Device-centric Lightweight Authentication with Dynamic Addition and User Revocation for the Internet of Medical Things on B5G Networks

B D Deebak, Seong Oun Hwang, *Senior Member, IEEE,*

***Abstract*—Epidemic-prone diseases transforming the regular activities of people and health systems have become unprecedented events around the world. In the event of health emergencies like COVID-19, device-centric applications can expand communications capacities and global connectivity using technologies to meet the competitive requirements of healthcare IT. Most computing applications use integrated sensors, mobile networks, and services of the Internet of Things (IoT) to disseminate sensitive information via beyond fifth-generation (B5G) networks to edge computing systems. However, among the services of the Internet of Medical Things (IoMT), and dedicated information systems must address various challenging device-connectivity issues (in terms of integrity, security, and privacy) when any patient's device exchanges confidential information with a medical expert. Most of the existing techniques rely on open wireless communications, and there is the possibility of unauthorized access to interactions among sensors and computing servers, as well as potential risks such as ransomware and side-channel attacks. Thus, in this paper, a framework for device-centric lightweight authentication with dynamic addition and revocation (DC-LADAR) is proposed as an access-control mechanism to establish a shared session key with authorized servers and to ensure secure connectivity in dynamic IoMT environments. In particular, integrated devices utilizing the features of lightweight authentication handle large groups of users via proper revocation to protect their computing data in the cloud. The proposed DC-LA assigns a unary token that maintains secure key management between the implantable devices and a mobile edge computing (MEC) server to enhance network security. The proposed DC-LADAR minimizes the computation and communication overhead of IoMT-enabled networks to achieve better transmission efficiency.**

***Index Terms*—Device-centric Applications, Internet of Medical Things, Authentication, Access Control, Computation Overhead**

I. Introduction

Pandemics have prevailed around the globe affecting the sustainable development goals of the United Nations [1], surpassing the social, radical, and pedagogical boundaries of provinces. Internet of Things (IoT) technologies provide an intelligent framework that regulates the processes of healthcare systems, including remote monitoring, health screening, contact tracing, social distancing, patient tracking, and Internet-based services. Implantable devices use intelligent networks to increase the scope of patient treatments that tailor patient admission rates [2]. In the current pandemic, the implementation of IoT technologies plays an essential role in reducing the costs of healthcare systems and improving clinical treatment and diagnosis [3].

The IoT has digital technologies and machine components to process data transmissions. Each technology can define its own network without human intervention to offer real-time surveillance through wearable devices. Surveillance systems associate unique identification codes with implantable sensory devices that recognize the real-time requirements of patients, including sources of infectious disease, transmission routes, and regions of vulnerability. However, individuals demand a robust patient-care system that derives new theories to provide cost-effective solutions to connect short- or long-range communications protocols, including Bluetooth and M2M networks. Moreover, protocols have user-centric designs to describe the importance of innovative technologies that integrate operational devices such as software applications, intelligent hardware, and network connectivity to develop a device-centric system [4].

Such a system can be more responsive to support data collection and distribution that utilizes a fixed set of applications to compute and store workloads across a cloud environment. Additionally, the cloud uses a dedicated architecture to integrate different computing phases with a pharmacovigilance system to discover drugs and vaccines. As a result, any disease outbreak characterizes its intervals into pre-pandemic and pandemic to analyze the transmissibility of the infection around the globe in terms of active and recovered cases and deaths. In case of protecting the communication of transmission systems, the designed protocol must have the supportive features of efficient key update and revocation to mitigate computation and communication cost while conditionally anonymizing the identity of the computing devices. A monitoring system, based on syndromic surveillance, aims to

1) utilize IoT methodologies to design a concentric loop that includes thermal cameras and sensors, cloud servers, edge gateways, data analytics, and application services to find root causes;
2) execute standard operation procedures, such as root cause analysis, and generate recommended solutions like proactive measurement in association with the closed domain of the IoMT;
3) regulate patient monitoring to examine biometric measurements such as heart rate, glucose level, and blood pressure, classifying the nature of treatment (namely, hospitalization or home isolation); and
4) employ a device-centric healthcare system to ensure the integrity of the treatment plan, which uses a proper key authentication phase with secure revocation to protect the identity of M_D.

Due to controlling software and hardware over the Internet, the IoT ecosystem integrated with sensor networks provides a trusted environment to access safety-critical functionalities using built-in web servers. However, accessing the services based on various assessment methods demands device heterogeneity to monitor the connectivity with third-party servers. Therefore, most of the research has primarily focused on real-time data access between the medical sensors over the authorized networks to improve security and confidentiality. Almajali et al. [5] presented a comprehensive architecture of mobile edge computing (MEC) for IoT environments. Jia et al. [6] designed an identity-based authentication with user anonymity for MEC that applies ECC to examine the salient features of security and privacy. Jayashree and Santhosh [7]

This work was supported by the Brain Pool Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and Information Communication Technology (ICT) under Grant No. 2022H1D3A2A02081848.

B D Deebak and Seong Oun Hwang are with the Department of Computer Engineering, Gachon Univesity, Gachon University, Seongnam 13120, South Korea e-mail: (deebak@gachon.ac.kr and sohwang@gachon.ac.kr).

constructed a lightweight authentication with enhanced privacy (LA-EP) to provide secure communication among the mobility devices. The LA-EP applies elliptic-curve cryptography and bilinear pairing to manage key generation and to handle multi-party key change. Adversely, the above authentication protocols cannot protect the data transmission among the computing devices as they lack in mutual authentication process. As a result, we prefer a trusted gateway to establish secure communication among real-time users to gain reliable connectivity and guaranteed data access. In other words, a common session key is primarily connected with the session key agreement process to guarantee the proper key construction between the sensors and the end users.

A. Motivation

The IoMT platform in B5G networks uses the convergence platform to meet the desired constraints of quarantine testing. It can regulate real-time data capturing to track the treatment procedures of infected patients. To manage the clinical roles, a virtual management system is introduced. It has key entities such as hospitals, local bureaucracy, diagnostic centers, online consultations, logistics platforms, and quarantine places to utilize the interconnected components. As a result, an effective flow structure can be maintained to exchange confidential data. This flow structure may enable an issue assessment for medical experts, patients, social workers, and civilians to use IoT strategies that monitor the activities of suspicious cases. Since it maintains a large number of interconnected devices, it can easily maintain a massive amount of generated data using B5G networks, incorporating the features of THz communication to provide a flexible network architecture. In trusted environments, third-party servers prefer proper authentication and key exchange (AKE) to handle an encryption process with appropriate random numbers in healthcare applications.

Therefore, in the literature, real-time entities associated with smart sensing units have used public networks to securely process sensing data for end users. Unfortunately, applications of the industrial IoT make connected networks more vulnerable because unauthorized professionals exploit unique keys via key agreement protocol to protect the system against past key compromise and to share secret keys securely with legitimate participants in order to ensure message integrity during communication. In order to prevent leaking information, end users protect an intermediate access controller via gateways, resisting node capture and secret-leaking attacks. At the same time, sensing units use public key encryption to initiate communications via a dedicated gateway to protect the privacy of the sensitive data to enhance security at the edge of the connected networks [8]–[10]. Gupta et al. [8] designed lightweight secure authentication for privacy preserving communications to meet the design requirements of smart cities. Their protocol applies challenge-response pairing among users and gateways to protect privacy in the application systems.

However, it cannot overcome the issue of privacy information leakage as it localizes connectivity between system gateways to achieve anonymity and traceability as the service providers temporarily hold back the original message within their cache prior to stage authentication. He et al. [9] constructed lightweight authentication and key exchange with user anonymity (LAKE-UA) to verify the security of IoT applications. At the same time, LAKE-UA cannot solve the issues of resource-constrained applications, including computation and storage costs, to provide service decentralization in most IoT scenarios. Inopportunely, the LAKE-UA cannot offer a proper mutual authentication process among the entities to achieve confidentiality and data integrity. Chaudhary et al. [10] reconstructed modified lightweight authentication with proper key agreement (MLA-PKA) to meet the design features of the Internet of Drones. Unfortunately, MLA-PKA cannot authenticate transmitted messages flowing through network channels to preserve user identities [11]. Moreover, the above key agreement techniques do not consider the key revocation process to maintain high-level security with the integrated entities.

In particular, Azees et al. [12] designed an anonymous authentication with conditional privacy (AA-CP) for vehicular ad hoc networks (VANET), and they generated the identity revocation list among the entities (vehicle and roadside unit) to preserve their privacy. However, the AA-CP cannot evaluate its effectiveness in security as it relies on a trusted authority to store and access the entity's certificate [13]. On the contrary, the proposed DC-LADAR uses a single sign-on $\langle SSO \rangle$ gateway, authorizing the establishment of a crypto-period for the key to meet a few basic constraints on application devices or users (i.e., computation and communication costs).

B. Contribution

Data security and user privacy are becoming challenging issues in IoMT [14] as real-time data transmitted by smart devices is growing exponentially. The potential vulnerabilities in battery-operated devices can expose the sensitive data of the patient to malicious users to cause a privacy violation. There might be a chance to hijack the medical devices, thereby altering some medical data might be used to execute the physical threats to human life. To provide an optimized solution, device-centric authentication is preferred. It has the potential strengths to deal with various beneficial factors such as improved security, cost minimization, device traceability, etc. Thus, in this paper, we investigate the proposed DC-LADAR using a two-layer architecture to secure data transmission with dynamic addition and revocation. The first layer identifies device participation, namely the patient/doctor and the medical sensor, to verify the secure authentication mechanism. The second layer uses the trusted gateway to share received medical data. Our contributions are as follows.

1) We present device-centric lightweight authentication with dynamic addition and revocation to validate the identities of smart devices, i.e., making patient/doctor and medical sensors more realistic. Additionally, we apply intractability problems based on elliptic-curve cryptography to solve the problem related to the mutual authentication process and to protect the identities of the computing devices.
2) We utilize a secure SSO gateway that strategically processes the payload data to experience less computation and communication overhead. Moreover, it employs a two-layer architecture to secure the communication among the associated entities in order to protect the connected devices during transmission.
3) We apply random integers using a hash function to protect device access, and we use proper handshakes via secured gateways to validate the identity of trusted gateways. Importantly, the randomized parameters control their $range_{size}$ to maximize the value of the generated integers to acquire control of the computing system and data access in order to prevent unauthorized access.
4) We demonstrate assessment patterns to show that the proposed DC-LADAR achieves lower computation and communication costs, including execution time and the number of message rounds, to improve system efficiency.
5) Lastly, we deploy a practical testbed using onboard simulation to prove that the proposed DC-LADAR achieves greater efficiencies compared to other state-of-the-art approaches including mutual authentication protocol (MAuth) [15], seamless authentication (SAuth) [16], blockchain-enabled authentication (BAuth) [17], lightweight authentication (LAuth) [18], authentication and key exchange with anonymity (AKA-A) [9], secure and lightweight

authentication (SLA) [19], and user authentication and key establishment (UA-KE) [20].

C. Structure of the Paper

The rest of the sections are arranged as follows. Section II discusses related work on AKE protocols to address the key issues in the IoMT. Section III designs constructive network and threat models to examine the key requirements of IoMT applications to B5G networks. Section IV presents the device-centric lightweight authentication (DC-LA) framework to illustrate the computing process involved in session key agreement. Section V presents computation and simulation analyses of the proposed DC-LA compared with other existing schemes. Finally, Section VI summarizes this work with key findings and a practical solution.

II. Related Work

This section discusses mobile-sink and IoT-based authentication schemes for IoMT. He et al. [9] demonstrated the security weaknesses of the Farash et al. paper, including vulnerability to password guessing, stolen smartcards, impersonation attacks, and information leakage. Conversely, He et al. [9] extended key authentication protocol with anonymity to improve security efficiency. Srinivas et al. [28] proved that Amin and Biswas [29] security weaknesses make it vulnerable to sensor key eavesdropping, information leakage, and stolen smartcards. As a result, Srinivas et al. enhanced key authentication using multiple gateway access for the IoT environment. Jiang et al. [30] proved that Amin and Biswas [29] protocol is vulnerable to privileged insider attacks and information leakage. To address the security issues, Jiang et al. [31] constructed three-factor authentication and key agreement that exploit significant features of medical environments.

Challa et al. [32] developed a user authentication scheme using elliptic-curve cryptography (ECC) to achieve properties such as user anonymity and traceability. However, their scheme cannot minimize assessment costs, including computation and communication to meet the key objectives of the IoT. Chaudhary et al. [33] merged technological aspects of the tactile Internet for intelligent transportation systems that create energy trading systems to validate service requests. Jindal et al. [34] introduced an energy trading system to secure real-time transactions. It uses a consensus-based technique to improve system efficiencies. It uses a consensus-based technique to improve system efficiencies. Merabet et al. [15] studied core functionalities such as M2M and machine-to-cloud networks to support healthcare information services. Lin et al. [35] constructed an anonymous key authentication scheme with a conditional operation to construct a decentralized payment system.

Chaudhary et al. [25] designed a lightweight authentication mechanism for drone communication. Their mechanism uses low computation resources, including a one-way hash function and a bitwise XOR operation to minimize the cost inefficiencies in communication and storage. The mechanism applies an efficient key agreement protocol to prevent potential attacks (namely, password guessing, impersonation, etc.). Shen et al. [21] presented a lightweight multi-layer protocol to secure key generation between digital assistants and sensor nodes. This protocol uses non-pairing-based certificate-less authentication to achieve low computation costs with high-level security. Nyangaresi [22] developed an anonymous lightweight authentication using elliptic-curve cryptography to demonstrate trustworthiness in smart home privacy. This applied strategy derives a specific session token to prevent potential attacks like replay and man-in-the-middle. To ensure anonymity within intractable certificate management, most researchers prefer identity-based solutions [36], realizing the effectiveness of storage costs. For instance, the identity-based uses public key infrastructure (PKI) to generate vehicles' secret keys in order to provide efficient batch verification among the protocol entities. Still, the key revocation is unsolved in identity-based solutions because users cannot use an expired secret key to revoke their identities, as they are considered to be public information [37].

Abdussami et al. [23] constructed lightweight authentication using a physically unclonable function to generate a unique response over time to prevent device capture. Chatterjee et al. [24] designed a lightweight cryptographic algorithm to maintain data confidentiality and access control. Their algorithm uses addition, substitution, and XOR to solve computational complexity in hardware. Kumar and Chand [26] presented secure and lightweight authentication with public verifiability to improve security in IoT-based healthcare systems. In-depth analysis, however, proves that most of the key agreement mechanisms fail to prevent potential vulnerabilities such as privileged insider attacks, node capture, integrity protection, anonymity, perfect secrecy, and key confirmation. Lin et al. [38] utilized blockchain technology and key derivation mechanism to formulate a lightweight anonymous authentication (LAN-AUTH). The LAN-Auth reconstructs key derivation scheme to reduce adverse effects caused by on-chain retrieval operations. Yet, it is not flexible enough to use revocation mechanism as in any industrial environment, the smart device often joins or leaves anonymously to fulfill the proper demands of verification and traceability [39].

Wang et al. [40] developed a blockchain-assisted anonymous authentication with flexible revocation (BAAA-FR) to achieve faster data access among the smart devices. The BAAA-FR applies a two-level key derivation algorithm over a blockchain network to handle the issue of key generation center. Zhang et al. [41] presented a cache-based access control with privacy preserving (CBAC-PP) to prove an efficient authentication among vehicles and other associated entities in order to achieve better communication efficiency and data privacy. Wei et al. [42] proposed a lightweight secure authentication (LS-Auth) to secure transmission of emergency messages in VANET in order to protect the systems against side-channel attacks. Additionally, the LS-Auth designed a specific system secret key using Shamir's algorithm to transmit the emergency messages via insecure networks. However, CBAC-PP and LS-Auth cannot pre-store any pseudonym without key generation center to process any computing data in real time as the integrated devices have limited processing and storage capacities. Notably, they cannot authenticate the content cache list of any smart devices using revocation mechanism in order to continue the message transmission in any industrial IoT environment.

Conversely, few authentication schemes were unsuccessful in applying lightweight operators to reduce the computation and communication costs [9], [16]. As a consequence, we highly prefer lightweight authentication to improve the quality metrics of IoT environments as shown in Table I. Also, we employ a dedicated S_S to register M_D and E_U via secure single sign-on SSO gateway to ensure data traceability, achieving flexible revocation of computing device. To meet the desired constraints, this paper presents a device-centric lightweight authentication (DC-LA) that uses one-way hashing, XOR operations, and a fuzzy extractor to protect the activities of the real-time devices [17], [19], [20], [27].

III. Network and Threat Models

This section discusses a few significant models, including network and threat models, to signify the constructive concept of technological innovation and to manage a serious constraint on security efficiency in cloud-centric healthcare systems.

TABLE I: key challenges of state-of-the-art approaches with significant attributes.

Technique Utilized	Reference	Year	Applied Primitive	Key Properties						Simulation	Advantage	Limitation
				Message Integrity	Session Key Agreement	Forward Secrecy	Impersonation Attack	Known-Session Key Attack	Man-In-The-Middle Attack			
Lightweight Authentication	[21]	2018	Elliptic curve Cryptography and Intractability	No	No	No	No	Partially Claimed	No	Not Used	Provide mutual authentication among the sensing units.	Difficult in managing the certificate to protect device integrity.
	[22]	2022	Elliptic curve Cryptography	No	No	No	Yes	No	No	Not Used	Enable secure authentication to maintain device anonymity.	Fail to link authentication requests with computing devices.
	[23]	2022	Physically Unclonable Function	No	No	No	No	No	No	Not Used	Ensure security with shorter-keys beneficial to power-constrained devices.	Require extensive processing to offer high-level security.
	[24]	2022	Block Cipher Technique	No	No	No	No	No	No	Not Used	Use secure data handling to offer better processing efficiency.	Not adaptable to handle secure data transmission.
	[25]	2022	One-Way Hash Function and Bitwise XOR	No	No	No	No	No	No	Not Used	Introduce dynamic identities with proper message integrity.	Inadequate mutual authentication process to support forward secrecy.
	[26]	2021	Bilinear Pairing	No	No	No	No	No	No	Not Used	Protect data verifiability against chosen-ciphertext attack.	Insecure due to selective data sharing approach.
	[27]	2022	Physically Unclonable Function and One-Way Hash Function	No	No	No	No	No	No	Not Used	Improve the computational performance of healthcare systems.	Consume more processing cost while verifying the authentication parameters.
	The Proposed DC-LADAR	2025	Intractability Problems Based on Elliptic-Curve Cryptography	Yes	Yes	Yes	Yes	Yes	Yes	Used	Ensure security connectivity and authorized access in dynamic environments.	Fail to address the challenges of mutual authentication process.

A. Network Model

A device-centric healthcare system comprises five integral entities: trusted authority T_A, subscriber server S_S, application server A_S, emergency unit E_U, and medical device M_D. They manage the pandemic conditions occurrences that the infection rates worldwide, providing better quality in medical care. Fig. 1 shows the cyclic architecture of an IoMT platform [43]. In order to provide a sophisticated diagnostic procedure, the modeling system categorizes communications into two levels: proactive and reactive. The proactive level includes $E_U \longleftrightarrow T_A$, $S_S \longleftrightarrow T_A$, and $E_U \longleftrightarrow A_S$ while the reactive level has $M_D \longleftrightarrow E_U$ and $M_D \longleftrightarrow M_D$ to enhance the performance of point-of-care used in cutting-edge healthcare. Advances in healthcare systems streamline data sharing to enable better real-time monitoring between proactive and reactive communications via secure wireless channels, namely, transport layer security. The design flow is as follows:

1) T_A is a trusted entity with sufficient storage and computing resources to offer accessibility and integrity in database systems.
2) S_S utilizes sufficient storage and processing capacities to register M_D and E_U providing a wide range of devices to monitor healthcare information remotely.
3) A_S offers healthcare services at the diagnostic center to enable seamless communication via its privileged system.
4) E_U is a semi-trusted environment with adequate computing resources, such as power and storage to simplify the process of patient care and operations.
5) M_D is an untrusted entity with limited computing capacity to manage a set of authentication credentials and to access the network service with the remote user interface.

B. Threat Model

According to the adversarial model adopted in [18], the participating entities attempting to establish secure communication over an unsecured network are assumed to be unreliable or untrustworthy. The adopted model utilizes key attributes of the Dolev-Yao (DY) and Canetti-Krawczyk (CK) models to probe the system parameters in the key authentication phase. To assess the capabilities of adversary $\mathcal{A}_\mathcal{D}$, the following assumptions are constructed using DY and CK models.

1) Based on the DY model,
   - $\mathcal{A}_\mathcal{D}$ can fully control the connected edge networks over probabilistic polynomial time (PPT) to exploit vulnerabilities in transmitted messages (e.g., to eavesdrop, intercept, modify, delete, and inject). However, $\mathcal{A}_\mathcal{D}$ cannot infer any features of the key attributes to perform cryptanalysis; and
   - $\mathcal{A}_\mathcal{D}$ can use decryption or a signature process to read transmitted messages when the associated values are known. Please note that $\mathcal{A}_\mathcal{D}$ can also create a new message using an associated value to disrupt the authenticity of messages.
2) Based on the CK model,
   - $\mathcal{A}_\mathcal{D}$ can use a session-oriented security protocol to perform a few systematic operations over the communications channel (i.e., message-driven and unauthenticated links) to obtain secret values of the participating entities.
   - $\mathcal{A}_\mathcal{D}$ can deal with the source attributes of the session state to compromise the shared session key of honest parties. Please note that $\mathcal{A}_\mathcal{D}$ can also run Oracle queries to establish an interactive session with active participants.

IV. The Device-centric Lightweight Authentication Framework Mechanism

This section shows the significant use of the elliptic-curve cryptosystem to an anonymous user U_{sr} to secure a single sign-on (SSO) gateway. It is proven that the sensor node, S_N, executes only lightweight cryptosystem operations, namely, hash evaluation, MAC generation/verification, and symmetric encryption/decryption. Initially, a cryptographic building block is described to imply the use of the proposed DC-LADAR scheme with other existing schemes [9], [15]–[20]. Further, intractability problems based on elliptic-curve cryptography are applied in the proposed DC-LADAR to improve security efficiency and minimize the computation and communication costs.

A. Cryptographic Primitive

Elliptic-Curve Computational Diffie Hellman (ECC-DH): Assume that G_{EC} is an elliptic-curve group for the given prime integer, q_p. Usually, G_{EC} has a minor group that defines the group of points over the elliptic-curve finite field. Note that any elliptic-curve finite field can be applied to instantiate G_{EC} as suggested by NIST [44]. In [45], Khan et al. [45] described the elliptic-curve group for the given prime integer q_p. Assume p be a key generator of G_{EC} such that a problem of ECC-DH problem is computed $<p,q \in G_{EC}>$ when the given elements $<p,q \in G^2_{EC}$ where $<p,q> \in_R Z^*_q$. It is claimed that the assumption of ECC-DH holds for G_{EC} if the ECC-DH problem for G_{EC} is computationally infeasible to solve. It is also assumed that $\mathcal{ADV}^{\mathcal{SSK}}_{\diamond_\infty}(\mathcal{A}_\mathcal{D})$ can be negligible for the PPT algorithm of $\mathcal{A}_\mathcal{D}$ (i.e. the assumption holds for ECC-DH). $\mathcal{ADV}^{\mathcal{SSK}}_{\diamond_\infty}(\mathcal{A}_\mathcal{D})(t)$ denotes maximizing the value of $\mathcal{ADV}^{\mathcal{SSK}}_{\diamond_\infty}(\mathcal{A}_\mathcal{D})$ that executes overall $\mathcal{A}_\mathcal{D}$ algorithm with respect to time (t). Consider group parameters such as $\{G,g,andq\}$ to be the key parameters to define a public key PE_k, a secret key $\{SS_k,PE'_k,SS'_k\}$ and application server A_S via S_S.

Long-term secret key S_k can be determined with random string length k. Also,$H:\{0,1\}^* \rightarrow \{0,1\}^k$ is used to define a hash function that tries to prevent the target collision. A pseudo-random function

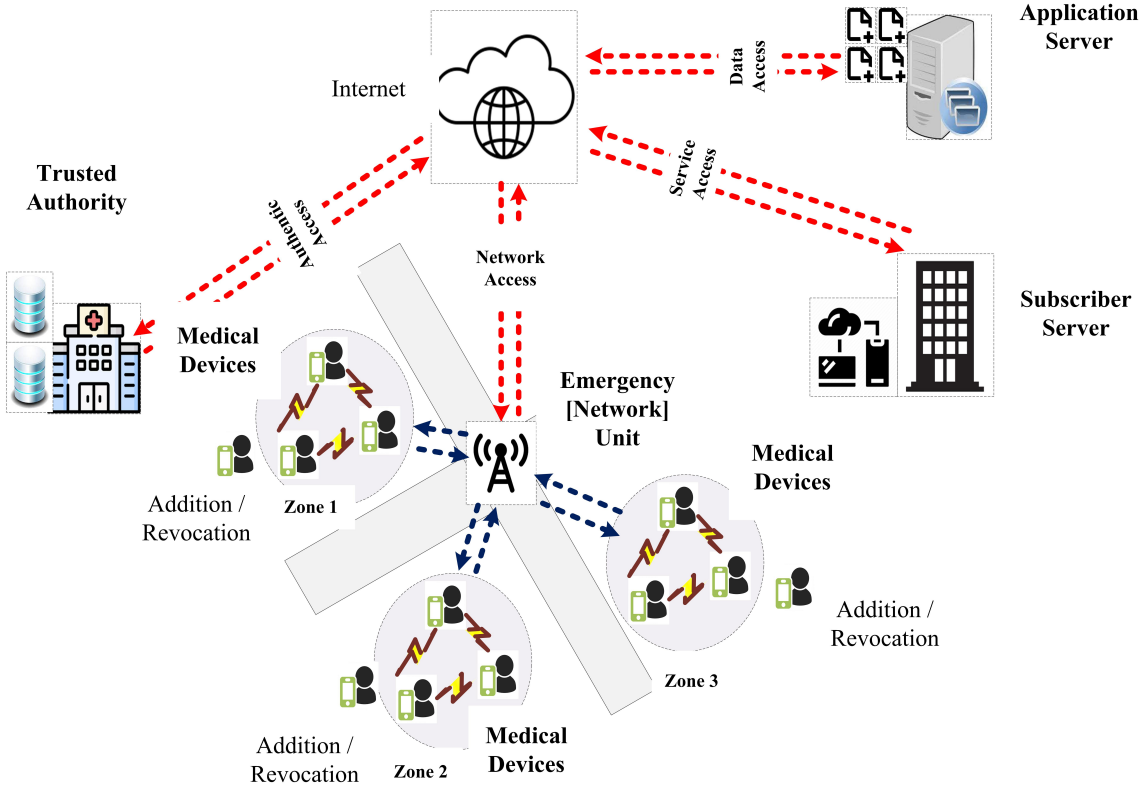


Fig. 1: Cyclic Architecture of an IoMT Platform.

key $PRF_{S_k}:\{0,1\}^k \rightarrow \{0,1\}^k$ is defined to use hash (conjugate) function $H^{'}:\{0,1\}^* \rightarrow \{0,1\}^k$ to conserve device identities. In the proposed DC-LADAR, S_k considers $H^{'}(S_k)$ as an input key.

Multimedia device M_d/Medical sensor M_{sen}, Application server A_S and $\langle SSO \rangle$ gateway constitute a real-time entity that does not employ any registration table for each registered communication entity. $\langle SSO \rangle$ gateway chooses two large prime integers p and q to compute $N=p.q$. $\langle SSO \rangle$ gateway uses the key pairs $\langle E_{NC}, D_{EC} \rangle$ such that $E_{NC}.D_{EC} \equiv 1 mod \phi(N)$, where $\phi(N)=(p-1).(q-1)$.

Moreover, $\langle SSO \rangle$ generates a key, k_g, over the finite-field Z_n^*, where n denotes a large prime number. Eventually, $\langle SSO \rangle$ protect the secret session-key SS_k that publishes the parameters, namely, $\langle E_{NC}, D_{EC}, g, n, N \rangle$. Note that the values of parameters p and q are removed after the system initialization to prevent security threats. M_d/M_{sen} and A_S store the public parameters (namely $\langle E_{NC}, g, g^a, n, N, H(.) \rangle$ published by $\langle SSO \rangle$ gateway, where superscript a indicates secret key-generation and $H(.)$ is a collision-resistant hashing function. Note. M_d/M_{sen} and A_S utilize their cryptographic token via SSO to support the workflow features of key storage and distribution to ensure a high-level physical protection over insecure networks.

System registration phase: A_S considers an $\langle SSO \rangle$ gateway to M_d/M_{sen} to achieve a higher level of security and user privacy in four steps. In the first step, M_d chooses a unary-identity I_d randomly to transmit it to A_S. In the second step, Upon receiving the unary-identity I_d, A_S computes $key = PRF_{S_k}(H(I_d)) \oplus H^{'}(S_{k_0})$ where S_{k_0} defines a secret-session key to validate the user's identity. In step 3, A_S issues a $\langle SSO \rangle$ gateway which is composed of $\{PE_k, PE_k^{'}, I_d, Key, p, g, q\}$.

In practice, the system parameters, excluding Key in the $\langle SSO \rangle$ gateway, are well defined in order to configure the $< SSO >$ gateway with M_d/M_{sen}. In the last step, M_d/M_{sen} initializes the device configuration set up with the users (e.g., patient/doctor) by using a common secret session key with A_S via $\langle SSO \rangle$ gateway to preserve data privacy.

Medical Sensor Registration Phase: M_{sen} selects an identity M_{sid}, a secret key $S_{MS} \in Z_q^*$ and compute $Puk_{M_{sens}} =< S_{MS}.S_k >$. Then, it computes $Puk_{M_{sens}-M_S} = \langle S_M S.key \rangle$ and compiles a transmission message as shown in equation 1.

$$T_M = (H(M_{sid} \| S_{MS} \| n \| Pu_{M_{sens}-M_S}) \oplus H(Puk_{M_{sens}-M_S}) \quad (1)$$

In this equation, n is a random nonce. Subsequently, M_{sen} sends parameters $\langle T_M, Puk_{M_{sens}} \rangle$ to A_S. Upon receiving this transmission, A_S computes $Puk_{M_{sens}-M_S} = Key.Puk_{M_{sens}}$ and $T_M \oplus H(Puk_{M_{sens}-M_S})$ to store transmission parameters M_{sid} and S_{MS} i.e. in its database. Note. A_S utilizes a protocol so-called transport layer security (TLS) to securely transmit the shared session key among the protocol entities.

After successful verification of message integrity, A_S computes $HMAC_{T_M} = HMAC(M_{sg} : \langle M_{sid}, Puk_{M_{sens}-M_S}, S_{MS} \rangle)$ and sends $HMAC_{T_M}$ to M_{sen}. Upon receiving the transmission, M_{sen} verifies $HMAC_{T_M}$.

System Login and Key-Authentication Phase:: This phase is carried via public network access. Assume that U_{sr} wishes to gain access to M_{sen} using U_{id}, Pwd_i and M_d. To achieve the security properties of mutual authentication and session key agreement, the execution phases are as follows.

Step 1: U_{sr} enters the key inputs U_{id} and Pwd_i into M_{d_i} (smart/mobile device). M_{d_i} computes $A_i = B_i \oplus H(U_{id} \| Pwd_i)$, $M_{id_i} = H(A_i \| U_{id})$ and $M_{Pwd_i} = H(M_{id_i}) \| Pwd_i)$.

Then, M_{d_i} computes $X_i^* = H(M_{id_i} \| M_{Pwd_i})$ to check whether X_i^* matches storage parameter $X_i = H(M_{id_i} \| M_{Pwd_i})$. If verification is successful, M_{d_i} verifies the corrective information of U_{sr}.

Step 2:M_{d_i} selects random integers such as SS_{k_i} and r_i^1 that generates a session key to update one-time pseudonym PD_i^{k+1}.M_{d_i} then computes Y_i^* as shown in Eq.(2):

$$\begin{aligned} Y_i^* = D_i \oplus H(M_{Pwd_i} \| X_i^*), \\ PD_i^{1*} = M_{id_i} \oplus H(M_{Pwd_i} \\ \| X_i^*), Z_i^{1*} = M_{id_i} oplus H(M_{Pwd_i} \\ \| X_i^* \| Y_i^*), MSG_1 = H(Z_i^{1*} \\ \| T_{S1}), MSG_2 = SS_{k_i}) \oplus H(Y_i^* \\ \| T_{S1}), MSG_3 = r_i^1 \oplus H(Z_i^{1*} \| Y_i^* \\ \| T_{S1}) and MSG_4 = H(MSG_1 \| MSG_2 \\ \| MSG_3 \| SS_{k_i} \| r_i^1 \| G_{SSO_i} \| T_{S1}) \end{aligned} \tag{2}$$

In Eq.(2), T_{S1} is the current timestamp, and G_{SSO_i} is the identity of $\langle$SSO $\rangle$gateway that locates U_{sr} availability. Lastly, M_d sends a login transmission $\langle PD_i^{1*}, MSG_1, MSG_2, MSG_3, MSG_4, T_{S1} \rangle$ to the $\langle SSO \rangle$ gateway.

Step 3: After receiving the login transmission, the $\langle SSO \rangle$ gateway verifies $|T_{S1} - T_S| < \triangle_{T_S}$. If verification succeeds, then the $\langle SSO \rangle$ gateway selects random integers SS_j and r_j^1. Random integers such as SS_{k_i} and r_i^1 are selected to generate a session key in order to update one-time pseudonym $PD_{SSO_i}^{k+1}$.

Using storage values $\langle u_j, v_j^1, PD_{SSO_i}^1 \rangle$, the $\langle SSO \rangle$ gateway then computes MSG_5 as shown in:

$$\begin{aligned} MSG_5 = H(v_j \| T_{S2} \oplus G_{SSO_i}), MSG_6 = SS_{k_i} \\ \oplus H(u_j \| T_{S2}), MSG_7 = r_i^1 \oplus H(v_j \| u_j \| T_{S1}) \\ and \\ MSG_8 = H(MSG_5 \| MSG_6 \| MSG_7 \| SS_{k_i} \| r_i^1 \\ \| PD_{SSO_i}^1 \| T_{S2}) \end{aligned} \tag{3}$$

where T_{S2} is the current timestamp of the $\langle SSO_i \rangle$ gateway. In order to achieve mutual authentication, the $\langle SSO \rangle$ gateway sends authenticate-message transmission $\langle PD_{SSO_i}^{1*}, MSG_5, MSG_6, MSG_7, MSG_8, T_{S2} \rangle$, including parameters $\langle PD_i^{1*}, MSG_1, MSG_2, MSG_3, MSG_4, T_S \rangle$ to A_S through the $\langle SSO \rangle$ gateway via public network access.

Step 4: Upon receiving the message from the $\langle SSO \rangle$ gateway, A_S verifies $|T_{S1} - T_S| < \triangle_{T_S}$. If the verification does not succeed, then A_S terminates any further execution process and sends a rejection notice to the $\langle SSO \rangle$ gateway. Otherwise, A_S extracts the G_{SSO_i} message parameters from the system database using $PD_{SSO_i}^1$ and computes $v_j^1 = H(PD_{SSO_i}^1 \| X_{SSO})$ and $G_{SSO_i}^* = MSG_5 \oplus H(v_j^{1*} \| T_{S2})$.

After successful computation of $G_{SSO_i}^*$, A_S tries to obtain $G_{SSO_i}^*$ that matches with G_{SSO_i} based on a one-time pseudonym. If $G_{SSO_i}^*$ does not match G_{SSO_i}, then A_S terminates DC-LADAR execution because the SSO_i gateway is not legitimate. In addition, A_S sends a rejection notice to U_{sr}/M_d or to the $\langle SSO \rangle$ gateway. Otherwise, A_S successfully authenticates SSO_i and U_{sr}/M_d. Then, A_S infers M_{d_i} from the system database using PD_i^1, and computes Z_i^{1*} and $M_{d_i}^*$ with Eq.4 to check whether the computation holds or not:

$$\begin{aligned} Z_i^{1*} = H(PD_i^1 \| X_d) M_{d_i}^* = MSG_1 \oplus \\ H(Z_i^{1*} \| T_{S1}) \end{aligned} \tag{4}$$

If verification holds, then M_d can establish its authenticity with A_S. Otherwise, A_S aborts session establishment and sends a notification message to M_d and the $\langle SSO_i \rangle$ gateway.

Step 5: Upon authentication of M_d and the $\langle SSO_i \rangle$ gateway, A_S derives random integers $\langle u_j^*, SS_{k_i}^*, r_i^{1*} \rangle$ to generate session key SS_k in order to update a one-time pseudonym. A_S computes u_j^*; the equation in Eq.(5) validate the correctness in receiving a message MSG_8.

$$\begin{aligned} u_j^* = H(G_{SSO_i} \| X_{SSO}), SS_{k_i}^{*s} = MSG_6 \oplus H(u_j^* \| T_{S2}), \\ r_i^{1*} = MSG_7 \oplus H(u_j^* \| v_j^1 \| T_{S2}) \quad and \\ MSG_8^* = H(MSG_5 \| MSG_6 \| MSG_7 \| SS_{k_i}^* \\ \| r_i^{1*} \| PD_i^1 \| T_{S2}) \end{aligned} \tag{5}$$

If validation is successful, A_S terminates session establishment. Otherwise, A_S computes Y_i^* and MSG_4^*, as seen in Eq.(6), to check the correctness in receiving message MSG_4:

$$\begin{aligned} Y_i^* = H(M_{d_i} \| X_d), SS_{k_i}^* = MSG_2 \oplus H(Y_i^* \| T_{S1}), \\ r_i^{1*} = MSG_3 \oplus H(Y_i^* \| Z_i^{1*} \| T_{S1}) \quad and \\ MSG_4^* = H(MSG_1 \| MSG_2 \| MSG_3 \| SS_{k_i}^* \| r_i^{1*} \\ \| G_{SSO_i} \| T_{S1}) \end{aligned} \tag{6}$$

If the last computation is not valid, A_S terminates session establishment and sends a rejection message to M_d and the $\langle SSO \rangle$ gateway. Otherwise, A_S executes the next step.

Step 6: A_S computes $N_{K_i} = H(SS_{k_i}^* \| M_{id_i})$ and $N_{K_j} = H(SS_{k_i}^* \| G_{SSO_i})$, which derive a session key between M_d and the $\langle SSO \rangle$ gateway. A_S then computes PD_i^2 and v_j^{2*} with Eq.(7).

$$\begin{aligned} PD_i^2 = H(PD_i^1 \| r_i^{1*}), PD_{SSO_i}^2 = H(PD_{SSO_i}^1 \| SS_{k_i}^*), \\ Z_i^2 = H(PD_i^2 \| X_d) \quad and \\ v_j^{2*} = H(PD_{SSO_i}^2 \| X_{SSO}) \end{aligned} \tag{7}$$

These computed values are used to derive one-time pseudonyms PD_i^2 and $PD_{SSO_i}^2$ that confirm derivatives Z_i^2 and v_j^{2*} in order to execute the next step between M_d and the $\langle SSO \rangle$ gateway, respectively.

Step 7: First, A_S computes MSG_9, MSG_{10}, MSG_{11}, MSG_{12}, MSG_{13} and MSG_{14} with the Eg.(8).

$$\begin{aligned} MSG_9 = N_{K_j} \oplus H(G_{SSO_i} \| Y_i^* \| T_{S3}), \\ MSG_{10} = Z_i^2 \oplus H(Y_i^* \| Z_i^{1*} \| T_{S3}), \\ MSG_{11} = H(MSG_9 \| MSG_{10} \| N_{K_i} \| N_{K_j} \\ \| PD_i^2 \| T_{S1} \| T_{S2} \| T_{S3} \| Y_i^*), \\ MSG_{12} = N_{K_i} \oplus H(PD_i^1 \| u_j^* \| r_i^{1*} \| T_{S3}), \\ MSG_{13} = v_j^2 \oplus H(u_j^* \| v_j^{1*} \| T_{S3}) \\ MSG_{14} = H(MSG_{11} \| MSG_{12} \| MSG_{13} \| N_{K_i} \\ \| N_{K_j} \| PD_{SSO_i}^2 \| T_{S1} \| T_{S2} \\ \| T_{S3} \| u_j^*) \end{aligned} \tag{8}$$

Then, A_S sends to the $\langle SSO \rangle$ gateway (via public network access) an authentication response that comprises $\{MSG_9, MSG_{10}, MSG_{11}, MSG_{12}, MSG_{13}, MSG_{14}, T_{S1}, T_{S2}, T_{S3}\}$. Key parameters such as PD_i^1 into PD_i^2 for M_d and $PD_{SSO_i}^1$ into $PD_{SSO_i}^2$ for SSO_i are updated in the storage database.

Step 8: Upon the receipt of an authentication response, SSO_i initially verifies whether $|T_{S3} - T_S| < \triangle_{T_S}$ is valid or not. If validation is not successful, then SSO_i terminates session establishment and sends a rejection message to M_d and A_S. Otherwise, SSO_i derives different values from the equations in Eq.(9):

$$\begin{aligned} N_{K_i}^* = MSG_{12} \oplus H(PD_i^1 \| u_j \| T_{S3}), \\ N_{K_j} = (SS_{k_i} \| G_{SSO_i}), \\ PD_{SSO_i}^2 = H(PD_{SSO_i}^1 \| SS_{k_i}), \\ v_j^2 = MSG_{13} \oplus H(u_j \| r_i^1 \| T_{S3}) \; and \\ MSG_{14}^* = H(MSG_{11} \| MSG_{12} \| N_{K_i}^* \| N_{K_j} \\ \| PD_{SSO_i}^2 \| T_{S1} \| T_{S2} \| T_{S3} \| u_j) \end{aligned} \tag{9}$$

These values are then used to check whether computation value MSG_{14}^{*} equates with MSG_{14}. If the computation holds, then SSO_i accepts that A_S and M_d are authentic. Otherwise, SSO_i terminates session establishment and sends a rejection message to M_d and A_S.

Step 9: After successful authentication of both M_d and A_S, SSO_i establishes a shared session key, $SSK_{M_d-SSO} = H(N_{K_i}^{*} \parallel N_{K_j})$, to compute $MSG_{15} = H(PD_i^1 \parallel G_{SSO_i} \parallel SSK_{M_d-SSO}^{*} \parallel T_{S1} \parallel T_{S2} \parallel T_{S3} \parallel T_{S4})$ and then sends to M_d (via public network access) login response $\langle MSG_9, MSG_{10}, MSG_{11}, MSG_{15}, T_{S1}, T_{S2}, T_{S3}, T_{S4} \rangle$. Finally, SSO_i modifies storage memory $\langle PD_{SSO_i}^1, v_j^1 \rangle$ to $\langle PD_{SSO_i}^2, v_j^2 \rangle$.

Step 10: Upon receiving the login response from SSO_i, M_d determines whether $|T_{S4} - T_S| < \triangle_{T_S}$ holds or not. If verification is not successful, M_d dismisses session establishment and subsequently sends a rejection message to SSO_i. Otherwise, M_d computes $N_{K_j}^{*}$ with Eq.(10) to check whether computation value MSG_{11}^{*} matches MSG_{11} or not:

$$\begin{aligned} N_{K_j}^{*} &= MSG_9 \oplus H(PD_{SSO_i}^1 \parallel Y_i^{*} \parallel T_{S3}), \\ N_{K_i} &= (SS_{k_i} \parallel M_{d_i}), PD_i^2 = H(PD_i^1 \parallel r_i^1), \\ Z_i^{2*} &= MSG_{10} \oplus H(Y_i^{*} \parallel Z_i^{1*} \parallel T_{S3}) \ \ and \\ MSG_{11}^{*} &= H(MSG_9 \parallel MSG_{10} \parallel N_{K_j}^{*} \parallel N_{K_i} \parallel PD_i^2 \\ &\qquad \parallel T_{S1} \parallel T_{S2} \parallel T_{S3} \parallel Y_i^{*}) \end{aligned} \tag{10}$$

If authentication is unsuccessful, M_d dismisses a login request and sends a rejection notice to SSO_i. If authentication is successful, A_S is an authentic server. Subsequently, M_d computes shared session key $SSK_{M_d-SSO} = H(N_{K_j}^{*} \parallel N_{K_i})$ to compute $MSG_{15} = H(PD_i^1 \parallel G_{SSO_i} \parallel SSK_{M_d-SSO}^{*} \parallel T_{S1} \parallel T_{S2} \parallel T_{S3} \parallel T_{S4})$ and sends authenticate access to the SSO_i gateway after verifying that $MSG_{15}^{*} = MSG_{15}$.

If authentication fails, M_d terminates the session and sends a rejection notice to $\langle SSO_i \rangle$ gateway. Otherwise, M_d finds an authentic $\langle SSO_i \rangle$ gateway to store data $\langle PD_i^1, Z_i^{1*} \rangle$ in $\langle PD_i^2, Z_i^{2*} \rangle$. Lastly, M_d successfully completes the login and authentication phase to share a common session key SSK_{M_d-SSO} between M_d and the $\langle SSO \rangle$ gateway.

Dynamic Addition and Revocation Phase: To meet a few standard constraints on application devices or users (e.g., time and cost), the proposed DC-LADAR provides supporting features such as adding a new M_d/M_{sen} pair and revoking registration process, enabling $\langle SSO \rangle$ to authorize the establishment of a crypto-period for the key. The significant steps in this process are as follows.

Step 1: When any U_{sr} wishes to join this platform, he/she initiates the registration request via $\langle SSO \rangle$ to authenticate his/her legitimacy with A_S.

Step 1.1: On obtaining the request, A_S evaluates U_{sr} over $\langle SSO \rangle$ for exchanging information with M_d/M_{sec}.

Step 1.2: Upon successful evaluation, $\langle SSO \rangle$ computes $\{A_i, M_{id_i}\}$ and accordingly sends system parameters $\{A_i, M_{id_i}, M_{Pwd_i}\}$ to U_{sr} enroute to all that are registered M_d/M_{sen}.

Step 2: Similarly, when U_{sr} wishes to revoke services with $\langle SSO \rangle$, $\langle SSO \rangle$ finds the essential parameters of U_{sr} in its source database, i.e., $\{M_{id_i}, M_{Pwd_i}\}$.

Step 2.1: Accordingly, $\langle SSO \rangle$ verifies the authenticity of U_{sr} using X_i to generate one-time pseudonym $PD_{SSO_i}^{k+1}$.

Step 2.2: At the same time, $\langle SSO \rangle$ communicates this revocation request to M_d/M_{sen} in order to cast the established pair $\{M_{id_i}, M_{Pwd_i}\}$ of U_{sr}.

Step 2.3: Lastly, U_{sr} successfully relinquishes connectivity with $\langle SSO \rangle$ to recover a few secured assets. However, the secured assets cannot be accessed without a proper $PD_{SSO_i}^{k+1}$ in order to preserve key features such as anonymity and untraceability.

System Password Update Phase: If any user wishes to update the current password without A_S recording it, the user must follow four steps. In the first step, U_{sr} enters key inputs U_{id} and old Pwd_i into smart/mobile device M_{d_i} i.e. smart/mobile device to obtain device or sensor access. In *Step 2*, M_{d_i} computes $A_i^{*} = B_i \oplus H(U_{id} \parallel Pwd_i^o ld)$, $M_{id_i}^{*} = H(A_i \parallel U_{id})$ and $M_{Pwd_i}^{*} = H(M_{id_i} \parallel Pwd_i^o ld)$.

Then, M_{d_i} computes $X_i^{*} = H(M_{id_i} \parallel M_{Pwd_i})$ to check whether X_i^{*} matches storage parameter $X_i = H(M_{id_i} \parallel M_{Pwd_i})$. If verification is successful, M_{d_i} verifies the information of U_{sr} to demand a new password Pwd_i^{new}. Otherwise, M_{d_i} terminates the password key execution phase. In next step, using a new password Pwd_i^{new}, M_{d_i} computes Y_i^{*} as shown in Eq.(11).

$$\begin{aligned} Y_i^{*} &= D_i \oplus H(M_{Pwd_i^{*}} \parallel X_i^{*}), PD_i^{1*} = M_{id_i} \\ &\quad \oplus H(M_{Pwd_i^{*}} \parallel X_i^{*}), Z_i^{K*} = M_{id_i} \oplus \\ &\qquad H(M_{Pwd_i^{*}} \parallel X_i^{*} \parallel Y_i^{*}), \end{aligned} \tag{11}$$

In Eq.(11), K is a key index of the next authentication integer. In the last step, M_{d_i} replaces parameters $\langle X_i^{*}, PD_i^{1*}, M_{id_i}, D_i, Y_i^{*} \rangle$ with $X_i^{*} = H(M_{id_i} \parallel M_{Pwd_i^{*}})$, $D_i^{'} = Y_i^{*} \oplus H(M_{Pwd_i^{*}} \parallel X_i^{*} \parallel Y_i^{*})$ and $M_{id_i} = PD_i^{K*} \oplus H(M_{Pwd_i^{*}} \parallel Y_i^{*}) = Z_i^{K*} \oplus H(MPwd_i^{*} \parallel X_i^{*} \parallel Y_i^{*})$ in the processed device in order to facilitate user access.

V. Security Assessment and Analysis

This section shows security proof based on the DY and CK models, adopting the capabilities of $\mathcal{A}_{\mathcal{D}}$ to assess the key attributes of device authentication and key agreement process. As constructed on the key agreement phase, we intend to devise a contest among $\mathcal{A}_{\mathcal{D}}$ and $\mathcal{C}_{\mathcal{H}}$ to prove the security efficiency of the proposed DC-LADAR.

A. Formal Model

To assess the strategies involved in the key agreement process, we adapt the constructive strategy applied in [46] on the Oracle settings to witness whether $\mathcal{A}_{\mathcal{D}}$ can produce a string related to $x^{'}$ over x or not. To make a proper relation with $\rho_{\mathcal{E},\Pi}^{\mathcal{R}}$, we must fulfill the key constraints of $\rho_{\mathcal{E},\Pi}^{\mathcal{R}}(x,x) = \rho_{\mathcal{E},\Pi}^{\mathcal{R}}(x,0^i) = 0$ for each $x \in \{0,1\}^{*}$, where $i \in N$, $R \in 2^{\infty}$, and $E, \Pi \in \{0,1\}^{*}$. To make the constraints more relevant to the gaming approach, we imitate computational modeling like the Turing Machine. It is abstracting its behavior to produce non-deterministic results, considering the capabilities of the devices including $D1_i$ and $D2_j$ whether i and j represent the service sessions of $D1$ and $D2$. Let us assume that $D1_i$ and $D2_j$ are associative partners while they have the unique session identifier to perform different roles agreeing upon a similar session key. To relate this assumption with practical findings, U_{sr} is assumed to be either $D1_i$ or $D2_j$. As stated by the threat model in Section III[B], $\mathcal{A}_{\mathcal{D}}$ is capable of issuing the following queries.

1) ***Send***(M_{sg}, U_{sr}) **query:** Using this query, we initiate a few active attacks such as replaying, deleting, modifying, and injecting over the transmitted messages to put data integrity and availability at risk. To assess its practicality, let's consider that $\mathcal{A}_{\mathcal{D}}$ is allowed to share the arbitrary message M_{sg} with U_{sr}. Importantly, based on the proposed DC-LADAR, this query generates its possible response via *Send (Start, U_{sr})* to U_{sr} to formulate the common random strings.
2) ***Execute***$(D1_i, D2_j)$ **query:** Applying this query, we simulate a passive attack i.e., eavesdropping which makes $\mathcal{A}_{\mathcal{D}}$ to responding to the transmitted messages executed by the honest protocol among $D1_i$ and $D2_i$.

3) ***Corrupt***$(U_{sr}, M_D/M_{sen})$ **query:** Operating this query, we administer two key entities namely U_{sr} and M_D/M_{sen} to act as initiator and target communicator. They handle the authentication process involving secret key SS_k to synchronize the shared session key between $D1_i$ and $D2_i$ to verify whether the source attributes of the entities are corrupted or not. Let ϵ_2 be a disparity in sharing the derived key with $D1_i$ and $D2_i$ to handle the following conditions.
   a) $\epsilon_2 \in \{-1,0,1\}$.
   b) In any event, U_{sr} and M_D/M_{sen} are required to synchronize their essential attributes to set up an authentic session. However, while the session terminates, it must hold a few key constraints:
      i) U_{sr} and M_D/M_{sen} have changed their derived key SS_k at least once;
      ii) The derived keys are in sync with U_{sr} and M_D/M_{sen} to execute the authentication process; and
      iii) In case [1] and [2] are successful, U_{sr} and M_D/M_{sen} find their shared session key via the SSO gateway to initiate the transmission request.
4) ***Reveal***$(U_{sr}, M_D/M_{sen})$ **query:** Operating this query, $\mathcal{A}_\mathcal{D}$ can learn the key attributes associated with the generation shared session key between U_{sr} and M_D/M_{sen}.
5) ***Test***$(U_{sr}, M_D/M_{sen})$ **query:** To make the query to be a more functional basis, we issue a fresh session key with U_{sr}, ensuring that the property of key freshness is guaranteed to provide explicit authentication to M_D/M_{sen}. Hither, we imply freshness to rely on a timestamp to check whether the queries such as *Corrupt*$(U_{sr}, M_D/M_{sen})$ and *Reveal*$(U_{sr}, M_D/M_{sen})$ are requested or not between the current session and its partner session to perform proper key verification, thereby detecting replay attack is practicable. On the other hand, the partner session associated with M_D/M_{sen} cannot be part of *Test* query to retrieve the key identifiers of U_{sr}. As a result, *Test*$(U_{sr}, M_D/M_{sen})$ query can make U_{sr} to flip a random bit b and accordingly, choose two separate classes 0 or 1. In case $b=1$, the session key can be returned; otherwise, it returns the random key of the equal sized.

We construct a game-based security model between the conditional challenger (checking the source attributes probabilistically) and an effective adversary (i.e., applying the algorithm using PPT) to analyze a set of game sequences (GA_i, where $i=0,1,2,...,N$). In each GA_i, we utilize a few source attributes of the authentication phase to set the winning event S_i and have the following assumptions to represent the capabilities of $\mathcal{A}_\mathcal{D}$:

1) Initiate a key impersonation attack to callously steal the sensitive data of the participating entities whereby $\mathcal{A}_\mathcal{D}$ can forge an honest session using its legal authenticator request. However, it cannot form the request with the partnering session to gain SSO gateway access fraudulently; and
2) Differentiate the applied request using its generated session key SSK to characterize the distinct feature of random integer, i.e., $b=b^{'}$, where b associates its session with *Test*$(U_{sr}, M_D/M_{sen})$ query and $b^{'}$ represents the guessing bits of $\mathcal{A}_\mathcal{D}$ to recover the part of message upon successful decryption.

Let us assume that $P_r[.]$ represents the occurrence of the probabilistic events, considering a secure session key exchange to show the objective difference with its aggregated results between $P_r[S_i]$ and $P_r[S_{i+1}]$ where $\forall i=0,1,2,...,N-1$ is insignificant and $P_r[S_N]$ is relatively closed to certain target probability in order to keep the probability of the encrypted messages more identical to each entity. Since the sequence of the game is determinate and limited in its form, $P_r[S_0]$ holding the real attack is considered to be negligible similar to *'target probability'* derived from triangular inequality as well. To achieve mutual benefits of PPT adversary, we apply the following lemma and theorem in the game.

Lemma 1: Key Difference Let us assume that X, Y, and F_L are the probabilistic events defined to guarantee key exchange with challenge-response in order to cross-examine the weakness of distributed session key SSK. In case X and Y desire to rely on the timestamp to find their deterministic random numbers identically, unless they obtain a failure event F_L, i.e., $X \wedge \neg F_L \Leftrightarrow Y \wedge \neg F_L$. Then, we have

$$\begin{aligned} X \leftarrow Y : r_i; \\ X \rightarrow Y : \{U_i, v_i, r_i, X^*\}; \text{ and} \\ |P_r[X] - P_r[Y]| \leq P_r[F_L]. \end{aligned} \tag{12}$$

Theorem 1: CCM-Based Model Using PRP Let us assume that F_L be a secure PRP based on $ECC-DH$ to represent its block size $K : \{0,1\}^{l_b} \rightarrow \{0,1\}^{l_b}$. Additionally, we assume a wide array of forgery attempts q_{FA}, block-cipher calls responding to encryption/decryption queries n_{EC}/n_{DC}, and the validity of attempted ciphertext c^* to analyze the merit of PPT adversary over the genuineness of CCM. As a result, we have

$$\mathrm{ADV}^{\mathrm{LAuth}}_{\mathrm{CCM}_{\mathrm{F_L}}}(A_D) \leq \epsilon_{pr} + \frac{q_{F_L}}{2^{l_t}} + \frac{(n_{EC}+n_{DC})^2}{2^{l_b}} \tag{13}$$

where l_t is the length of the tag and l_b is the size of the block. In every case of secure PRP, the random function involved in ϵ_{pr} is negligible with practical effort.

Theorem 2: SK Secure To make the key distribution more reliable, we hold the attributes of SSK associated with partner functions, addressing the behavior of $\mathcal{A}_\mathcal{D}$ to maintain the state of the security parameters $k \in N$. In most situations, $\mathcal{A}_\mathcal{D}$ utilizes the networking protocols to control the communication of the distributed environment to forge authentication requests. To realize the robustness of the proposed DC-LADAR, let us assume that $\mathcal{A}_\mathcal{D}$ be the merit of applying PPT adversary with *Hash* q_H, *Send* q_S, and *Execute* q_E queries to capture its capabilities over the selected session key. We also utilize lightweight authentication (LA) and key exchange (KE) properties to represent its execution states Π_1 and Π_2, respectively. The executed states made the queries return the sessional behavior of the entities to define the length of challenge-response pair l_{crp}, hash output h_o, and block-cipher of CCM b_c. As a result, the merit of using $\mathcal{A}_\mathcal{D}$ on Π_1 defines its session state as:

$$\mathcal{ADV}^{\mathcal{SSK}}_{\diamond_\infty}(\mathcal{A}_\mathcal{D}) \leq 3\epsilon_{ec} + \epsilon_{pr} + \frac{7q_S + 31(q_S+q_E)^2 + q_H^2}{2^{l_{crp}}} \tag{14}$$

Similarly, the merit of using $\mathcal{A}_\mathcal{D}$ on Π_2 defines its session state as:

$$\mathcal{ADV}^{\mathcal{SSK}}_{\diamond_\in}(\mathcal{A}_\mathcal{D}) \leq 2\epsilon_{ec} + \epsilon_{pr} + \frac{7q_H + 59(q_S+q_E)^2 + q_H^2}{2^{l_{crp}}} \tag{15}$$

Since ϵ_{ec} and ϵ_{pr} are insignificant, $\mathcal{ADV}^{\mathcal{SSK}}_{\diamond_\infty}$ and $\mathcal{ADV}^{\mathcal{SSK}}_{\diamond_\in}$ can be negligible till $2^{\frac{l_{crp}}{2}}$ calls of the executed queries. Considering the substitute values of k over l_{crp}, the proposed DC-LADAR can provide sufficient changes over the sessions to update the authenticated keys $(\tilde{P}_{xa}, \tilde{P}_{xb})$. Moreover, after the successful execution of the queries i.e., $\ll 2^{\frac{l_{crp}}{2}}$, we prove that the proposed DC-LADAR can make the attack probability to be negligible. Similarly, we can prefer the block length ≥ 256 bits to secure SK using the pseudo-random function (PRF) or pseudo-random permutation (PRP) to improve the security level of the systems, instead of using advanced encryption standard (AES).

Proof. To determine the robustness of the session key security in DC-LADAR, we perform a sequence of games systematically and imply some effective statistical or computational methods to distinguish a small change made by $\mathcal{A}_\mathcal{D}$. The sequence of games proceeds its transition from $Game_0$ to $Game_6$ and is as follows.

$Game_0$ In any transition, the target probability always argues that a distinguish algorithm D_A interpolates the source elements of the key agreement framework to define an event S_i in order to distinguish it with certain failure events F_L. Let us assume that the PPT $\mathcal{A}_{\mathcal{D}}$ initiates a vulnerable attack against the proposed DC-LADAR to assess the auxiliary inputs drawn over the random variables. As a result, the adversary merits in breaking the semantic security of the proposed DC-LADAR is defined as.

$$\mathcal{ADV}_{\diamond_\infty}^{\mathcal{SSK}} = |P_r[S_0] - \frac{1}{2}| \tag{16}$$

$Game_1$ Using this game, we observe the constructive behavior of *Hash* oracles under the random oracle model to distinguish the characteristic changes involved in public key encryption schemes. To assess its feasibility syntactically, we simulate the queries *Send*, *Execute*, *Corrupt*, *Reveal*, and *Test* as an authentic attacker to execute the key instances associated with *Hash* query. This modeling instance attempts to compile a transmission message $\{M_{sid}, PD_i^{k+1}, Y_i, X_i, A_i, T_S\}$ via $D1_i$ and $D2_j$ to obtain their public information and has the following routines.

1) l_{crp} assigns its occurrence with $l_b \xleftarrow{R} \{0, 1\}$ and $l_b^{'} \leftarrow \mathcal{A}_{\mathcal{D}}{}^{\mathcal{O}}(D1_i, D2_j)$ to analyze key freshness and impersonation attacks.
   - $D1_i$ initiates its session using $T_S \xleftarrow{R} \{0,1\}^{l_b}$ to verify whether $T_S \in N$ holds or not to return the source attributes of the authentication request; and
   - Accordingly, $D1_i$ verifies the identifier of M_{sid} to authenticate the session via SSO gateway to initialize the message transmission.
   - In a similar way, $D2_j$ assigns its own source attributes via SSO gateway to respond to the message request made by $D1_i$; and
   - After that, $D2_j$ tries to authenticate its session with $D1_i$ to explore the following queries: *Send*, *Execute*, *Corrupt*, *Reveal*, and *Test* in order to store their response with A_S.

To execute the modeling process of *Hash* query, we derive a truly random function i.e., RO_H which makes $\mathcal{A}_{\mathcal{D}}$ and l_{crp} to access the black-box of the random oracle. In order to store the query response of l_{crp}, a dedicated list $\mathcal{L}_{\mathcal{H}}$ is generated. This list maintains the query results to check whether $\mathcal{A}_{\mathcal{D}}$ obtains any authentic response to validate its message request with SSO gateway. In case of receiving the response, $\mathcal{A}_{\mathcal{D}}$ relies on SSO gateway to authorize the session key generation. Otherwise, a random value is returned as a new record in $\mathcal{L}_{\mathcal{H}}$. Since the transition of the query response is indistinguishable, the simulation always observes its computed hash output with $Game_0$ to choose the random value uniformly. Therefore, the cross-entropy smoothing assumption for the given *hash* is claimed as.

$$|P_r[S_1] - P_r[S_0]| \leq 3\epsilon_{ec} \tag{17}$$

$Game_2$ This game is similar to $Game_1$ except collision strategy. This strategy makes the honest responder choose its random nonce over two sessions whereby $\mathcal{A}_{\mathcal{D}}$ extracts the source attributes of the proposed DC-LADAR to initiate an honest session via SSO gateway. As SSO gateway relies on honest responders with an appropriate random nonce, we prepare q queries over a random function l_b to return the collision probability within $\frac{q^2}{2}.2^{-l_b}$ using birthday paradox bound. In order to validate the random nonce over two sessions (i.e., $D1_i$ and $D2_j$), we uniformly distribute a pair of nonce $(q_S + q_E)$ in the calling function. Additionally, we assume that $\mathcal{A}_{\mathcal{D}}$ chooses its pairing nonce q_S to initiate a legitimate session with M_D/M_{sen} delimited by $\frac{(q_S+q_E)^2}{2^{l_b+1}}$. Hence, the responder session chooses its nonce randomly to generate the pairing key at most $\frac{q_S}{2^{l_b}}$ via SSO gateway to validate the occurrence of collision probability. It is also noted that the session can be aborted when the pairing key is generated out of the *Hash* query. Importantly, $Game_2$ and $Game_1$ are similar except for the event of *abort* due to collision occurrence. According to *Lemma 1*, we have

$$|P_r[S_2] - P_r[S_1]| \leq \frac{(q_S+q_E)^2}{2^{l_b+1}} + \frac{q_S}{2^{l_b}} + \frac{q_H^2}{2^{l_b+1}} \tag{18}$$

$Game_3$ In this game, we assume that $\mathcal{A}_{\mathcal{D}}$ exclude a few discreet events such as session identifier, responder, and authenticator to analyze the robustness of key distribution protocol with SSO gateway. However, to formalize the secrecy of fresh keys, we let $\mathcal{A}_{\mathcal{D}}$ to tampering the source parameters of the transmitted messages i.e., MSG_9 to MSG_{15}. To validate the security weakness with honest devices, we explore the following messages i.e., MSG_{11}, MSG_{14}, and MSG_{15}. However, the leakage of any source parameters makes $\mathcal{A}_{\mathcal{D}}$ to infer the attributes $\{u_j, SS_k, r_i, N_{K_i}^*, N_{K_j}\}$ in order to authenticate a session with M_D/M_{sen}. While $\mathcal{A}_{\mathcal{D}}$ obtained the session key, we are supposed to set an alert flag $alt_{P_{1d1}}$ or $alt_{P_{1d2}}$. Otherwise, the session key generating by $\mathcal{A}_{\mathcal{D}}$ cannot create a session with M_D/M_{sen} and U_{sr} to rely on a constructive protocol Q_A/Q_B which is practically insecure. As a result, $\mathcal{A}_{\mathcal{D}}$ applies a few strategies over the access model $A_M : \{Send, Execute, Corrupt, Reveal, Test, Random_{Access}\}$ to perform the following analyses.

1) Assign $(Q_A, T_S) \leftarrow MSG_{14}$ to validate the computational value of MSG_{14}^* over T_S;
2) Initialize a session with $D1_i$ to find the values $\{N_{K_i}^*, N_{K_j}, PD_{SSO_i}^2, v_j^2\}$
3) Check the connectivity between M_d and A_S to verify whether MSG_{14} i genuine to perform,
   - if $corrupt_{D2_j} = True$ & $(guess_{r_i} \,\|\, guess_{alt_{P_{1d2}}} = True) \,\|\, corrupt_{D2_j} = False$ & $guess_{alt_{P_{1d1}}} = True$;
   - return $alt_{P_{1d1}} = True$; Abort
4) Apply a valid query using $\sum := (M_{sid}, Q_B)$ to verify whether $(v_j^2, \hat{T_S}) \leftarrow N_{K_i}^*$ is authentic or not.
5) If $T - S = \hat{T_S}$
   - Change $Status_{D1_i}[M_{sid}] := Accept$ to authenticate the session with SSO_i
   - Compute a common session key SSK_{M_d} to find an authentic SSO_i to store the computed data $\langle PD_i^1, Z_i^{1*} \rangle$
6) Similarly, $D2_j$ initializes its session with $D1_i$ to perform the following processes,
   - Use a proper forgery query to Validate the session identifier with $corrupt_{D1_j} = True$ & $(guess_{r_i} \,\|\, guess_{alt_{P_{1d1}}} = True) \,\|\, corrupt_{D1_j} = False$ & $guess_{alt_{P_{1d2}}} = True$;
   - Set an alert $alt_{P_{1d2}} = True$; Abort
   - Otherwise, $D1_j$ and $D2_j$ authenticate the session with SSO_i to generate the message output.

When generating the outputs with $D1_j$ and $D2_j$, we utilize $alt_{P_{1d1}}$ and $alt_{P_{1d2}}$ to set off an alert with SSO_i. Otherwise, SSO_i terminates its session. To validate its genuineness, $\mathcal{A}_{\mathcal{D}}$ obtains $P_{1d1}(P_{1d1})$, whereby $\hat{P_{1d1}} = P_{1d1} = \phi_1 \oplus P_{1d2}$, $\hat{P_{1d2}} = P_{1d2} = \phi_2 \oplus P_{1d2}$, and $P_{1d2}(P_{1d2})$ are validated in the following ways.

1) $\mathcal{A}_{\mathcal{D}}$ uses *Corrupt* $(D2_j)$ to read the value ϕ_1 and accordingly, obtains P_{1d2} to verify the property of key freshness. Additionally, $\mathcal{A}_{\mathcal{D}}$ infers a random guessing bit r_i to obtain the value P_{1d1}. Otherwise, it directly guesses a bit value P_{1d1}, while the probability is not more than $\frac{q_{F_L}}{2^{l_t}}$.
2) On the other hand, $\mathcal{A}_{\mathcal{D}}$ cannot directly guess a random bit value $\hat{P_{1d1}}$, while it fails its strategy over *Corrupt* $(D2_j)$. In this case, the probability of guessing bit is not higher than $\frac{q_S}{2^{l_b}}$.

3) Similarly, $\mathcal{A}_{\mathcal{D}}$ uses *Corrupt* $(D1_j)$ to read the value ϕ_2 and accordingly, obtains P_{1d1} to verify the property of key freshness. Additionally, $\mathcal{A}_{\mathcal{D}}$ infers a random guessing bit r_i to obtain the value $P_{1d2}(H(H(r_i,M_{sid})))$. Otherwise, it directly guesses a bit value $P_{1d2}=H(P_{1d1})$, while the probability is not more than $\frac{3q_S}{2^{l_b}}$.
4) $\mathcal{A}_{\mathcal{D}}$ cannot directly guess a random bit value $\hat{P_{1d2}}$, while it fails its strategy over *Corrupt* $(D1_j)$. In this case, the probability of guessing bit i.e., r_i or $\hat{P_{1d2}}$ uses $P_{1d2}(H(H(r_i,M_{sid})))$ or $P_{1d2}=H(P_{1d1})$ to guess the probability which is not higher than $\frac{3q_S}{2^{l_b}}$.

Since the guessing bit has its occurrence on P_{1d1} or r_i to repeat the process over $\frac{3q_S}{2^{l_b}}$, we have

$$|P_r[S_3]-P_r[S_2]|\leq\frac{6q_S}{2^{l_b}} \tag{19}$$

$Game_4$ In this game, We use a strategy of $Game_3$ in $Game_4$ to validate the execution of the authentication phase with proper session key agreement where the relying protocol Q_A validates its connectivity with $D1_i$ $D2_j$ over SSO_i. While deriving the session key SSK_i, we set off a flag $alt_{P_{1d1}}$ to execute the following processes,

1) Check whether the output message corresponds to the previous records or not;
2) Rely on Q_A protocol to verify the connectivity with $D1_i$ and $D2_j$; and
3) Examine the session reliability with SSO_i to validate the genuineness of SSK_i.

In this game, we consider the above strategies to abort any adverse event that uses a constructive protocol Q_A over $\mathcal{A}_{\mathcal{D}}$ to explore the forgery attempts q_{FA} using **Therorem 1**. According to this, when any $alt_{P_{1d1}}$ is triggered via SSO_i, $\mathcal{A}_{\mathcal{D}}$ constructs a query-based structure to submit the forgery message i.e., $\langle M_{sid},Q_A\rangle$. Since the constructed protocol does not handle this forgery message in prior, $\mathcal{A}_{\mathcal{D}}$ may win this game successfully between $D1_i$ and $D2_j$ to set its SSK_i. However, $\mathcal{A}_{\mathcal{D}}$ cannot form a legal transmission request as the proposed DC-LADAR realizes its protection over *CCM-Encryption* using AES to secure under *PRP*. To realize its security features, the encrypted query iterates a block ciphertext (r_i,T_S) over *six times*. Moreover, this iteration process makes $\mathcal{A}_{\mathcal{D}}$ to realize the encryption calls using *Send* and *Execute* queries to return forgery attempts and encrypteed queries at most q_S and (q_S+q_E). In case the size of the query tag l_{st} and cipher-block l_b are similar, according to Eq.14, we have

$$|P_r[S_4]-P_r[S_3]|\mathrm{ADV}^{\mathrm{LAuth}}_{\mathrm{CCM_{F_L}}}(\mathcal{A}_{\mathcal{D}})\leq\epsilon_{pr}+\frac{q_{F_L}}{2^{l_t}}+\frac{q_S+(14q_S+7q_E)}{2^{l_b}} \tag{20}$$

$Game_5$ In this game, we exclude a few discrete parameters $\{N^*_{K_i},N_{K_j},PD^2_{SSO_i},v^2_j\}$ to obtain the attributes r_i and r_j by using $\mathcal{A}_{\mathcal{D}}$. It is worth noting that $Game_4$ has not utilized the constructive protocol Q_A to obtain r_i and r_j in order to forge the requests directly with $\hat{P_{1d1}}$ and $\hat{P_{1d2}}$. Since the session utilizes random integers i.e., r_i and r_j and ephemeral keys, $\mathcal{A}_{\mathcal{D}}$ obtaining the proper session key is not more than $\frac{2q_S}{2^{l_b}}$. The above analysis shows that the steps involved in $Game_4$ and $Game_5$ are identical unless $\mathcal{A}_{\mathcal{D}}$ successfully guesses random bits r_i and r_j. Hence, we have

$$|P_r[S_5]-P_r[S_4]|\leq\frac{2q_S}{2^{l_b}} \tag{21}$$

$Game_6$ In this game, we consider that $\mathcal{A}_{\mathcal{D}}$ finds S_k by using the previous or last session key establishment between $D1_i$ and $D2_j$. According to the applied strategy, the parameters involved in $H(r_i,r_j)=H(\hat{r_i},r_j)=H(r_i,\hat{r_j})$ cannot be related to any other session S_k as they are generated at random to pass its arguments. Since it does not hold any winning probability over $\mathcal{A}_{\mathcal{D}}$, we thus have the advantage of playing $Game_6$ as,

$$|P_r[S_6]-P_r[S_5]|=0 \tag{22}$$

However, $\mathcal{A}_{\mathcal{D}}$ may have a non-zero advantage over the passing arguments to represent,

1) Set *Impersonation* is *True*;
2) Otherwise, Use *Property of Key Freshness* as *True*; and
3) Set *Key Length* $\langle l^{'}_b=l_b\rangle$.

Let us assume that *impersonation attack* is true while $\mathcal{A}_{\mathcal{D}}$ forges the processing query to interrupt the protocol Q_A. Using $Game_3$, $\mathcal{A}_{\mathcal{D}}$ tampers the parameter Q_A to exclude the use of either $\hat{P^{'}_{1d1}}$ or $\hat{P^{'}_{1d2}}$ to recover long-term secret key S_k. However, $Game_4$ aborts its probability of occurrence with $\mathcal{A}_{\mathcal{D}}$ whenever it attempts to recover the attributes under *CCM*. Hence, in the use of $Game_6$, *impersonation attack* is always set to be *False*. Additionally, it is worth noting that we already set the property of *Key Freshness* as *False* to analyze whether $\mathcal{A}_{\mathcal{D}}$ sets the guessing bits $l^{'}_b$ to 0 or not. Accordingly, we use the guessing probability over $l^{'}_b=l_b$ to define success ratio $\frac{1}{2}$ leading to 0 defined by $\mathcal{A}_{\mathcal{D}}$. To analyze the source features, $\mathcal{A}_{\mathcal{D}}$ may attempt to discover the value S_k which may use $Game_1$ to draw them out of the hash function uniformly using random oracle. Hence, in $Game_6$, the winning probability for $\mathcal{A}_{\mathcal{D}}$ is

$$P_r[S_6]=\frac{1}{2} \tag{23}$$

Considering the differences in winning probabilities i.e., from Eq.16 to Eq.23, we have

$$\begin{aligned}\mathcal{ADV}^{\mathcal{SSK}}_{\diamond_\infty}&=|P_r[S_0]-\frac{1}{2}|=|P_r[S_0]-P_r[S_6]|\\&\leq 3\epsilon_{ec}+\epsilon_{pr}+\frac{7q_S+31(q_S+q_E)^2+q^2_H}{2^{l_{crp}}}\blacksquare\end{aligned} \tag{24}$$

B. Informal Analysis

In this section, we explore a few key attributes of the proposed DC-LADAR numerically to show its security efficiency in B5G networks. The most significant properties are as follows.

P_1 **- Proper Mutual Authentication:** Owing to the use of $key=PRF_{S_k}(H(I_d))\oplus H^{'}(S_{k_0})$, $\langle SSO\rangle$ gateway can register M_d/M_{sen} to validate its unary identity I_d randomly to set up the connectivity with A_S. As a result of this, the configuration involved with M_d/M_{sen} can rely on a dedicated secret session key to protect the end-users like patients/doctors. Additionally, M_d/M_{sen} uses their system attributes to find its common session key to synchronize its configuration with dedicated server A_S in order to authorize the transmission requests. Thus, we claim the proposed DC-LADAR can uphold the property of mutual authentication to protect the privacy of end-users.

P_2 **- Message Integrity:** In the proposed DC-LADAR, the requests exchanging between M_d/M_{sen} and A_S are well protected via SSO_i gateway which proactively utilize a generated key k_g over Z^*_n to prevent redirection or modification attacks. Thus, we claim the proposed DC-LADAR can uphold the integrity of the transmission request to prevent any unintentional changes to sensitive information.

P_3 **- Secret Session Key Agreement:** The derived secret session key S_{s0} sharing among M_d/M_{sen} and A_S protects the end-users connectivity using a proper transmission request $MSG_6 = SS_{k_i}\oplus H(u_j \parallel T_{S2})$ in order to confirm the user identities. For any legal M_d/M_{sen}, $\langle SSO\rangle$ gateway utilizes its one-time pseudonym to validate the correctness of MSG_6 using $SS^*_{k_i} = MSG_2\oplus H(Y^*_i \parallel T_{S1})$. On the other hand, we apply **Theorem 1** to prove the robustness of session key security under the probability of *PPT* adversary and also show that breaking session key

indistinguishable is negligible to emphasize the property of key freshness depending on the deterministic random numbers. Thus, we claim the proposed DC-LADAR can uphold the session key agreement's property to enhance the communication systems' security level.

P_4 - **Soundness:** Let us assume that M_d/M_{sen} attempted to generate an authentic message via $\langle SSO_i \rangle$ and accordingly, derived its secret key S_k to verify whether the identity of U_{sr} is genuine or not. On the other hand, U_{sr} can obtain its derived key using $key = PRF_{S_k}(H(I_d)) \oplus H^{'}(S_{k_0})$ to generate a message request from M_d/M_{sen}. Additionally, it uses the key pairs of $\langle SSO_i \rangle$ to verify the negotiated S_k over $SS^*_{k_i}$. Thus, we claim the proposed DC-LADAR can uphold the property of computational soundness to protect U_{sr} against *stolen verifier*.

P_5 - **Perfect (Forward) Secrecy:** From P_2 and P_4, we can observe that M_d/M_{sen} uses $\{MSG_6, MSG_7, MSG_8\}$ to find the negotiated key $SS^*_{k_i}$. Precisely, the property of *Soundness* ensures that $\langle SSO_i \rangle$ gateway allows S_k to update at least once in order to signify the use of a one-way secure hash function. Moreover, this function cannot be exploited further to compute the previous S_k even if the current one is exposed to $\mathcal{A}_\mathcal{D}$. Thus, the proposed DC-LADAR supports *Perfect (Forward) Secrecy* to prevent sensitive data from being compromised.

In addition to the above properties, we analyze a few potential attacks such as impersonation, man-in-the-middle, and the known session keys to prove the security efficiency of the proposed DC-LADAR.

R_1 - **Impersonation Attack:** Let us assume that $\mathcal{A}_\mathcal{D}$ tries to impersonate U_{sr}/M_d and accordingly, communicates its message request with another entity to derive the master key of U_{sr}/M_d. Considering U_{sr_i}/M_{d_i} as a practical instance, $\mathcal{A}_\mathcal{D}$ is necessitated to compute legal message requests MSG_1 and MSG_2 to determine the master key of the current authentication process. On the other hand, $\mathcal{A}_\mathcal{D}$ may obtain the random number from U_{sr_i}/M_{d_i}, however, the sensitive information stored in U_{sr_i}/M_{d_i} cannot be tampered as it duly adhere to tamper-proof memory. Additionally, $\mathcal{A}_\mathcal{D}$ cannot derive a valid master key or forge a legal message request to validate the authentication process with U_{sr_i}/M_{d_i} due to irreversible hash function. on account of this, $\mathcal{A}_\mathcal{D}$ cannot even infer any random numbers of U_{sr_i}/M_{d_i} to derive any authorized key to validate their authentication requests. Thus, the proposed DC-LADAR can resist the impersonation attack to prevent fraudulent access.

R_2 - **Man-In-The-Middle Attack:** Assume $\mathcal{A}_\mathcal{D}$ is capable of obtaining the transmission requests MSG_1 to MSG_7 from U_{sr}/M_d in order to generate a forged request to the other entity i.e., MSG^*_1 to MSG^*_7. Since the proposed DC-LADAR can resist impersonation attacks and support the property of mutual authentication, $\mathcal{A}_\mathcal{D}$ cannot generate any forged message request to perform any authentication process with U_{sr}/M_d. Thus, the proposed DC-LADAR can protect the entities against man-in-the-middle attacks.

R_3 - **Reply Attack:** Every transmission request holds an attribute of timestamp T_S to validate the property of freshness to SSO_i i.e., $\{PD_i, G_{SSO_i}, r_i, T_{S_i}\}$. Upon obtaining any request, the receiving entity checks the freshness of T_S using $|T_{S_1} - T_S| \leq \Delta_{T_S}$ to validate the message integrity. Additionally, in the process of key agreement, the proposed DC-LADAR embeds T_S in all the transmitted requests i.e., MSG_1 to MSG_7 to verify the genuineness of key protection. Thus, the proposed DC-LADAR can prevent redundant message requests to resist replay attacks.

R_4 - **Eavesdropping Attack:** Since the given message requests hold the property of freshness, $\mathcal{A}_\mathcal{D}$ cannot extract any system attributes to generate a legitimate request to the other entity. Moreover, $\mathcal{A}_\mathcal{D}$ cannot derive an authentic master key without genuine random numbers r_i of the trusted gateway SSO_i. In other words, U_{sr}/M_d secretly holds the information of the master key using an irreversible hash function to prove the feature of tamper-proof memory. Thus, the proposed DC-LADAR can prevent $\mathcal{A}_\mathcal{D}$ from obtaining S_k to resist the eavesdropping attacks.

R_5 - **Known Session Key Attack:** Using this attack, $\mathcal{A}_\mathcal{D}$ can obtain the session key of U_{sr}/M_d to gain unauthorized access over SSO_i in an effort to capture a valid token i.e., legal session identity. However, $\mathcal{A}_\mathcal{D}$ cannot compute a previous or successive session key SS_{k_i} without proper S_k as it is updated only at the end of each session with U_{sr}/M_d via SSO_i. Thus, the proposed DC-LADAR can hold the malicious activities over known S_K to resist the known session key attacks.

R_6 - **Side-Channel Attack:** Using this attack, $\mathcal{A}_\mathcal{D}$ tries to capture the activities of the associated entities E_U and M_D in order to extract the secret tokens reserved in their storage space. To probe this attempt, M_d/M_{sen} and A_S shares their parameters $\{PE_k, PE^{'}_k, I_d, Key, p, g, q\}$ with $\mathcal{A}_\mathcal{D}$ during registration phase. On obtaining the parameter values, $\mathcal{A}_\mathcal{D}$ attempts to compute $Puk_{M_{sens}-M_S} = \langle S_M S.key \rangle$. However, $\mathcal{A}_\mathcal{D}$ cannot determine a legitimate transmission request $T_M = (H(M_{sid} \,\|\, S_{MS} \,\|\, n \,\|\, Pu_{M_{sens}-M_S}) \oplus H(Puk_{M_{sens}-M_S})$ because the integrity of transmission is verified with A_S through SSO gateway. Thus, the proposed DC-LADAR can resist both side-channel and physical capture attacks.

VI. Performance Analysis

This section discusses the assessment of the computation costs and the simulation analysis in order to show the significance of execution times and the number of message rounds in the proposed DC-LA compared with schemes [15]–[18]

A. Computation Analysis

Because of their limited energy resources, wireless medical sensor networks are designed to use lightweight authentication schemes employing only a one-way hash function. Please note that the hash function requires very little computation time, unlike other cryptographic operations (public-key cryptographic functions and symmetric key encryption/decryption), to produce an effective data structure, mapping the encoded data into a fixed value. As provided in [18], execution times for the key hashing $\langle T_{hash} \rangle$, symmetric encryption/decryption $\langle T_{(E/D)} \rangle$, and fuzzy extractor $\langle T_{(fe)} \rangle$ are set as follows: $\langle T_{hash} \rangle$=0.0004 s, $\langle T_{(E/D)} \rangle$=0.1303 s, and $\langle T_{(fe)} \rangle$=0.0292 s. To simulate the process involved in the key agreement phase, we set up a computing service using the Google Cloud Platform operating over a dedicated system equipped with an AMD Ryzen 7 5700U 4.3GHz, 16GB RAM, running 64-bit Ubuntu 22.04.3 LTS.

Additionally, to find the execution times of the proposed DC-LADAR and other existing schemes [9], [15]–[20], we created a Linux-based virtual environment in Google Cloud to provide end-to-end service connectivity using Google Galaxy Nexus, installed on an Intel Core 2 Quad 2.66GHz with 4GB RAM running Android Ver. 4.0.3. Table **??** compares computation costs for A_S, SSO gateway, and M_d/M_{sen} when using the proposed DC-LADAR and other authentication schemes. The comparison results prove that the proposed DC-LADAR had a shorter execution time and fewer message rounds, which improved the architectural efficiency of the medical sensor networks. From Table II, it is also evident that the DC-LADAR mechanism can reduce computation costs $\approx 0.0184s$ to overcome the constraints on medical sensor networks, including battery-power limitations, in comparison with other existing schemes. Please note that in the proposed DC-LADAR, the entities E_U, M_D,

TABLE II: Comparative Assessment of Computation Cost

Schemes	Computation Cost	Execution Time $\langle s \rangle$	Message Rounds
MAuth	$14T_{hash}$+$6T_{E/D}$	0.787	3
SAuth	$24T_{hash}$+$6T_{E/D}$	0.794	3
BAuth	$19T_{hash}$+$4T_{E/D}$	0.529	3
LAuth	$34T_{hash}$+$4T_{E/D}$	0.535	4
AKA-A	$22T_{hash}$+$10T_{E/D}$	1.312	4
SLA	$29T_{hash}$+$2T_{fe}$	0.07	4
UA-KE	$30T_{hash}$+$1T_{fe}$	0.0412	4
Proposed DC-LADAR	$46T_{hash}$	0.0184	3
Symmetric Encryption/Decryption $(T_{E/D})$; Hash Function (T_{hash}); and Fuzzy Extractor T_{fe}			

and A_S can optimize the data size of the computational parameters via SSO gateway to significantly improve the capabilities of low-power devices in order to achieve better computation efficiency.

B. Communication Analysis

To evaluate the communication efficiencies of the proposed DC-LADAR and other existing schemes, we consider the length of a few key attributes namely, the hash function h_f, random integers r_i, timestamp t_s, symmetric encryption/decryption function s_{ed}, and key elements in elliptic-curve cryptography e_{ecc} as cited in [16]. The key sizes of the attributes are as follows: $h_f \approx 160$ bits, $r_i \approx 32$ bits, $t_s \approx 32$ bits, $s_{ed} \approx 128$ bits, and $e_{ecc} \approx 320$ bits. Additionally, we set the servers' key length $s_{kl} \approx 1024$ bits to estimate the integrity of the message transmission during the login and authentication phase. While estimating the efficiencies of the schemes, we noticed that the proposed DC-LADAR had three message rounds to authorize the session establishment between A_S, SSO gateway, and M_d/M_{sen}. Each authorized session utilizes $\{PD_i^{1*}, MSG_1, MSG_2, MSG_3, MSG_4, T_{S1}\} \approx 512$ bits $\{PD_{SSO_i}^{1*}, MSG_5, MSG_6, MSG_7, MSG_8, T_{S2}\} \approx 704$ bits, and $\{MSG_9, MSG_{10}, MSG_{11}, MSG_{12}, MSG_{13}, MSG_{14}, T_{S1}, T_{S2}, T_{S3}\} \approx 896$ bits to generate an authentic message request.

Therefore, the proposed DC-LADAR's overall cost is $\{512 + 704 + 896\} \approx 2112$ bits. Similarly, we compute the communication costs of other existing schemes and they are as follows: 1696 bits [15], 2212 bits [16], 2656 bits [17], 3456 bits [18], 2720 bits [9], 3232 bits [19], and 2944 bits [20], respectively. As shown in Table III, we can observe that the proposed DC-LADAR consumes less communication cost than other existing schemes to improve the performance efficiency between U_{sr}/M_d via SSO_i except Merabet et al. [15].

C. Storage & Energy Efficiency

To analyze the efficiency of the storage unit in IoMT and monitor its continuous physiological signals (i.e., blood pressure, pulse rate, etc.) via bio-sensing units, we developed a practical testbed with Raspberry Pi3 (Model B+), a wireless access point (802.11.b/g/n), and Bluetooth (V4.1). Importantly, we mounted an integrated pulse oximetry and heart rate sensor, i.e., MAX-30100, on Model B+, allowing its shared bus to detect the signaling units, including pulse oximetry and heart rate. To functionalize key agreement techniques, we preferred to use Python v3.9.2 as a programming tool to access the user interface graphically via the IP Scanner. The interface utilized Google Android Studio to operate the system communication, allowing Firebase to act as an application server to perform device registration, authentication, and data exchange associated with authorized users $\langle 240 \rangle devices$.

Further, the user credentials recorded in the application server provide secure access via a dedicated wireless gateway i.e., SSO to exchange the sensory data to diagnose the conditional state of the patients. The developed interface integrated with a healthcare device executes the login and authentication phases of the proposed DC-LADAR and other existing schemes to analyze the metrics such as storage cost and energy efficiency and resolve the problem related to key exposure. In this study, the bio-sensing units with applied key agreement techniques monitor the vital signs of the registered patients continuously using SSO gateway to analyze the critical factors (storage cost and energy efficiency) involved in the data transmission. The computerized code led the communication protocol to aggregate the data value established in a constructed message. While verifying the performance efficiencies, i.e., storage and energy of the sensing units, we observe that the proposed DC-LADAR reduces the usage rate of system resources to improve the system efficiency of resource-constrained IoMT devices than other existing schemes [9], [15]–[20].

Using this built-in architecture, the deployed sensing units monitor the physiological signals of the registered users via the SSO gateway to examine the transmission rate of the proposed DC-LADAR and other existing schemes. Since the proposed DC-LADAR operates key agreement strategy efficiently via an edge node (with proper mutual authentication) to protect the system against adversarial attacks, it can prudently handle the source attributes to minimize its cost efficiencies including storage and energy to improve the network performance, as shown in Fig. 2a and Fig. 2b.

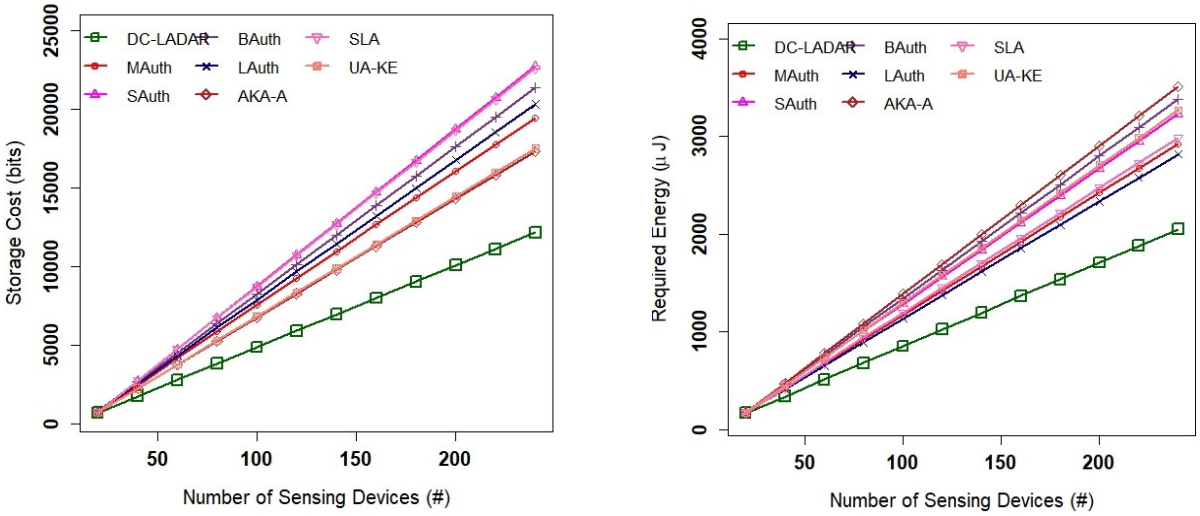
D. Simulation Analysis

In this study, we utilized a discrete-event simulation using NS-3 to analyze a few key metrics of the communication systems, namely packet delivery ratio, latency, and throughput rate [47]. The simulation testbed performed its operation over two Desktop PCs including i7-13700 (comprising 32GB RAM, 2TB HDD, and clock rate 5.2 GHz) and i7-12700H (equipping 32GB RAM, 1TB HDD, and clock rate 4.7GHz) to set the application interface with A_S and to emulate the computing service between the sensing units and SSO gateway. To support tools such as LTE, TCP, and UDP protocols, the testing modules were examined using Python/C++. Table IV shows the important parameters used in the NS-3 simulator. Using Ubuntu 14.04.6 LTS, the proposed DC-LADAR and other existing schemes adopted their login and authentication modules to implant sensing units in a grid layout. In this study, the deployed environment had a few implantable devices, i.e., 20 to 420, verified over 40 devices to probe the key agreement modules of the proposed DC-LADAR and other existing schemes. Additionally, a specific SSO gateway was deployed to carry out data exchange between sensing units and A_S. A smart hospital environment proactively built its simulation model encompassing various IoMT devices (wearable sensing units and diagnostic devices) to transmit medical data via SSO to perform efficient data transmission among the communication entities (E_U, M_D, and A_S). Moreover, it allows E_U and M_D to provide emergency alert through T_A to perform proper diagnosis and treatment plan. Lastly, the generated data by M_D assesses the transmission characteristics of the network to analyze significant communication metrics, such as packet delivery ratio, jitter, and throughput.

Before verifying the identities of the sensing units and application interface, the unrealistic cyber-security environment initiated the authentication process with a few more messages via SSO gateway i.e., wireless sink node to perform a proper session key establishment. Accordingly, the dedicated A_S processed a challenge-response pair

TABLE III: Comparative Assessment of Communication Cost

Schemes	Transmitted Messages	Total Cost $\langle bits \rangle$
MAuth	$\{R_S\}$, $\{R_i, Auth_i, M\}$, and $\{Auth_S\}$	1696
SAuth	$\{M_1, M_2, k_{3_i}, A_{ID_i}\}$, $\{r_1, I_{d_i}\}$, and $\{SS_k, S_d, R_s\}$	2212
BAuth	$\{M_1, M_2, TS_1\}$, $\{M_3, M_4, TS_2\}$, $\{M_5, TS_3\}$, and $\{SK_{U_i, CS_j'}\}$	2656
LAuth	$\{RID_{U_i}, TC_{U_i}, R_i, K_i^{'}\}$, $\{M_1, M_2, M_3, M_4, T_1\}$, $\{M_6, M_7, M_8, T_2\}$, and $\{M_{10}, M_{11}, M_{12}, T_3\}$	3456
AKA-A	$\{I_i, Pid_i, C_k^{'}, A_i^{'}, R_i^{'}\}$, $\{N_i, T_1, E_i\}$, $\{T_i, T_2\}$, and $\{MN_i, T_3\}$	2720
SLA	$\{M_1, Auth_{us}, RID_i, T_1\}$, $\{C_j, M_2, Auth_{sd}, T_2\}$, $\{M_3, Auth_{ds}, Auth_{du}, T_3\}$, $\{M_4, Auth_{du}, Auth_{su}, T_4\}$	3232
UA-KE	$\{TID_{U_k}, M_1, M_2, T_1\}$, $\{M_3, M_4, M_5, T_2, T_1\}$, $\{M_6, M_7, T_3\}$, and $\{M_6, M_8, M_9, T_3, T_4\}$	2944
Proposed DC-LADAR	$\{PD_i^{1*}, MSG_1, MSG_2, MSG_3, MSG_4, T_{S1}\}$, $\{PD_{SSO_i}^{1*}, MSG_5, MSG_6, MSG_7, MSG_8, T_{S2}\}$, $\{MSG_9, MSG_{10}, MSG_{11}, MSG_{12}, MSG_{13}, MSG_{14}, T_{S1}, T_{S2}, T_{S3}\}$	2112



(a) Storage Cost (b) Energy Efficiency

Fig. 2: Performance Analysis Using Raspberry Pi.

TABLE IV: Important Parameters Used in NS-3

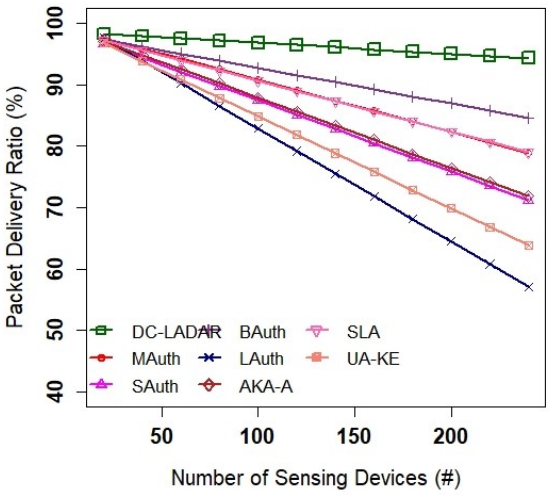
Parameters	**Descriptions**
Area of IoT-Enabled Devices	$400 \times 100\ m^2$
Number of Medical Experts	2 Nos.
Communication Speed	2 ms
Area of Medical Experts	$400 \times 100\ m^2$
Simulation Time	$\langle 1800 sec \rangle$
Language	C++/Python
Communication Protocol	TCP and UDP
Wireless Standard	IEEE 802.11ah
Deployment	Random
Control Message	20 bits
Packet amount	1000
Retransmission Amount	10 (Max.)
Packet Interval	10 μs
Data Rate	88Mbps (Max.)
Slot Amount	32
Slot Period	10 μs
Number of Communication Rounds	150

(CRP) of the registered users through SSO gateway via a secure channel,

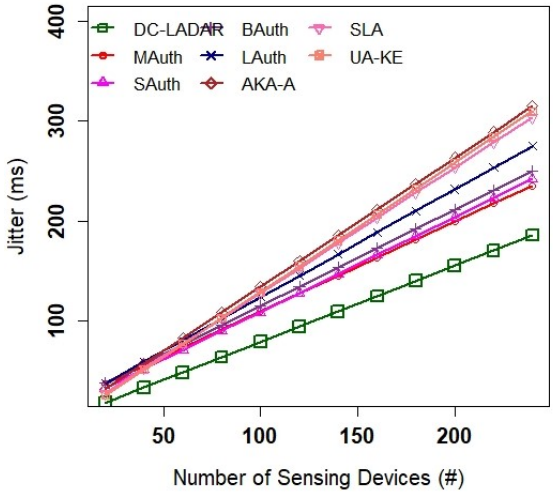
1) To handle data transmission effectively among the sensing units and A_S and test the transmission ratio at the speed of $\langle 2ms \rangle$, we assumed a fully trusted entity, i.e., SSO gateway, randomly among the sensing units;
2) To analyze the data packet transmission in every $\langle 4sec \rangle$, we set IEEE 802.11ah as a wireless standard. Additionally, the configured A_S uses the CPR of the registered users to carry out data exchange among the sensing units and employs SSO gateway (sink node) to update the records of the registered users; and
3) To explore the available sensing units and probe simulation area $400 \times 300\ m^2$, we set a simulation time i.e. $\langle 1800 sec \rangle$ with four referral messages including $\langle 52 bytes \rangle$, $\langle 84 bytes \rangle$, $\langle 116 bytes \rangle$ and $\langle 168 bytes \rangle$ respectively.

The packet delivery ratio is considered to test the successful packet transmission of a network scenario. From Fig. 3a, we can observe that the packet delivery ratio starts dropping when more active users attempt to process the transmission requests via the SSO gateway between sensing units and A_S. Due to more active users, the connected networks experience more processing time to achieve mutual authentication and session key agreement between the sensing units and the A_S. From the simulation, we also noted that the aggregated data from the registered users may cause additional transmission delay when the packet transmissions proportionally grow via a dedicated SSO gateway. However, the proposed DC-LADAR holds the delay with strict limits between the sensing units and A_S to improve the processing cost of the networking environment 96.26% than other existing schemes [9], [15]–[20].

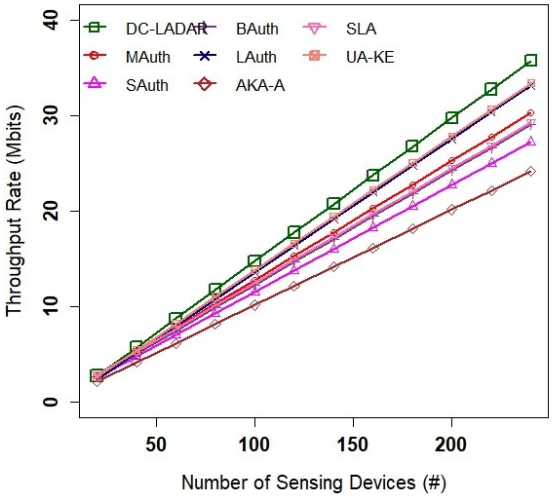
Accordingly, we extended the densities of the computing nodes, including 80 sensing units per SSO gateway, 160 sensing units per SSO gateway, and 240 sensing units per SSO gateway to probe the key metrics such as latency and throughput. From Fig.3b and Fig.3c, we can see that the channel width with 80 MHz provides



(a) Packet Delivery Ratio



(b) Jitter



(c) Throughput Rate

Fig. 3: Simulation Results Using NS3.

better channel efficiency to improve the transmission rates of the proposed DC-LADAR and other existing schemes. In order to validate the modeling scenario, the proposed DC-LADAR and other existing schemes investigate the optimal range of their transmission with an increasing number of sensing units via the SSO gateway. Additionally, the modeling scenario holding the key-index structure of the secret key S_k makes the sensing units to process the encrypted data using SSO gateway, i.e., a wireless sink node to A_S.

Since the channel width was set to 80 MHz, we found that the best transmission range for the sensing units was in the range of 50 m to shorten the interference and wavelength frequencies in order to minimize delay and jitter between the sensing units and A_S. The obtained results (i.e., of Fig. 3b and Fig.3c) reveal that the proposed DC-LADAR manages well in a dense environment with the support accessible SSO gateway via its long-term secret key S_k to satisfy the property of mutual authentication in order to achieve less acceptable jitter-level and improve throughput rate than other existing schemes [9], [15]–[20].

VII. Conclusion

Lately, cloud computing and the Internet of Things (IoT) have converged for the advancement of various wireless multimedia medical sensor networks that allow the user to access the data wirelessly. To improve security and performance efficiencies, a strategy for the DC-LADAR mechanism was proposed that meets the current demands of wireless multimedia medical sensor networks. To achieve less execution time, the proposed DC-LADAR mechanism annulled the clock synchronization problem. In this research, the analysis of elliptic-curve cryptosystems is applied to ensure a lightweight cryptographic function, such as hashing and symmetric encryption/decryption, to withstand various potential attacks, such as forgery, insider, replay, DoS, eavesdropping, and masquerade, etc. In addition, the performance analysis proves that the proposed DC-LADAR can ensure better computation efficiencies in terms of execution time and message rounds than other existing schemes. Additionally, we prove that the proposed DC-LADAR may technologically be agreeable to fulfill the pandemic demands of digital eHealth, such as security and privacy, and to improve the transmission efficiency of the digital processing systems.

REFERENCES

[1] N. Saeed, A. Bader, T. Y. Al-Naffouri, and M.-S. Alouini, "When wireless communication faces covid-19: Combating the pandemic and saving the economy," *arXiv preprint arXiv:2005.06637*, 2020.

[2] K. T. Kadhim, A. M. Alsahlany, S. M. Wadi, and H. T. Kadhum, "An overview of patient's health status monitoring system based on internet of things (iot).," *Wireless Personal Communications*, vol. 114, no. 3, 2020.

[3] M. Mohammed, S. Desyansah, S. Al-Zubaidi, and E. Yusuf, "An internet of things-based smart homes and healthcare monitoring and management system," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1450, 2020, p. 012 079.

[4] W. Abdelghani, C. A. Zayani, I. Amous, and F. Sèdes, "User-centric iot: Challenges and perspectives," in *UBICOMM 2018: The Twelfth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, 2019, pp. 27–34.

[5] S. Hamdan, M. Ayyash, and S. Almajali, "Edge-computing architectures for internet of things applications: A survey," *Sensors*, vol. 20, no. 22, p. 6441, 2020.

[6] X. Jia, D. He, N. Kumar, and K.-K. R. Choo, "A provably secure and efficient identity-based anonymous authentication scheme for mobile edge computing," *IEEE Systems Journal*, vol. 14, no. 1, pp. 560–571, 2019.

[7] S. Jayashree and S. Santhosh Kumar, "Lapep—lightweight authentication protocol with enhanced privacy for effective secured communication in vehicular ad-hoc network," *Wireless networks*, vol. 30, no. 1, pp. 151–178, 2024.

[8] S. Gupta, F. Alharbi, R. Alshahrani, *et al.*, "Secure and lightweight authentication protocol for privacy preserving communications in smart city applications," *Sustainability*, vol. 15, no. 6, p. 5346, 2023.

[9] D. He, Y. Cai, S. Zhu, Z. Zhao, S. Chan, and M. Guizani, "A lightweight authentication and key exchange protocol with anonymity for iot," *IEEE Transactions on Wireless Communications*, 2023.

[10] D. Chaudhary, T. Soni, K. L. Vasudev, and K. Saleem, "A modified lightweight authenticated key agreement protocol for internet of drones," *Internet of Things*, vol. 21, p. 100 669, 2023.

[11] A. Badshah, G. Abbas, M. Waqas, *et al.*, "Usaf-iod: Ultralightweight and secure authenticated key agreement framework for internet of drones environment," *IEEE Transactions on Vehicular Technology*, 2024.

[12] M. Azees, P. Vijayakumar, and L. J. Deboarh, "Eaap: Efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2467–2476, 2017.

[13] B. Gong, C. Guo, C. Guo, Y. Sun, M. Waqas, and S. Chen, "Slim: A secure and lightweight multi-authority attribute-based signcryption scheme for iot," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 1299–1312, 2023.

[14] B. D. Deebak, F. Al-Turjman, M. Aloqaily, and O. Alfandi, "An authentic-based privacy preservation protocol for smart e-healthcare systems in iot," *IEEE Access*, vol. 7, pp. 135 632–135 649, 2019.

[15] F. Merabet, A. Cherif, M. Belkadi, O. Blazy, E. Conchon, and D. Sauveron, "New efficient m2c and m2m mutual authentication protocols for iot-based healthcare applications," *Peer-to-Peer Networking and Applications*, vol. 13, no. 2, pp. 439–474, 2020.

[16] A.-T. Fadi and B. D. Deebak, "Seamless authentication: For iot-big data technologies in smart industrial application systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2919–2927, 2020.

[17] N. Garg, M. Wazid, A. K. Das, D. P. Singh, J. J. Rodrigues, and Y. Park, "Bakmp-iomt: Design of blockchain enabled authenticated key management protocol for internet of medical things deployment," *IEEE Access*, vol. 8, pp. 95 956–95 977, 2020.

[18] M. Wazid, A. K. Das, V. Bhat, and A. V. Vasilakos, "Lam-ciot: Lightweight authentication mechanism in cloud-based iot environment," *Journal of Network and Computer Applications*, vol. 150, p. 102 496, 2020.

[19] S. Yu, A. K. Das, Y. Park, and P. Lorenz, "Slap-iod: Secure and lightweight authentication protocol using physical unclonable functions for internet of drones in smart city environments," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 10, pp. 10 374–10 388, 2022.

[20] M. Wazid, S. Thapliyal, D. P. Singh, A. K. Das, and S. Shetty, "Design and testbed experiments of user authentication and key establishment mechanism for smart healthcare cyber physical systems," *IEEE Transactions on Network Science and Engineering*, 2022.

[21] J. Shen, S. Chang, J. Shen, Q. Liu, and X. Sun, "A lightweight multi-layer authentication protocol for wireless body area networks," *Future generation computer systems*, vol. 78, pp. 956–963, 2018.

[22] V. O. Nyangaresi, "Lightweight anonymous authentication protocol for resource-constrained smart home devices based on elliptic curve cryptography," *Journal of Systems Architecture*, vol. 133, p. 102 763, 2022.

[23] M. Abdussami, R. Amin, and S. Vollala, "Lassi: A lightweight authenticated key agreement protocol for fog-enabled iot deployment," *International Journal of Information Security*, vol. 21, no. 6, pp. 1373–1387, 2022.

[24] K. Chatterjee, R. R. K. Chaudhary, and A. Singh, "A lightweight block cipher technique for iot based e-healthcare system security," *Multimedia Tools and Applications*, pp. 1–30, 2022.

[25] D. Chaudhary, T. Soni, K. L. Vasudev, and K. Saleem, "A modified lightweight authenticated key agreement protocol for internet of drones," *Internet of Things*, p. 100 669, 2022.

[26] M. Kumar and S. Chand, "A provable secure and lightweight smart healthcare cyber-physical system with public verifiability," *IEEE Systems Journal*, 2021.

[27] C. Pu, H. Zerkle, A. Wall, S. Lim, K.-K. R. Choo, and I. Ahmed, "A lightweight and anonymous authentication and key agreement protocol for wireless body area networks," *IEEE Internet of Things Journal*, 2022.

[28] J. Srinivas, A. K. Das, N. Kumar, and J. J. Rodrigues, "Tcalas: Temporal credential-based anonymous lightweight authentication scheme for internet of drones environment," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 7, pp. 6903–6916, 2019.

[29] R. Amin and G. Biswas, "A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks," *Ad Hoc Networks*, vol. 36, pp. 58–80, 2016.

[30] Q. Jiang, S. Zeadally, J. Ma, and D. He, "Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks," *IEEE Access*, vol. 5, pp. 3376–3392, 2017.

[31] Q. Jiang, Y. Qian, J. Ma, X. Ma, Q. Cheng, and F. Wei, "User centric three-factor authentication protocol for cloud-assisted wearable devices," *International Journal of Communication Systems*, vol. 32, no. 6, e3900, 2019.

[32] S. Challa, A. K. Das, V. Odelu, *et al.*, "An efficient ecc-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks," *Computers & Electrical Engineering*, vol. 69, pp. 534–554, 2018.

[33] M. Chowdhury, E. Steinbach, W. Kellerer, and M. Maier, "Context-aware task migration for hart-centric collaboration over fiwi based tactile internet infrastructures," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 6, pp. 1231–1246, 2018.

[34] A. Jindal, G. S. Aujla, and N. Kumar, "Survivor: A blockchain based edge-as-a-service framework for secure energy trading in sdn-enabled vehicle-to-grid environment," *Computer Networks*, vol. 153, pp. 36–48, 2019.

[35] C. Lin, D. He, X. Huang, M. K. Khan, and K.-K. R. Choo, "Dcap: A secure and efficient decentralized conditional anonymous payment system based on blockchain," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2440–2452, 2020.

[36] J. Zhang, J. Cui, H. Zhong, Z. Chen, and L. Liu, "Pa-crt: Chinese remainder theorem based conditional privacy-preserving authentication scheme in vehicular ad-hoc networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 2, pp. 722–735, 2019.

[37] S. Tu, A. Badshah, H. Alasmary, and M. Waqas, "Eake-wc: Efficient and anonymous authenticated key exchange scheme for wearable computing," *IEEE Transactions on Mobile Computing*, vol. 23, no. 5, pp. 4752–4763, 2023.

[38] C. Lin, X. Huang, and D. He, "Ebcpa: Efficient blockchain-based conditional privacy-preserving authentication for vanets," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 3, pp. 1818–1832, 2022.

[39] B. Gong, G. Zheng, M. Waqas, S. Tu, and S. Chen, "Lcdma: Lightweight cross-domain mutual identity authentication scheme for internet of things," *IEEE Internet of Things Journal*, vol. 10, no. 14, pp. 12 590–12 602, 2023.

[40] F. Wang, J. Cui, Q. Zhang, D. He, and H. Zhong, "Blockchain-assisted flexible revocable anonymous authentication in industrial internet of things," *IEEE Transactions on Network Science and Engineering*, 2024.

[41] X. Zhang, H. Zhong, C. Fan, I. Bolodurina, and J. Cui, "Cbacs: A privacy-preserving and efficient cache-based access control scheme for software defined vehicular networks," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1930–1945, 2022.

[42] L. Wei, J. Cui, Y. Xu, J. Cheng, and H. Zhong, "Secure and lightweight conditional privacy-preserving authentication for securing traffic emergency messages in vanets," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1681–1695, 2020.

[43] L. Hernandez, H. Cao, and M. Wachowicz, "Implementing an edge-fog-cloud architecture for stream data management," in *2017 IEEE Fog World Congress (FWC)*, IEEE, 2017, pp. 1–6.

[44] N. Nist, "Recommended elliptic curves for federal government use," *Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. SP*, pp. 800–186, 1999.

[45] A. A. Khan, V. Kumar, and M. Ahmad, "An elliptic curve cryptography based mutual authentication scheme for smart grid communications using biometric approach," *Journal of King Saud University-Computer and Information Sciences*, 2019.

[46] D. Dolev, C. Dwork, and M. Naor, "Non-malleable cryptography," in *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, 1991, pp. 542–552.

[47] L. Tian, S. Deronne, S. Latré, and J. Famaey, "Implementation and validation of an ieee 802.11 ah module for ns-3," in *Proceedings of the Workshop on ns-3*, 2016, pp. 49–56.

B D Deebak received his Ph.D. degree in computer science from SASTRA Deemed University, Thanjavur, India, in 2016. He is currently a Brain Pool Fellow with the Department of Computer Engineering, Department Computer Engineering, Gachon University, Seongnam 13120, Korea. He is an Active Member of professional societies like IE (I), CSI, and ISTE. His research interests include multimedia networks, network security, the Internet of Things, and machine learning.

SEONG OUN HWANG (Senior Member, IEEE) received his Ph.D. degree in computer science from the Korea Advanced Institute of Science and Technology, in 2004, South Korea. He is currently a Full Professor at the Department of Computer Engineering, Gachon University, Seongnam 13120, South Korea. His research interests include cryptography, cybersecurity, data-centric artificial intelligence, and artificial intelligence.