonable functions," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2831–2843, 2018.

[12] M. Wolf and D. Serpanos, "Safety and security in cyber-physical systems and internet-of-things systems," *Proceedings of the IEEE*, vol. 106, no. 1, pp. 9–20, 2017.

[13] H.-L. Yeh, T.-H. Chen, P.-C. Liu, T.-H. Kim, and H.-W. Wei, "A secured authentication protocol for wireless sensor networks using elliptic curves cryptography," *Sensors*, vol. 11, no. 5, pp. 4767–4779, 2011.

[14] N. Khalil, M. R. Abid, D. Benhaddou, and M. Gerndt, "Wireless sensors networks for internet of things," in *2014 IEEE ninth international conference on Intelligent sensors, sensor networks and information processing (ISSNIP)*. IEEE, 2014, pp. 1–6.

[15] F. Wu, L. Xu, S. Kumari, and X. Li, "A privacy-preserving and provable user authentication scheme for wireless sensor networks based on internet of things security," *Journal of Ambient Intelligence and Humanized Computing*, vol. 8, no. 1, pp. 101–116, 2017.

[16] W.-B. Hsieh and J.-S. Leu, "A robust user authentication scheme using dynamic identity in wireless sensor networks," *Wireless personal communications*, vol. 77, no. 2, pp. 979–989, 2014.

[17] L. Zhou, X. Li, K.-H. Yeh, C. Su, and W. Chiu, "Lightweight iot-based authentication scheme in cloud computing circumstance," *Future Generation Computer Systems*, vol. 91, pp. 244–251, 2019.

[18] X. Li, J. Peng, J. Niu, F. Wu, J. Liao, and K.-K. R. Choo, "A robust and energy efficient authentication protocol for industrial internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1606–1615, 2018.

[19] A. Esfahani, G. Mantas, R. Matischek, F. B. Saghezchi, J. Rodriguez, A. Bicaku, S. Maksuti, M. G. Tauber, C. Schmittner, and J. Bastos, "A lightweight authentication mechanism for m2m communications in industrial iot environment," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 288–296, 2019.

[20] X. Li, J. Niu, M. Z. A. Bhuiyan, F. Wu, M. Karuppiah, and S. Kumari, "A robust ecc-based provable secure authentication protocol with privacy preserving for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3599–3609, 2018.

[21] S. Paliwal, "Hash-based conditional privacy preserving authentication and key exchange protocol suitable for industrial internet of things," *IEEE Access*, vol. 7, pp. 136 073–136 093, 2019.

[22] A. K. Das, M. Wazid, N. Kumar, A. V. Vasilakos, and J. J. Rodrigues, "Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial internet of things deployment," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4900–4913, 2018.

[23] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on information theory*, vol. 29, no. 2, pp. 198–208, 1983.

[24] C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: A tutorial," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1126–1141, 2014.

[25] Y. Zhang, C. Chen, and J. He, "Dos attack on networked control system: From the viewpoint on communication-control cost," in *2019 Chinese Automation Congress (CAC)*. IEEE, 2019, pp. 5695–5700.

[26] N. Enneya, A. Baayer, and M. ElKoutbi, "A dynamic timestamp discrepancy against replay attacks in manet," in *International Conference on Informatics Engineering and Information Science*. Springer, 2011, pp. 479–489.

[27] P. Gope, A. K. Das, N. Kumar, and Y. Cheng, "Lightweight and physically secure anonymous mutual authentication protocol for real-time data access in industrial wireless sensor networks," *IEEE transactions on industrial informatics*, vol. 15, no. 9, pp. 4957–4968, 2019.

[28] C.-C. Chang and H.-D. Le, "A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks," *IEEE Transactions on wireless communications*, vol. 15, no. 1, pp. 357–366, 2015.

[29] S. Fajarado, "CM5000 Datasheet," http://www.epssilon.cl/files/EPS5000.pdf, 2010, online; accessed February 15, 2020.