Rotating behind Privacy: An Improved Lightweight Authentication Scheme for Cloud-based IoT Environment

SHEHZAD ASHRAF CHAUDHRY, Department of Computer Engineering, Faculty of Engineering and Architecture, Istanbul Gelisim University, Istanbul, Turkey
AZEEM IRSHAD, Department of Computer Science, International Islamic University Islamabad, Pakistan
KHALID YAHYA, Department of Mechatronics Engineering, Faculty of Engineering and Architecture, Istanbul Gelisim University, Istanbul, Turkey
NEERAJ KUMAR, Computer Science and Engineering Department, Thapar Institute of Engineering and Technology, India Department of Computer Science and Information Engineering, Asia University, Taiwan Department of Computing, UPES, India King Abdul Aziz University, Saudi Arabia
MAMOUN ALAZAB, College of Engineering, IT and Environment, Charles Darwin University, 0810 NT, Australia
YOUSAF BIN ZIKRIA, Department of Information and Communication Engineering, Yeungnam University, South Korea

The advancements in the internet of things (IoT) require specialized security protocols to provide unbreakable security along with computation and communication efficiencies. Moreover, user privacy and anonymity has emerged as an integral part, along with other security requirements. Unfortunately, many recent authentication schemes to secure IoT-based systems were either proved as vulnerable to different attacks or prey of inefficiencies. Some of these schemes suffer from a faulty design that happened mainly owing to undue emphasis on privacy and anonymity alongside performance efficiency. This article aims to show the design faults by analyzing a very recent hash functions-based authentication scheme for cloud-based IoT systems with misunderstood privacy cum efficiency tradeoff owing to an unadorned design flaw, which is also present in many other such schemes. Precisely, it is proved in this article that the scheme of Wazid et al. cannot provide mutual authentication and key agreement between a user and a sensor node when there exists more than one registered user. We then proposed an improved scheme and proved its security through formal and informal methods. The proposed scheme completes the authentication cycle with a minor increase in computation cost but provides all security goals along with privacy.

Authors' addresses: S. A. Chaudhry, Department of Computer Engineering, Faculty of Engineering and Architecture, Istanbul Gelisim University, Istanbul, Turkey, 34310; email: sashraf@gelisim.edu.tr; A. Irshad, Department of Computer Science, International Islamic University Islamabad, Islamabad, Pakistan, 44000; email: irshadazeem2@gmail.com; K. Yahya, Department of Mechatronics Engineering, Faculty of Engineering and Architecture, Istanbul Gelisim University, Istanbul, Istanbul, Turkey, 34310; email: koyahya@gelisim.edu.tr; N. Kumar, Computer Science and Engineering Department, Thapar Institute of Engineering and Technology, Patiala, India, Department of Computer Science and Information Engineering, Asia University, Taiwan, Department of Computing, UPES, Dehradun, India, King Abdul Aziz University, Jeddah, Saudi Arabia; email: neeraj.kumar@thapar.edu; M. Alazab, College of Engineering, IT and Environment, Charles Darwin University, 0810 NT, Darwin, Australia; email: alazab.m@ieee.org; Y. B. Zikria (corresponding author), Department of Information and Communication Engineering, Yeungnam University, Gyeongsan, 38541, South Korea; email: yousafbinzikria@ynu.ac.kr.

1533-5399/2021/05-ART78 $15.00
https://doi.org/10.1145/3425707

CCS Concepts: • **Security and privacy → Mobile and wireless security**; **Security protocols;**

Additional Key Words and Phrases: Security, key-agreement, cloud security, IoT, incorrectness, traceability, anonymity

ACM Reference format:
Shehzad Ashraf Chaudhry, Azeem Irshad, Khalid Yahya, Neeraj Kumar, Mamoun Alazab, and Yousaf bin Zikria. 2021. Rotating behind Privacy: An Improved Lightweight Authentication Scheme for Cloud-based IoT Environment. *ACM Trans. Internet Technol.* 21, 3, Article 78 (May 2021), 19 pages.
https://doi.org/10.1145/3425707

1 INTRODUCTION

The concept of the **internet of things (IoT)** is an emerging framework signifying toward new directions in the evolution of the internet. The IoT era has radically transformed the ways of transmitting the information. IoT-based technology has not only realized the goal of accessibility and connection to every physical object in our surroundings but has almost revolutionized the whole communication ecosystem. The burgeoning applications of the IoT paradigm may encompass patient healthcare, smart homes, smart transportation and vehicles, wearable devices, smart traffic and environment monitoring, smart retail, industrial internet, and geologic disaster forecasting, and so on. Nevertheless, the IoT system comprises sensors with limited power and memory storage, which may find it difficult to manage an unprecedented amount of data produced by those sensors. Cloud computing in IoT-based environment not only enhances the accessibility and reliability of data but also overcomes the scarcity of resourceful back-end servers. Cloud computing provides reliable as well as scalable data storage and computational services. In many critical applications of IoT, the real-time data are collected from the field sensors of tough terrain and then transmitted as well as readily processed by the cloud systems. The remote users may access these critical data from cloud servers at any instant, time, or place if these are duly authorized. In many critical applications, real-time access becomes increasingly important, and it is only possible if we authorize the remote users to access this real-time data from the deployed IoT sensors directly. For this purpose, the user and IoT sensor establish a session key after mutually authenticating one another, which enables the entities to exchange real-time data securely. Thus, security is of utmost concern for such networks, since an adversary may intrude into the system to get illegitimate access to the resources if the proper authentication mechanisms are not followed. Although the cloud-based IoT applications are well addressed and universally accepted, yet to exploit the real gains in those applications, we need not only lightweight authentication protocols but also the protocols free from current known threats. Mostly, the IoT sensors are deployed in hostile environments, and the insecure communication channels that carry this captured information toward servers or intended users, lead to many privacy and security concerns. An adversary may initiate multiple attacks [20, 22, 27, 40] on those protocols including offline guessing attacks, forgery attacks, denial-of-service attacks, and physical devices tampering attacks [4, 9, 23, 34, 36]. As a result, the adversary might perform any unauthorized task on behalf of legitimate participants. Indeed such kind of circumstances renders those protocols inapplicable for practical implementations. Moreover, the devices in the IoT environment are power constrained, which necessitates the construction of lightweight and efficient protocols.

Lately, Wolf and Serpanos [49] presented a security threat-model as well as illustrated different security problems in the cyber-physical system and IoT-based systems. Then, Ni et al. [39] discussed the characteristics and architecture of fog-computing, significance of fog nodes with respect to some promising IoT applications. They also identified some privacy and security threats in

the fog computing domain. Next, Yeh et al. [51] presented **elliptic curve cryptography- (ECC)** based authenticated key agreement protocol for **wireless sensor networks (WSNs)**. Nonetheless, Reference [51] could not provide mutual authentication to the communicants [42]. To counter this threat, Shi and Gong [42] presented a novel ECC-based authenticated key agreement protocol for WSNs. Next, Turkanovic et al. [46] demonstrated another lightweight user authentication protocol for heterogeneous WSNs. Nevertheless, the scheme is vulnerable to the user and sensor node impersonation attack, stolen smart card attack, offline identity, and password guessing attacks. After that, Khalil et al. [30] presented an authentication protocol with a significant contribution of IoT-WSN integration; however, the scheme bears many security loopholes, including replay, impersonation, and sensor node capture attacks. Later, Farash et al. [19] put forward an authentication scheme for the heterogeneous nature of WSNs, equally applicable in the IoT network. Then, Amin et al. [6] designed another related scheme to cover the drawbacks of Farash et al. [19]. Srinivas et al. [45] proved many weaknesses in Reference [6], including user and server spoofing attacks, stolen smart card attacks, and exposure of long-term systems as well as sensor keys. Besides, Reference [45] also presented an improved authenticated key agreement protocol for IoT-based WSNs. Moreover, Jiang et al. [29] demonstrated that Amin et al. [6] was prone to several threats. To counter the security loopholes in Reference [6], Jiang et al. [29] presented an improved three-factor authentication protocol for IoT-enabled WSNs. Hsieh and Leu [24] introduced an enhanced authentication protocol after identifying weaknesses in References [17, 31, 47]. Thereafter, Wu et al. [50] found many drawbacks in Reference [24], such as offline guessing attack, privileged insider attack, sensor node capture attack, and forgery attacks. Moreover, the scheme [24] does not confer session-key security or mutual authenticity to the participants. To counter such weaknesses, Wu et al. presented an ECC-based authentication protocol for IoT-enabled WSN. Later, Challa et al. [10, 11] presented two ECC-based authenticated key agreement protocols for IoT-enabled applications. Chaudhry et al. [15] argued that both of the schemes of Challa et al. suffer from correctness issues and cannot provide a key agreement between a server and a user in IoT-based environments. Similarly, Das et al. also proposed another scheme [16] for industrial IoT. Hussain and Chaudhry [25] proved some critical flaws in their scheme and suggested some remedies. In 2020, Ali et al. [3] proved that the scheme presented in Reference [44] is also incorrect and insecure against several attacks. Some more recent articles on latest developments are as follows [2, 13, 21, 33, 38, 41, 43].

1.1 Motivations and Contributions

Recently, a number of authentication schemes were proposed to secure IoT-based and related systems [8, 10, 11, 44, 48] with faulty design owing to an undue emphasis on privacy cum efficiency tradeoff. This article aims to analyze one of the most recent schemes [48] "lightweight authentication mechanism for IoT-based cloud systems" *LAM-CIoT* proposed by Wazid et al. to show the critical design flaws present in the same. Precisely, *LAM-CIoT* cannot work when there is more than one user registered with the system. The normal working of *LAM-CIoT* can be realized only when there is one and only system user. Such a single user system is unrealistic and cannot be generalized. It is argued in this study that the emphasis on untraceable security along-with the computation and communication efficiencies caused such severe flaws resulting in the complete in-applicability of the scheme to work with more than one user. Unfortunately, same flaw exists in some other schemes [8, 10, 11, 44] as well. While trying to accommodate the untraceable anonymity and efficiency together, the designers of these protocols [8, 10, 11, 44] stuck into the correctness issue. Moreover, this article introduces an improved, lightweight authentication mechanism for IoT-based cloud systems (*iLAM-CIoT*). *iLAM-CIoT* is designed carefully to avoid any such incorrectness and to provide a better tradeoff between security and efficiency. *iLAM-CIoT* is tested

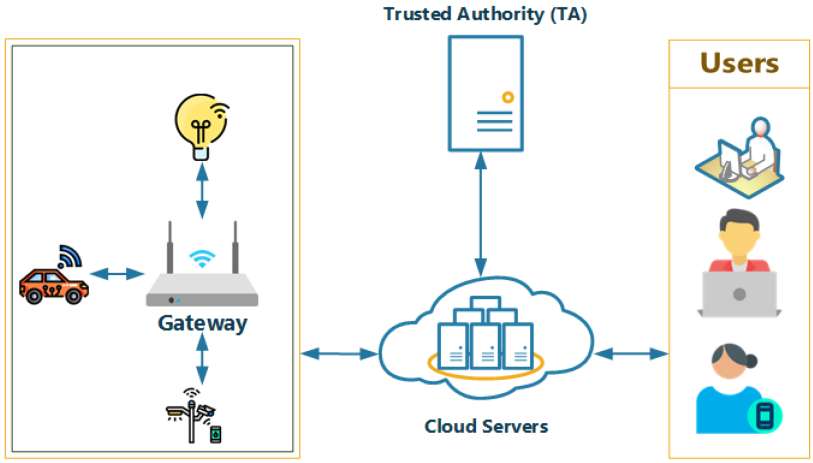

Fig. 1. System model.

under a formal security model supplemented by a brief discussion on attack resilience. This article is organized as follows: Section 2 explains the network and attack models, while the Section 3 provides a brief review of *LAM – CIoT* proposed by Wazid et al. Section 4 proves the inaction of Wazid et al.'s *LAM – CIoT* to share a session key among user and sensor, when multiple users register with the system.. Proposed *iLAM – CIoT* is presented in Section 5, whereas Section 6 provides the security analysis of *iLAM – CIoT* using formal ROR model supported by an informal discussion on security features and threat avoidance. The comparisons with respect to security and performance are given in Section 7. Last, Section 8 presents the concluding remarks.

2 SYSTEM MODEL

This section presents the network and attack model to elaborate on the working of the contributed system model.

2.1 Network model

The network model of the proposed scheme is depicted in Figure 1. This model comprises different types of nodes, i.e., $\mathcal{U}_i$, gateway $\mathcal{GW}$, and IoT sensors $\mathcal{SN}_j$. The IoT sensors ($\mathcal{SN}_j$) may be deployed in remote and sensitive areas, while the users may need to access the data directly, as soon as it is captured from $\mathcal{SN}_j$. This calls for establishing a mutually agreed session key between $\mathcal{U}_i$ and $\mathcal{SN}_j$, and then these nodes may securely communicate with each other. The $\mathcal{GW}$ enables $\mathcal{SN}_j$ to communicate with remote users $\mathcal{U}_i$, since $\mathcal{GW}$ is a gateway to the internet. The cloud servers store the entire sensed data captured from sensors through $\mathcal{GW}$. A trusted authority ($\mathcal{TA}$) initializes the system set up by storing the crucial parameters in the memories of $\mathcal{SN}_j$ and $\mathcal{GW}$ before deployment. $\mathcal{U}_i$ gets registered itself from $\mathcal{TA}$ using a confidential channel. In IoT sensor-based environment, a user may access the sensor node's data either directly through real-time capturing or may retrieve the previously stored data on the cloud server. In many critical applications such as battlefield or healthcare domain, data need to be accessed in real-time from the IoT sensors. Alternatively, the data may also be accessed from the cloud servers, which are kept updated and processed with the data collected from different IoT sensors through gateway nodes. The direct access of the IoT sensor's data by the user requires constructing an agreed and

Table 1. Notation Guide

Notations	Description
$\mathcal{SN}_j, \mathcal{U}_i, \mathcal{TA}$	Sensor, User, Trusted Authority
PW_i, SC_i,ID_i	$\mathcal{U}_i$'s password, smartcard, & identity
BIO_i SK_{ij}	Biometrics, Session key
$\mathcal{GW}$, ID_i, ID_j	Gateway, $\mathcal{U}_i$ and $\mathcal{SN}_j$'s Identities
σ_i, τ_i	Biometrics: Secret key& reproduction parameter
$Gen(..)$, $Rep(..)$	Generation & Reproduction functions
p, Z_p, E_p	Large prime, Finite Prime Field, Elliptic curve
T_x, ΔT	Time stamp of x^{th} party, Max. allowable delay
$h(.)$, $\oplus$, $\|\|$	Hash, XOR, Concatenation functions

secure session key between both entities, in prior. The constrained resources of IoT sensors call for a lightweight, but secure key agreement protocol between IoT sensor and user.

2.2 Attack model

We employ the widely known **Dolev-Yao (DY)** threat model [18] for the analysis of the proposed model. The DY model assumes a public and insecure channel for communication between $\mathcal{U}_i$ and $\mathcal{SN}_j$. Employed in a variety of proposals [5, 14, 26, 28, 35], the precise points regarding the adopted attack model are given as follows:

- The adversary is considered as competent enough to overhear, intercept, modify, delete, or synthesize any message exchanged during transmission between the legitimate participants.
- The adversary may extract and examine the credentials recovered [37] from the IoT sensor, to further manipulate for its malicious intention, such as session key computation or initiating of forgery attacks.
- The identities are public and assumed to be known by all internal and external participants.
- The secret keys of $\mathcal{TA}$ and $\mathcal{GW}$ are secure, and no adversary has the capabilities to expose the secret keys.
- The gateway is assumed to be a physically protected system that also stores significant registration secrets enabling the entities in the verification process.

3 REVIEW OF *LAM-CIOT*

The detail of Wazid et al.'s *LAM-CIoT* is given in the following subsections. For clarity, the notation wrap-up is given in Table 1.

3.1 Pre-deployment Phase

For pre-deployment of sensor nodes, $\mathcal{TA}$ picks 160 bit shared key K_j and for each sensor node $S_j : \{j = 1 \ldots n\}$, $\mathcal{TA}$ selects corresponding unique ID_{Sj} and computes $RID_{Sj} = h(ID_{Sj}||K_j)$ and $TC_{Sj} = h(ID_{Sj}||K_j||RTS_{Sj})$, where RTS_{Sj} represents the registration timestamp of S_j. $\mathcal{TA}$ stores RID_{Sj}, TC_{Sj} and $h(RID_{Sj}||K_j)$ in S_j's memory. Now $\mathcal{TA}$ engenders RID_{Sj}, TC_{Sj}, and $h(RID_{Sj}||K_j)$ for each sensor node $S_j : \{j = 1 \ldots .n\}$ in the memory of gateway. $\mathcal{TA}$ then deploys all sensor nodes and gateway node. The gate way node is placed under a physical lock.

3.2 User Registration

For registration, $\mathcal{U}_i$ picks an identity ID_{ui} and sends it to $\mathcal{TA}$, and upon reception $\mathcal{TA}$ generate K_i and computes $RID_{Ui} = h(ID_{Ui}||K_i)$ and $TC_{Ui} = h(ID_{Ui}||K_i||RTS_{Ui})$. $\mathcal{TA}$ then picks X secretly and computes $R_i = h(ID_{Ui}||X)$ and $K_i^{'} = h(K_i||R_i)$. $\mathcal{TA}$ then issues a smart card SC_i engenders

with $\{RID_{Ui}, TC_{Ui}, R_i, h(.), K_i^{'}\}$ to $\mathcal{U}_i$ through private channel. On receiving SC_i, $\mathcal{U}_i$ after selecting password PW_i, secret number k and imprinting biometrics BIO_i, computes $(\sigma_i, \tau_i) = Gen(BIO_i)$, $RPW_i = h(PW_{Ui}||k)$, $A_i = h(ID_{Ui}||\sigma_i) \oplus k$, $F_i = h(RID_{Ui}||RPW_i||\sigma_i||R_i||TC_{Ui})$, $R_i^* = R_i \oplus h(ID_{Ui}||k)$, $RID_{Ui}^* = RID_{Ui} \oplus h(k||\sigma_i)$, $TC_{ui}^* = TC_{Ui} \oplus h(ID_{ui}||\sigma_i)$ and $K_i^* = K_i^{'} \oplus h(ID_{ui}||k||PW_i||\sigma_i)$. $\mathcal{U}_i$ then replaces $RID_{Ui}, TC_{Ui}, K_i^{'}$ by $RID_{Ui}^*, TC_{Ui}^*, K_i^*$ in SC_i. Finally, the smart card memory contains $\{RID_{Ui}^*, TC_{Ui}^*, K_i^*, R_i^*, F_i, A_i, \tau_i, h(.), Gen(.), Rep(.), t\}$.

3.3 Login Phase

To initiate login, $\mathcal{U}_i$ using his smart card SC_i enters the tuple $\{ID_{Ui}, PW_{Ui}, BIO_{Ui}\}$. The SC_i computes $\sigma_i^* = Rep(BIO_i^*, \tau_i)$, $k = A_i \oplus h(ID_{Ui}||\sigma_i^*)$, $R_i = R_i^* \oplus h(ID_{Ui}||k)$, $RID_{Ui} = RID_{Ui}^* \oplus h(k||\sigma_i^*)$, $TC_{Ui} = TC_{Ui}^* \oplus h(ID_{Ui}||\sigma_i^*)$, $RPW_i^* = h(PW_{Ui}^*||k)$ and $F_i^* = h(RID_{Ui}||RPW_i^*||\sigma_i^*||R_i||TC_{Ui})$. SC_i then checks $F_i^* \stackrel{?}{=} F_i$ and refutes the request if $F_i^* \neq F_i$. In other case, login attempt of $\mathcal{U}_i$ is treated as successful. The SC_i selects T_1, r_1 and computes $K_i^{**} = K_i^* \oplus h(ID_{Ui}||k||PW_i^*||\sigma_i^*)$, $M_1 = RID_{Ui} \oplus h(K_i^{**}||T_1)$, $M_2 = RID_{Sj} \oplus h(TC_{Ui}||ID_{Ui}||T_1)$, $M_3 = h(RID_{Ui}||TC_{Ui}||T_1) \oplus r_1$ and $M_4 = h_(ID_{Ui}||RID_{Sj}||TC_{Ui}||r_1||T_1)$. At last, the tuple $Msg_1 = \{M_1, M_2, M_3, M_4, T_1\}$ is sent to $\mathcal{GW}$ by $\mathcal{U}_i$.

3.4 Authenticated Key Agreement

For normal execution of this phase, the following steps are performed between different communicating entities:

Step AP 1: The $\mathcal{GW}$ in response to the received request, checks the freshness of T_1 by comparing with current timestamp $T_{cur} - T_1 \leq \Delta T$, disapproves the request in case of failure and in case of success, $\mathcal{GW}$ computes $RID_{Ui} = M_1 \oplus h(h(K_i||R_i)||T_1)$, $RID_{Sj} = M_2 \oplus h(TC_{Ui}||ID_{Ui}||T_1)$, $r_1 = M_3 \oplus h(RID_{Ui}||TC_{Ui}||T_1)$ and $M_5 = h_(ID_{Ui}||RID_{Sj}||TC_{Ui}||r_1||T_1)$. $\mathcal{GW}$ aborts if $M_5 \neq M_4$; otherwise, generates r_2, T_2 and computes $M_6 = h(TC_{Sj}||RID_{Sj}) \oplus r_2$, $M_7 = h(RID_{Ui}||TC_{Ui}||r_1) \oplus h(TC_{Sj}||T_2)$, $M_8 = h(RID_{Sj}||TC_{Sj}||h(RID_{Sj}||K_j)||r_2||T_2)$ and sends tuple $Msg_2 = \{M_6, M_7, M_8, T_2\}$ to $\mathcal{SN}_j$.

Step AP 2: The $\mathcal{SN}_j$ in response to received request, checks the freshness of T_2 by comparing with current timestamp $T_{cur} - T_2 \leq \Delta T$, disapproves the request in case of failure and in case of success, $\mathcal{SN}_j$ computes $r_2 = M_6 \oplus h(TC_{Sj}||RID_{Sj})$, $h(RID_{Ui}||TC_{Ui}||r_1) = M_7 \oplus h(TC_{Sj}||T_2)$, and $M_9 = h(RID_{Sj}||TC_{Sj}||h(RID_{Sj}||K_j)||r_2||T_2)$. $\mathcal{SN}_j$ aborts if $M_9 \neq M_8$ and otherwise generates r_3, T_3 and computes $M_{10} = h(h(RID_{Ui}||\ TC_{Ui}||r_1)||T_3) \oplus r_3$, $M_{11} = h(h(RID_{Ui}||TC_{Ui}||r_1)||RID_{Sj}||r_3) \oplus h(h(RID_{Sj}||K_j)||T_3)$, $SK_{ij} = h(h(h(RID_{Sj}||K_j)||T_3)||\ h(RID_{Ui}||TC_{Ui}||r_1)||RID_{Sj}||r_3||T_3)$, $M_{12} = h(SK_{ij}||T_3)$ and sends tuple $Msg_3 = \{M_{10}, M_{11}, M_{12}, T_3\}$ to $\mathcal{U}_i$.

Step AP 3: $\mathcal{U}_i$ upon reception of response, checks the freshness of T_3 by comparing with current timestamp $T_{cur} - T_3 \leq \Delta T$, disapproves the request in case of failure and in case of success, $\mathcal{U}_i$ computes $r_3 = M_{10} \oplus h(h(RID_{Ui}||\ TC_{Ui}||r_1)||T_3)$, $h(h(RID_{Sj}||K_j||T_3)) = M_{11} \oplus h(h(RID_{Ui}||TC_{Ui}||r_1)\ ||RID_{Sj}||r_3)$, key $SK_{ij}^{'} = h(h(h(RID_{Sj}||K_j)||T_3)\ ||h(RID_{Ui}||\ TC_{Ui}||r_1)||\ RID_{Sj}||r_3||T_3)$ and $M_{13} = h(SK_{ij}^{'}||T_3)$. $\mathcal{U}_i$ checks $M_{13} \stackrel{?}{=} M_{12}$ and considers $\mathcal{SN}_j$ authenticated on success and keeps $SK_{ij}^{'}$ as shared key with $\mathcal{SN}_j$ for secure communication.

4 WEAKNESSES OF *LAM-CIOT*

This section shows that Wazid et al.'s *LAM-CIoT* could not complete the key agreement and authentication process. Precisely, the gateway upon reception of the request by a specific user could not recognize him and no further communication between the user, gateway, and sensor node occurs. Moreover, the requesting user is also non-recognizable by the sensor node, and he cannot

send any message to the user. For a clear picture, the following explanation of the working of *LAM-CIoT* shows the incorrectness of their scheme:

(1) $\mathcal{U}_i$ after completing a login, computes the following:

$$M_1 = RID_{Ui} \oplus h(K_i^{**}||T_1), \tag{1}$$

$$M_2 = RID_{Sj} \oplus h(TC_{Ui}||ID_{Ui}||T_1), \tag{2}$$

$$M_3 = h(RID_{Ui}||TC_{Ui}||T_1) \oplus r_1, \tag{3}$$

$$M_4 = h_(ID_{Ui}||RID_{Sj}||TC_{Ui}||r_1||T_1). \tag{4}$$

$\mathcal{U}_i$ sends $\{M_1, M_2, M_3, M_4, T_1\}$ to $\mathcal{GW}$.

(2) $\mathcal{GW}$ after verifying freshness of T_1 computes the following:

$$RID_{Ui} = M_1 \oplus h(h(K_i||R_i)||T_1), \tag{5}$$

$$RID_{Sj} = M_2 \oplus h(TC_{Ui}||ID_{Ui}||T_1), \tag{6}$$

$$r_1 = M_3 \oplus h(RID_{Ui}||TC_{Ui}||T_1), \tag{7}$$

$$M_5 = h_(ID_{Ui}||RID_{Sj}||TC_{Ui}||r_1||T_1). \tag{8}$$

(3) To compute RID_{Ui} through Equation (5), $\mathcal{GW}$ needs $h(h(K_i||R_i)||T_1)$. $\mathcal{GW}$ received T_1 and maintains a verifier database containing tuples of the form: $\{ID_{Ui}, RID_{Ui}, TC_{Ui}, h(RID_{Ui}||K_i), h(K_i||R_i)\}$ for each user such that $\{i = 1 \ldots .n\}$. Therefore, to extract $h(K_i||R_i)$, $\mathcal{GW}$ needs to know either ID_{Ui} or RID_{Ui}. However, $\mathcal{GW}$ does not know any of these values. Moreover, to compute RID_{Sj}, r_1, M_5 through Equations (6), (7), and (8), respectively, $\mathcal{GW}$ needs to know TC_{Ui} and to extract TC_{Ui} from verifier database, $\mathcal{GW}$ should receive known ID_{Ui} or RID_{Ui}, both of these are unknown. Therefore, $\mathcal{GW}$ may be unable to compute any parameter and the authentication process may be aborted abnormally.

(4) Similarly, in the reply message $\mathcal{SN}_j$ sends $\{M_{10}, M_{11}, M_{12}, T_3\}$ to $\mathcal{U}_i$- the requesting user, without recognizing him, as the information of requesting user is unknown to $\mathcal{SN}_j$. Furthermore, the message $\{M_6, M_7, M_8, T_2\}$ from $\mathcal{GW}$ does not contain any other information about $\mathcal{U}_i$, rather $\mathcal{GW}$ itself is unable to recognize a specific user. Hence, $\mathcal{SN}_j$ cannot send any message directly to $\mathcal{U}_i$.

5 PROPOSED SCHEME (*ILAM-CIOT*)

This section explains the working of the proposed improved lightweight authentication mechanism for cloud-assisted IoT-based systems (*iLAM-CIoT*). The scheme *iLAM-CIoT*, as summarized in Figure 2, is explained as below.

5.1 Pre-deployment Phase

For pre-deployment of sensor nodes, $\mathcal{TA}$ picks a 160-bit shared key K_j and for each sensor node $S_j : \{j = 1 \ldots n\}$, $\mathcal{TA}$ selects corresponding unique ID_{Sj} and computes $RID_{Sj} = h(ID_{Sj}||K_j)$ and $TC_{Sj} = h(ID_{Sj||K_j}||RTS_{Sj})$, where RTS_{Sj} represents the registration timestamp of S_j. $\mathcal{TA}$ stores RID_{Sj}, TC_{Sj} and $h(RID_{Sj}||K_j)$ in S_j's memory. Now $\mathcal{TA}$ after generating a secret shared key X, engenders RID_{Sj}, TC_{Sj} and $h(RID_{Sj}||K_j)$ for each sensor node $S_j : \{j = 1 \ldots .n\}$ in the memory of gateway along with X. $\mathcal{TA}$ then deploys all sensor nodes and gateway node.

5.2 User Registration

For registration, $\mathcal{U}_i$ picks an identity ID_{ui} and sends it to $\mathcal{TA}$ and upon reception $\mathcal{TA}$ generate the pair $\{K_i, TD_i\}$ and computes $TC_{Ui} = h(ID_{Ui}||K_i)$. $\mathcal{TA}$ further computes $R_i = h(ID_{Ui}||X)$,

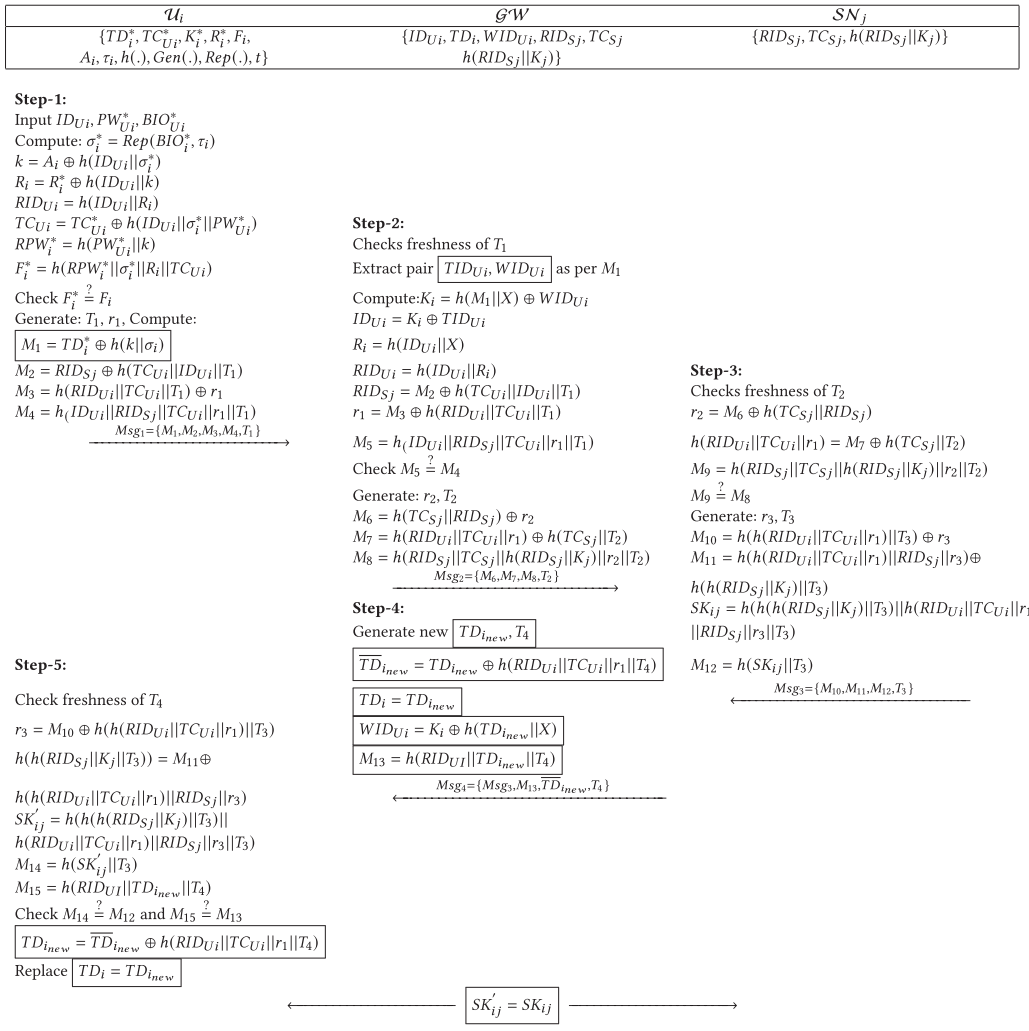

Fig. 2. Proposed *iLAM – CIoT*.

$RID_{Ui} = h(ID_{Ui}||R_i)$ and $K_i^{'} = h(K_i||R_i)$. $\mathcal{TA}$ stores TD_i, $TID_{Ui} = ID_{Ui} \oplus K_i$, $WID_{Ui} = K_i \oplus h(TD_i||X)$ in its memory and sends a smartacrd SC_i engenders with $\{TD_i, TC_{Ui}, R_i, h(.)\}$ to $\mathcal{U}_i$ through private channel. Moreover, $\mathcal{TA}$ also stores the $\{TD_i, TID_{Ui}, WID_{Ui}\}$ on the verifier maintained by corresponding $\mathcal{GW}$. On receiving SC_i, $\mathcal{U}_i$ after selecting password PW_i, secret number k and imprinting biometrics BIO_i, computes $(\sigma_i, \tau_i) = Gen(BIO_i)$, $RPW_i = h(PW_{Ui}||k)$, $A_i = h(ID_{Ui}||\sigma_i) \oplus k$, $F_i = h(RPW_i||\sigma_i||R_i||TC_{Ui})$, $R_i^* = R_i \oplus h(ID_{Ui}||k)$, $TD_i^* = TD_i \oplus h(k||\sigma_i)$ and $TC_{ui}^* = TC_{Ui} \oplus h(ID_{ui}||\sigma_i||PW_{Ui})$. $\mathcal{U}_i$ then replaces TD_i, TC_{Ui} by TD_i^*, TC_{Ui}^* in SC_i. Finally, the smart card memory contains $\{TD_i^*, TC_{Ui}^*, R_i^*, F_i, A_i, \tau_i, h(.), Gen(.), Rep(.), t\}$.

5.3 Login Phase

To initiate login, $\mathcal{U}_i$ using his smart card SC_i enters the tuple $\{ID_{Ui}, PW_{Ui}, BIO_{Ui}\}$. SC_i computes $\sigma_i^* = Rep(BIO_i^*, \tau_i)$, $k = A_i \oplus h(ID_{Ui}||\sigma_i^*)$, $R_i = R_i^* \oplus h(ID_{Ui}||k)$, $RID_{Ui} = h(ID_{Ui}||R_i)$,

$TC_{Ui} = TC^*_{Ui} \oplus h(ID_{Ui}||\sigma^*_i||PW^*_{Ui})$, $RPW^*_i = h(PW^*_{Ui}||k)$ and $F^*_i = h(RPW^*_i||\sigma^*_i||R_i||TC_{Ui})$. SC_i then checks $F^*_i \stackrel{?}{=} F_i$ and refutes the request if $F^*_i \neq F_i$. In other case, login attempt of $\mathcal{U}_i$ is treated as successful. The SC_i selects T_1, r_1 and computes $M_1 = TD^*_i \oplus h(k||\sigma_i)$, $M_2 = RID_{Sj} \oplus h(TC_{Ui}||ID_{Ui}||T_1)$, $M_3 = h(RID_{Ui}||TC_{Ui}||T_1) \oplus r_1$ and $M_4 = h(ID_{Ui}||RID_{Sj}||TC_{Ui}||r_1||T_1)$. At last, the tuple $Msg_1 = \{M_1, M_2, M_3, M_4, T_1\}$ is sent to $\mathcal{GW}$ by $\mathcal{U}_i$.

5.4 Authenticated Key Agreement

Step AC 1: The $\mathcal{GW}$ in response to received request, checks the freshness of T_1 by comparing with current timestamp $T_{cur} - T_1 \leq \Delta T$, disapproves the request in case of failure and in case of success, $\mathcal{GW}$ extracts TID_{Ui}, WID_{Ui} corresponding to M_1 and using the shared secret key X, computes $K_i = h(M_1||X) \oplus WID_{Ui}$, $ID_{Ui} = K_i \oplus TID_{Ui}$, $R_i = h(ID_{Ui}||X)$, $RID_{Ui} = h(ID_{Ui}||R_i)$, $RID_{Sj} = M_2 \oplus h(TC_{Ui}||ID_{Ui}||T_1)$, $r_1 = M_3 \oplus h(RID_{Ui}||TC_{Ui}||T_1)$ and $M_5 = h_(ID_{Ui}||RID_{Sj}||TC_{Ui}||r_1||T_1)$. $\mathcal{GW}$ aborts if $M_5 \neq M_4$ and otherwise generates r_2, T_2 and computes $M_6 = h(TC_{Sj}||RID_{Sj}) \oplus r_2$, $M_7 = h(RID_{Ui}||TC_{Ui}||r_1) \oplus h(TC_{Sj}||T_2)$, $M_8 = h(RID_{Sj}||TC_{Sj}||h(RID_{Sj}||K_j)||r_2||T_2)$ and sends tuple $Msg_2 = \{M_6, M_7, M_8, T_2\}$ to $\mathcal{SN}_j$.

Step AC 2: The $\mathcal{SN}_j$ in response to received request, checks the freshness of T_2 by comparing with current timestamp $T_{cur} - T_2 \leq \Delta T$, disapproves the request in case of failure and in case of success, $\mathcal{SN}_j$ computes $r_2 = M_6 \oplus h(TC_{Sj}||RID_{Sj})$, $h(RID_{Ui}||TC_{Ui}||r_1) = M_7 \oplus h(TC_{Sj}||T_2)$ and $M_9 = h(RID_{Sj}||TC_{Sj}||h(RID_{Sj}||K_j)||r_2||T_2)$. $\mathcal{SN}_j$ aborts if $M_9 \neq M_8$ and otherwise generates r_3, T_3 and computes $M_{10} = h(h(RID_{Ui}||TC_{Ui}||r_1)||T_3) \oplus r_3$, $M_{11} = h(h(RID_{Ui}||TC_{Ui}||r_1)||RID_{Sj}||r_3) \oplus h(h(RID_{Sj}||K_j)||T_3)$, $SK_{ij} = h(h(h(RID_{Sj}||K_j)||T_3)||h(RID_{Ui}||TC_{Ui}||r_1)||RID_{Sj}||r_3||T_3)$, $M_{12} = h(SK_{ij}||T_3)$ and sends tuple $Msg_3 = \{M_{10}, M_{11}, M_{12}, T_3\}$ to $\mathcal{GW}$.

Step AC 3: $\mathcal{GW}$ after receiving Msg_3, checks the freshness of T_3 by comparing with current timestamp $T_{cur} - T_3 \leq \Delta T$, disapproves the request in case of failure and in case of success, $\mathcal{GW}$ generates $\{TD_{i_{new}}, T_4\}$ and computes $\overline{TD}_{i_{new}} = TD_{i_{new}} \oplus h(RID_{Ui}||TC_{Ui}||r_1||T_4)$, $TD_i = TD_{i_{new}}$, $M_{13} = h(RID_{UI}||TD_{i_{new}}||T_4)$ and $WID_{Ui} = K_i \oplus h(TD_{i_{new}}||X)$. $\mathcal{GW}$ further sends the tuple $Msg_4 = \{Msg_3, M_{13}, \overline{TD}_{i_{new}}, T_4\}$ to $\mathcal{U}_i$.

Remark: To avoid any de-synchronization between $\mathcal{GW}$ and $\mathcal{U}_i$, the $\mathcal{GW}$ stores previous TD_i in some temporary variable and on next login, it is updated with current one. The purpose of storing two-session identities is to avoid a situation where $\mathcal{GW}$ updates user identity in its verifier, and the attacker stops the reply message. In such a situation, the user and gateway would have de-synchronized user identities, and the next login may fail.

Step AC 4: $\mathcal{U}_i$ upon reception of response, checks the freshness of T_4 by comparing with current timestamp $T_{cur} - T_4 \leq \Delta T$, disapproves the request in case of failure and in case of success, $\mathcal{U}_i$ computes $r_3 = M_{10} \oplus h(h(RID_{Ui}||TC_{Ui}||r_1)||T_3)$, $h(h(RID_{Sj}||K_j)||T_3)) = M_{11} \oplus h(h(RID_{Ui}||TC_{Ui}||r_1)||RID_{Sj}||r_3)$, $SK'_{ij} = h(h(h(RID_{Sj}||K_j)||T_3)||h(RID_{Ui}||TC_{Ui}||r_1)||RID_{Sj}||r_3||T_3)$, $M_{14} = h(SK'_{ij}||T_3)$ and $M_{15} = h(RID_{Ui}||TD_{i_{new}}||T_4)$. $\mathcal{U}_i$ then checks $M_{14} \stackrel{?}{=} M_{12}$ & $M_{15} \stackrel{?}{=} M_{13}$; Upon success, $\mathcal{U}_i$ computes $TD_{i_{new}} = \overline{TD}_{i_{new}} \oplus h(RID_{Ui}||TC_{Ui}||r_1||T_4)$ and replaces TD_i by $TD_{i_{new}}$ and considers $\mathcal{SN}_j$ authenticated on success and keeps SK'_{ij} as shared key with $\mathcal{SN}_j$ for secure communication.

6 SECURITY ANALYSIS

6.1 Formal Security Analysis

In this subsection, we present the **real-or-random (RoR)** model [1], and then Theorem 1 is pretested to describe the security of the session key for the proposed scheme.

6.1.1 RoR Model. The formal security proof of the proposed scheme is provided using the RoR model. A user $\mathcal{U}_i$, gateway $\mathcal{GW}$, trusted agent $\mathcal{TA}$ and sensor node $\mathcal{SN}_j$ are the participants of proposed scheme.

Participants: $\Pi^a_{\mathcal{U}_i}$, $\Pi^b_{\mathcal{GW}}$, $\Pi^c_{\mathcal{SN}_j}$, $\Pi^d_{\mathcal{TA}}$ indicates the instances a, b, c, d of $\mathcal{U}_i$, $\mathcal{GW}$, $\mathcal{TA}$, and $\mathcal{SN}_j$ respectively, which are treated as oracles [12].

Accepted State: An instance Π^a is called accepted state if it enters into an accept state after receiving the last expected message. The session identification is dealt with as an ordered concatenation of the transmitted messages, including received and sent messages of any instance Π^a for an ongoing session.

Partnering: Suppose that Π^{a1} and Π^{a2} are two instances. They will be considered partners if the following criteria are satisfied:

LA 1: Π^{a1} and Π^{a2} are in accept state.
LA 2: Π^{a1} and Π^{a2} are mutually authenticated and share same session identification.
LA 3: Π^{a1} and Π^{a2} are mutual partners.

Freshness: $\Pi^a_{\mathcal{U}_i}$ or $\Pi^c_{\mathcal{SN}_j}$ is assumed fresh, if the session key shared between $\mathcal{U}_i$, $\mathcal{SN}_j$ is not disclosed to an adversary using Reveal query defined in References [12, 48].

Adversary: Adversary has full control over the communication channel of the network. So, an adversary can not only intercept but can also modify, inject delete the messages in the network. Furthermore, an adversary will have access to subsequent queries:

• **Execute**($\Pi^a, \Pi^b, \Pi^c, \Pi^d$) : After executing this query, adversary can breach all the messages share between the instance $\mathcal{U}_i$, $\mathcal{GW}$, $\mathcal{TA}$ and $\mathcal{SN}_j$. This query is known as an eavesdropping attack.

• Π^a, *message* Upon the execution of this query, an adversary can send a message to a participant entity Pi^a, and can receive the response to the message. Such type of query is modeled as an active attack.

• **Reveal**(Π^a): After the execution of this Reveal query, the current session key SK_{ij} calculated, by $\mathcal{U}_i$ is revealed to adversary.

• **Corrupt Smart Card**($\Pi^a_{\mathcal{U}_i}$): While executing this query, an adversary can easily extract all the information stored in the smart card by stealing or from lost smart card SC_i.

• **Corrupt Sensor**($\Pi^c_{\mathcal{SN}_j}$): Due to all execution of this query, all the credentials stored in $\mathcal{SN}_j$, are leaked to adversary.

According to References [12, 48] both corrupt sensors and corrupt smart card queries are assured due to the weak corruption model.

• **Test**(Π^a): The semantic security of session key SK_{ij} shared between $\mathcal{U}_i$ and $\mathcal{SN}_j$ is tested with the help of this query. Firstly, coin cn is flipped before the commencement of the experiment. The resultant is keep secret from an adversary, and it decides about the result of the Test query. An entity Pi^a will return session key SK_{ij} when cn = 1; otherwise, SK_{ij} will be a random number when $cn = 0$. Other then this scenario, the outcome will be $\perp$.

Here, an important restriction is imposed that an adversary can access only a limited number of *CorruptSmartCard* and *CorruptSensor* queries. But, an adversary can access $Test(\Pi^a)$ query as many times as he wants. Threat model supposes that the $\mathcal{GW}$ and $\mathcal{TA}$ are trusted. So, it is supposed that $\mathcal{A}$ cannot access another corrupt query related to $\mathcal{GW}$ and $\mathcal{TA}$. $\mathcal{A}$ can have huge number of queries to either $\Pi^a_{\mathcal{U}_i}$ or $\Pi^c_{\mathcal{SN}_j}$. The resultant of the test query needs to remain consistent with the random bit bt. If $\mathcal{A}$ wins the game under condition $bt' = bt$, then he returns the

guessed bit $bt^{'}$. If *Succ* event wins the game, then the advantage $Adv_{A}^{iLAM-CIoT}$ of $\mathcal{A}$ in breaking the security of the proposed scheme against deriving SK_{ij} between $\mathcal{U}_i$ and $\mathcal{SN}_j$ is given by $Adv_{A}^{iLAM-CIoT} = |2Pr[Suc] - 1|$.

Random Oracle: A collision resistant hash function h(.) is accessible to all the participating entities including $\mathcal{A}$. We have modeled h(.) as a random oracle, calls O_H [12].

6.1.2 Security Proof. The subsequent Theorem 1 is given to explain the semantic security of *iiLAM − CIoT*.

Theorem 1. *Suppose $\mathcal{A}$ is an adversary running in polynomial time t_m against our proposed protocol in RoR model. D be a uniform distributed dictionary and l be the number of bits presented in biometric key σ_i. The advantage of an adversary to break the security of the proposed scheme is as follow: $Adv_{\mathcal{A}}^{iLAM-CIoT} \leq \frac{q_{hash}^2}{|HASH|} + \frac{q_s}{2^{l-1}|D|}$, where q_{hash}, q_s, $|Hash|$ and $|D|$ are the number of O_H queries, Send queries, the space range of $h(.)$ and size of D, respectively.*

Proof: The sequence of four games $Game_g (g = 0, 1, 2, 3)$ is defined herein which $Succ_g$ indicates an event that $\mathcal{A}$ can guess the bit *bt* in $game_g$ and succeed that particular game. The starting game $GameG_0$ i the real attack and the game finishes with game G_3. Following is the detailed description of all these game.

• **Game** G_0: This game is modeled as real attack in which $\mathcal{A}$ needs to choose bit *bt*. It follows that

$$Adv_{\mathcal{A}}^{iLAM-CIoT} = |2Pr[Succ_0] - 1|. \tag{9}$$

• **Game** G_1: An eavesdropping attack is modeled using this game . *Execute* query follows by *Test* query is initiated by $\mathcal{A}$. After that $\mathcal{A}$, needs to verify that either SK_{ij} is real or random. The calculated session key is $SK_{ij} = h(h(h(RID_{\mathcal{SN}_j}\|K_j)\|T_3)\|h(RID_{\mathcal{U}_i}\|TC_{\mathcal{U}_i}\|r_i)\|RID_{\mathcal{SN}_j}\|r_3\|T_3)$. Suppose that $\mathcal{A}$ breaches the $message_1$, $message_2$, and $message_3$ that are transmitted during authentication phase. However, no message from all these will enable $\mathcal{A}$ to compute valid SK_{ij} and during secret credentials RID_{sj}, k_j, $RID_{\mathcal{U}_i}$, and $TC_{\mathcal{U}_i}$ will be very hard computationally. So, it is proved that both game $GameG_1$ and G_2 are identical, and we got:

$$Pr[Succ_1] = Pr[Succ_0]. \tag{10}$$

• **Game** G_2: Using this query, $\mathcal{A}$ can query O_H and *Send* queries. $\mathcal{A}$ can fabricate any message of participating entity. To create valid messages Msg_1, Msg_2, Msg_3, and Msg_4 $\mathcal{A}$ needs the secret parameters $ID_{\mathcal{U}_i}, K_i, TC_{\mathcal{U}_i}, RID_{\mathcal{U}_i}, K_j$, and $RID_{\mathcal{SN}_j}$. But, all these credentials can kept hidden from $\mathcal{A}$ using hash function. Moreover, session specific random numbers and timestamps are used in each different session. The only difference between $GameG_1$ and G_2 is that in G_2 *Send* and O_H queries are simulated. As per birthday paradox, the resultant is as follows:

$$Pr[Succ_1] - Pr[Succ_2] \leq \frac{q_{hash}^2}{2|Hash|}. \tag{11}$$

• **Game** G_3: In this game, $\mathcal{A}$ calls the *CorruptSensor* and *CorruptSmartcard* queries. Thats why adversary have all information $\{RID^*_{\mathcal{U}_i}, TC_{\mathcal{U}_i}, K_i^*, R_i^*, F_i, A_i, \tau_i, h(.), Gen(.), Rep(.), t\}$ stored in smart card with the help of dictionary attack. $\mathcal{A}$ can guess valid Pw_i of user $\mathcal{U}_i$ from the breached information of SC_i. Therefore, *iLAM-CIoT*implements fuzzy extractor that allows to select some random bits for biometric key $\sigma_i|$. The chances of guessing $\sigma \in 0, 1^l$ by $\mathcal{A}$ is at most $\frac{1}{2^l}$. Additionally, without knowing Pw_i and σ_i, $\mathcal{A}$ can retrieve the important secret credentials $R_i, RID_{\mathcal{U}_i}, T_{\mathcal{U}_i}$, and K_i from R_i^*, $RID^*_{\mathcal{U}_i}$, $T^*_{\mathcal{U}_i}$, and K_i^*. So, without knowing these credentials, it is not feasible to

compute SK_{ij}. Thus, we got:

$$Pr[Succ_2] - Pr[Succ_3] \leq \frac{q_s}{2^l|D|}. \tag{12}$$

Consequently, $\mathcal{A}$ has sent any knowledge of bit bt, because the session keys are generated randomly by $\mathcal{SN}_j$ and $\mathcal{U}_i$, we have the following:

$$Pr[Succ_3] = \frac{1}{2}. \tag{13}$$

We got the following resultant by combining Equations (9), (10), and (13):

$$\begin{aligned} \tfrac{1}{2}Adv_{\mathcal{A}}^{iLAM-CIoT} &= |Pr[Succ_0] - \tfrac{1}{2}| \\ &= |Pr[Succ_1] - Pr[Succ_3]|. \end{aligned} \tag{14}$$

The solution of Equations (11), (12), and (13) gives the following resultant:

$$\begin{aligned} |Pr[Succ_1] - Pr[Succ_3]| &\leq |Pr[Succ_1] - Pr[Succ_2]| \\ +|Pr[Succ_2] - Pr[Succ_3]| &\leq \frac{q_{hash}^2}{2|Hash|} + \frac{q_s}{2^l|D|}, \end{aligned} \tag{15}$$

while Equations (14) and (15) give the following resultant:

$$|Pr[Succ_1] - \frac{1}{2}| \leq \frac{q_{hash}^2}{2|Hash|} + \frac{q_a}{2^l|D|}. \tag{16}$$

Equations (15) and (16) generate the following output:

$$\frac{1}{2}Adv_{\mathcal{A}}^{iLAM-CIoT} \leq \frac{q_{hash}^2}{2|Hash|} + \frac{q_a}{2^l|D|}. \tag{17}$$

Consequently, by multiplying both sides of Equation (17) by 10, the resultant is as follows:

$$Adv_{\mathcal{A}}^{iLAM-CIoT} \leq \frac{q_{hash}^2}{|Hash|} + \frac{q_{send}}{2^{l-1}|D|}. \tag{18}$$

6.2 Functionalities Provision Analysis

This section discusses the security and related features provision analysis of the proposed iLAM-CIoT on informal lines.

6.2.1 Correctness. The proposed iLAM-CIoT completes the authentication process correctly between $\mathcal{U}_i$ and $\mathcal{SN}_j$ through the mediation of $\mathcal{GW}$. The scheme was designed to provide an insight into the common mistake, and it can serve as a template for future works free of such correctness issues. In the user registration phase of the proposed iLAM-CIoT, a random and dynamic identity TD_i is generated by $\mathcal{TA}$, and it is also stored in the memory of user's smartcard as well as in the verifier table maintained by the respective $\mathcal{GW}$. This random identity TD_i is updated after each login and authentication request on both the smart card in possession of the user and verifier maintained by the $\mathcal{GW}$. Moreover, each user has a different random and dynamic identity. Therefore, the $\mathcal{GW}$ on the reception of each authentication request can easily distinguish the distinct users and can extract the corresponding values from the verifier. Hence, proposed iLAM-CIoT does not suffer from correctness issues.

6.2.2 Replay Attack. In the proposed scheme, the exchanged messages $Msg_1 - Msg_4$ among the interacting entities U_i, GW, SN_j employ time stamps $T_1 - T_4$ to foil the potential replay attacks. In the mutual authentication phase, the requirement for communication messages needs to be responded in a limited timestamp threshold ΔT nullifies the chances of an adversary to manipulate the messages and initiate the replay attack. Without updating the parameters $M_2 - M_4$, M_7, M_8, $M_{10} - M_{12}$, the fresh timestamp cannot be employed. At the same time, to update these parameters with updated timestamps, the adversary needs access to U_i's identity, password, biometrics, and other shared factors among the U_i, GW, and SN_j. Hence, the replay attack is not possible in our scheme.

6.2.3 Man-in-the-middle Attack. To initiate the man-in-the-middle attack, an adversary may attempt to reconstruct the Msg_1, i.e., $Msg_1 = M_1 - M_4, T_1$ where $M_1 = \oplus h(k||\sigma_i)$, $M_2 = RID_{Sj} \oplus h(TC_{Ui}||ID_{Ui}||T_1)$, $M_3 = h(RID_{Ui}||TC_{Ui}||T_1) \oplus r1$, $M_4 = h(ID_{Ui}||RID_{Sj}||TC_{Ui}||r_1||T_1)$. However, to meet this objective, the adversary needs to construct those message parametes with fresh timestamp T1, fresh nonce r1, which is not possible until it has access to $k, TC_{Ui}, TD_i^*, ID_{Ui}$, and PW_{Ui} along with biometrics. Likewise, the adversary will not be able to reconstruct other messages $Msg_2 - Msg_4$ in the protocol with updated timestamps and nonces without having access to crucial parameters in possession with those participating entities. Therefore, our scheme is protected from a man-in-the-middle attack.

6.2.4 Privileged-insider Information Leading to Guessing Attack. In case, a malicious privileged insider gets the user's identity IDUi during the registration process, and the adversary may attempt an offline-password guessing attack. Here, we also assume that the adversary has access to the smart card, while its contents such as $\{TD_i^*, TC_{Ui}^*, R_i^*, F_i, A_i, \tau_i, h(.), Gen(.), Rep(.), t\}$ parameters have been approached using power analysis attack [37]. However, the adversary may not be able to compute k, and ultimately the password PW_i from F_i using offline-password guessing attack, where $F_i^* = h(RPW_i^*||\sigma_i^*||R_i||TC_{Ui})$. It is because the crucial biometric factor σ_i^* is not available to the adversary, which makes the computation of RPW_i, R_i, and TC_{Ui} parameters a hard problem for the adversary. Hence, this scheme is immune to an offline-password guessing attack if a privileged insider attempts to initiate an attack.

6.2.5 Impersonation Attack. This sub-section elaborates on the defense of the scheme against user, gateway, and sensor node impersonation attacks.

(1) User impersonation attack: If an attacker attempts to launch an impersonation attack on behalf of user, then it needs to construct a valid login request message Msg_1', i.e., $Msg_1' = M_1' - M_4', T_1'$ where $M_1' = TD_i^* \oplus h(k||\sigma_i)$, $M_2' = RID_{Sj} \oplus h(TC_{Ui}||ID_{Ui}||T_1')$, $M_3' = h(RID_{Ui}||TC_{Ui}||T_1') \oplus r_1'$, $M_4' = h(ID_{Ui}||RID_{Sj}||TC_{Ui}||r_1'||T_1')$ with updated nonce r_1' and timestamp T_1'. Nonetheless, as we see earlier, it is a hard problem to recover $k, TC_{Ui}, TD_i^*, ID_{Ui}, PW_{Ui}$ and biometric values for constructing $Msg_1' = M_1' - M_4', T_1'$. Thus, our protocol is protected from any user-impersonation attack.
(2) GN impersonation attack: Likewise, an adversary may try to instigate a forgery attack toward SN_j on behalf of GN. For this purpose, it needs to construct the $Msg_2' = M_6', M_7', M_8', T_2'$ with fresh nonce r_2' and timestamp T_2' and will require some crucial parameters such as $TC_{Sj}, RID_{Sj}, RID_{Ui}, TC_{Ui}$, and K_j secret. It would be a computationally hard problem to compute those parameters from previously intercepted M_6, M_7, and M_8 messages. Hence, our protocol is safe from the GW impersonation attack.
(3) IoT sensor impersonation attack: In the case where the adversary attempts to launch SN_j impersonation attack toward GW or U_i, it needs to build the message Msg_3 with updated nonce r_3' and timestamp T_3', where $Msg_3 = \{M10', M11', M12', T3'\}$ and $M_{10}' = h(h(RID_{Ui}$

$||TC_{Ui}||r_1)||T_3)oplusr'_3$, $M'_{11} = h(h(RID_{Ui}||TC_{Ui}\ ||r_1)||RID_{Sj}||r'_3) \oplus h(h(RID_{Sj}\ ||K_j)||T'_3)$, $M'_{12} = h(SK_{ij}||T'_3)$. However, the adversary may not be able to construct the valid message parameters in Msg_3 until it is granted access to K_j secret as well as $RID_{Sj}, RID_{Ui}, TC_{Ui}$. Hence, this inability of the adversary to compute or recover those parameters in polynomial time renders our protocol protected against the sensor node impersonation attack.

6.2.6 Anonymity and Untraceability. In mutual authentication phase, the proposed scheme employs random nonces r_1, r_2, and r_3 as well as timestamps T_1, T_2, and T_3 in the communication messages $Msg_1 - Msg_3$. In this manner, the adversary may not distinguish among the communication messages of different sessions, which render our scheme untraceable publicly. At the same time, the adversary may not identify the user's identity, since the messages employ pseudo-identities, i.e., RID_{Sj}, RID_{Ui} in the messages instead of original identities that are bounded under the hardness of one-way hash function bearing collision-resistance features. Thus, our scheme maintains anonymity and untraceability features.

6.2.7 Password Modification Attack. In case, the adversary is able to get unauthorized access to the smart card and its contents $\{TD_i^*, TC_{Ui}^*, R_i^*, F_i, A_i, \tau_i, h(.), Gen(.), Rep(.), t\}$ by employing power analysis attack [37], it may further attempt to update the password PW_i with a forged password PW'_i. However, the smart card will not permit the adversary to proceed for this modification until the right parameters such as identity ID_{Ui}, password PW_i, and biometrics are inserted and locally verified using $F_i^*? = F_i$ condition. Hence, our scheme is resistant to any unauthorized password modification.

6.2.8 IoT Sensor Physical Capture Attack. Let us assume that the adversary has managed to physically capture and compromise the c number of IoT sensors. The robustness of the proposed scheme regarding IoT sensor's physical capture attack may be evaluated as the impact of all secure communications as conciliated by capturing c IoT sensors, excluding the communication related to conciliated sensors directly involved. We assume $Pe(c)$ represents the probability of the adversary's capability for decrypting secure communications between the user and other non-compromised IoT sensors SN'_j after conciliating c the number of IoT sensors. Our scheme may resist well against this attack if the condition of $Pe(c) = 0$ holds. The adversary may get the contents $\{RID_{Sj}, TC_{Sj}, h(RID_{Sj}||K_j)\}$ from the IoT sensor S_j's memory after physical capturing. However, he would only be able to compromise the session key for the seized IoT sensor S_j as mutually agreed with the user. In contrast, the other session keys built among non-compromised IoT-sensors and other users remain safe and protected. Thus, the proposed scheme warrants the robustness against IoT sensor physical capture threat.

6.2.9 Denial of Service Attack. The proposed scheme protects against **denial of service (DoS)** attack as the smart card initially authenticates the Ui with the use of $F_i^*? = F_i$ condition. This condition will only hold valid if the U_i inserts its legal identity ID_{Ui}, password PW_i as well as biometric factors into the smart card to compute $F_i^* = h(RPW_i^*||\sigma_i||R_i||TC_{Ui})$. In this case, the verification of U_i will be successful locally, and then it will be allowed to forward an authentication request to GW. The same procedure is followed in the password and biometric update phases to prevent the incorrect modification of these parameters. Therefore, our scheme is immune to a DoS attack.

6.2.10 Stolen Smart Card Attack. Assume, an attacker steals smart card and gets all of its contents $\{TD_i^*, TC_{Ui}^*, R_i^*, F_i, A_i, \tau_i, h(.), Gen(.), Rep(.), t\}$ using power analysis attack [37]. Nevertheless, it may not be able to compute the low entropy identity ID_{Ui} and password PW_i, since the computation of these parameters depends on the availability of biometric factor σ_i to the attacker.

At the same time, it is a computationally hard problem for an attacker to guess those biometric values. It sufficiently warrants that the adversary may not be able to compute either k, or other parameters such as R_i, TC_{Ui}, and RPW_i. Based on this fact, we can infer that our scheme is immune to stolen smart card attacks.

6.2.11 Session Key Security. In proposed scheme, the computed session key, i.e., $SK_{ij} = h(h(h(RID_{Sj}||K_j)||T_3)||h(RID_{Ui}||TC_{Ui}||r_1)||RID_{Sj}||r_3||T_3)$ is mutually agreed between user U_i and sensor node S_j. The established SK_{ij} remains protected as long as one of the two secrets, i.e., either short-term secret (random nonce r_3 or timestamp T_3) or long-term secret (RID_{Sj} or TC_{Ui} or K_j or RID_{Ui}) is not revealed to the adversary. For instance, if an adversary is able to compromise the short-term secret as used in the protocol, then the session key cannot be successfully computed as long as the long-term secret is hidden. Similarly, if the long-term secret is revealed to the adversary, then the scheme remains protected until the short-term secrets are also exposed to the attacker.

7 SECURITY AND PERFORMANCE COMPARISONS

This section presents a brief security and performance comparison of the proposed scheme with related schemes [7, 8, 10, 11, 19, 48], each of the schemes for comparison purposes is selected based on relevance, significance, as well as a timeline, started from 2016 to 2020. Following subsections present the performance and security comparisons:

7.1 Computation Complexity

Table 2 presents the computed running time of the proposed and related schemes. The notations adopted for computation time representations are briefed as follows: T_{ho}, T_{eds}, and T_{ef} represent the computation complexity of hash (one way), encryption/decryption (symmetric), and fuzzy extractor functions, whereas T_{me} and T_{ae} represent running time for computing multiplication and addition of elliptic curve points. The results of experiment conducted in Reference [32] are taken as it is. As per Reference [32], T_{ho} and T_{eds} take 0.0023 ms and 0.0046 ms for an execution cycle, whereas T_{me} and T_{ae} take 2.226 ms and 0.0288 ms for successful completion. As per References [8, 48], T_{ef} incurs same computation complexity as of T_{me}, so $T_{ef} \approx 2.226$ ms. The experiment in Reference [32] was performed on a E2200 dual PC with a 2.20-GHz processor and 2 GB RAM while using Ubuntu OS and PBC Library. In proposed scheme, for a single cycle of authentication $\mathcal{U}_i$ executes $1T_{ef}$ and $17T_{ho}$ operations, whereas $\mathcal{GW}$ and $\mathcal{SN}_j$ executes $13T_{ho}$ and $9T_{ho}$ operations. Therefore, a single authentication cycle between $\mathcal{U}_i$ and $\mathcal{SN}_j$ completes with the help of $\mathcal{GW}$ by performing $1T_{ef} + 39T_{ho}$ operations collectively. The running time of the proposed scheme to complete the process is 2.3157 ms. Referring to Table 2, the proposed scheme has slightly higher running time in comparison with the scheme proposed by Wazid et al. [48], Farash et al. [19], and Amin et al.[7]. In contrast, it has the same computation cost as of Banerjee et al.'s scheme [8] and has much less complexity in computation cost as compared with the two schemes of Challa et al. [10, 11]. Note that only the proposed scheme provides required security features, whereas all competing schemes are unable to extend some of the security features.

7.2 Communication Complexity

To compute fair and realistic communication complexity of the proposed and related schemes [7, 8, 10, 11, 19, 48] and for simplicity, we consider the size of each identity and random number as 160 bits, while the size of the timestamp is fixed at 32 bits. The selected hash (SHA-1) is 160 bits long, using the announced secure size of ECC point, we adopted 160 bits for each coordinate of the point ($Q = (Q_x, Q_y) = 160 + 160 = 320$ bits) on the curve. The proposed scheme completes authentication in four messages: $Msg_1 = \{M_1, M_2, M_3, M_4, T_1\}$ consumes $160 * 4 + 32 = 672$ bits. In

Table 2. Performance Comparisons

↓Protocols/Cost→	Computation (*ms*)	Bits
iLAM-CIoT	$1T_{ef}$ + $39T_{ho}$= 2.3157	2,560
LAM-CIoT [48]	$1T_{ef}$ + $34T_{ho}$= 2.3042	1,696
Banerjee et al. [8]	$1T_{ef}$ + $19T_{ho}$ + $10T_{eds}$= 2.3157	2,304
Amin et al. [7]	$23T_{ho}$= 0.0529	2,816
Challa et al.-I [11]	$1T_{ef}$ + $14T_{me}$ + $12T_{ho}$= 33.4176	2,528
Challa et al.-II [10]	$1T_{ef}$ + $2T_{me}$ + $20T_{ho}$= 6.724	1,536
Farash et al. [19]	$32T_{ho}$= 0.0736	2,752

Table 3. Security Features

Features↓	Our	[48]	[8]	[7]	[11]	[10]	[19]
$\mathcal{R}_{s1}$	✓	✗	✗	✓	✗	✗	✓
$\mathcal{R}_{s2}$	✓	✓	✓	✗	✓	✓	✓
$\mathcal{R}_{s3}$	✓	✓	✓	✓	✓	✓	✓
$\mathcal{R}_{s4}$	✓	✓	✓	✓	✓	✓	✓
$\mathcal{R}_{s5}$	✓	✓	✓	✓	✓	✗	✓
$\mathcal{R}_{s6}$	✓	✓	✓	✗	✓	✓	✗
$\mathcal{R}_{s7}$	✓	✓	✓	✓	✓	✗	✗
$\mathcal{R}_{s8}$	✓	✓	✓	✓	✓	✓	✓
$\mathcal{R}_{s9}$	✓	✓	✗	✓	✓	✓	✓
$\mathcal{R}_{s10}$	✓	✓	✓	✓	✓	✓	✓
$\mathcal{R}_{s11}$	✓	✓	✓	✓	✓	✓	✗
$\mathcal{R}_{s12}$	✓	✓	✓	✗	✓	✓	✗

Note: $\mathcal{R}_{s1}$: Correctness; $\mathcal{R}_{s2}$: user impersonation; $\mathcal{R}_{s3}$: gateway impersonation; $\mathcal{R}_{s4}$: IOT Sensor impersonation; $\mathcal{R}_{s5}$: Replay; $\mathcal{R}_{s6}$: smart card stolen; $\mathcal{R}_{s7}$: user anonymity; $\mathcal{R}_{s8}$: man in the middle; $\mathcal{R}_{s9}$: stolen verifier; $\mathcal{R}_{s10}$: Sensor physical capturing; $\mathcal{R}_{s11}$: offline password guessing; $\mathcal{R}_{s12}$: Biometric security ✓: Secure or extends; ✗:Insecure against or not provides

message: $Msg_2 = \{M_6, M_7, M_8, T_2\}$, $160 * 3 + 32 = 512$ bits are sent on a channel, while communication complexity of $Msg_3 = \{M_{10}, M_{11}, M_{12}, T_3\}$ is also $160 * 3 + 32 = 512$ bits and for transmission of $Msg_4 = \{Msg_3, M_{13}, \overline{TD}_{i_{new}}, T_4\}$, the required communication cost is 512 + 160 + 32 = 864 bits. Hence, the total communication complexity for the authentication phase of the proposed scheme is 672 + 512 + 512 + 864 = 2,560 bits. Referring to Table 2, the communication complexity of the proposed *iLAM-CIoT* is greater than the schemes of Wazid et al. ($LAM-CIoT$), Banerjee et al., and Challa et al. (I & II), whereas it is less than Farash et al.'s and Amin et al.'s schemes.

7.3 Security Features

Table 3 presents a comparison of the security features extended by each of the competing schemes. Referring to Table 3, the proposed *iLAM-CIoT*achieves the required features related to realistic security under the DY model as outlined in Subsection 2.2, whereas Wazid et al.'s scheme [48] is incorrect and cannot complete authentication as proved in Subsection 4. The schemes of Banerjee et al. [8] and Challa et al. (I & II) [10, 11] also suffer from same incorrectness. Both cannot complete even a single cycle of authentication in the presence of more than one user of the system. In the schemes of References [8, 10, 11, 48], the trusted authority or gateway generates user-specific

credentials without knowledge of the user requesting for authentication. All four schemes in References [8, 10, 11, 48] can work when there is one and only one user registered with the system. If there is only one registered user, the receiving party in advance knows the requesting user's identity. Additionally, the scheme in Reference [8] is vulnerable to stolen verifier attack. Challa et al. (II) [10] is susceptible to replay and impersonation attacks. The schemes of Amin et al. [7] and Farash et al. [19] do not provide biometric security, the scheme of Amin et al. [7] is insecure against the impersonation of the user and stolen card attacks, while the scheme of Farash et al. [19] cannot resist password guessing and card stolen attacks as well as in the absence of user anonymity.

8 CONCLUSION

This article aimed to expose a critical design flaw leading toward complete in-applicability, persistent in some latest authentication schemes. As a case study, a scheme (*LAM-CIoT*) presented in 2020 has been reviewed and analyzed. The analysis in this article has proven that *LAM-CIoT* cannot facilitate mutual authentication as well as the key agreement between users and sensors through gateway owing to the incorrectness. We then proposed an improved scheme, *iLAM-CIoT*, to provide cloud-based mutual authentication and key agreement between users and sensors. The *iLAM-CIoT* is provably secure in the formal model. We have also provided an attack resilience discussion of the proposed *iLAM-CIoT*. The security analysis has shown that *iLAM-CIoT* withstand several attacks and provides known required security features. The comparisons have shown that proposed *iLAM-CIoT* provides the best tradeoff between security and performance as compared with some of the latest schemes.

REFERENCES

[1] M. Abdalla, P. Fouque, and D. Pointcheval. 2005. Password-based authenticated key exchange in the three-party setting. In *Proceedings of the 8th International Workshop on Theory and Practice in Public Key Cryptography (PKC'05)*, Lecture Notes in Computer Science , 65–84.

[2] Shubhani Aggarwal and Neeraj Kumar. 2020. Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges. *Comput. Commun.* 149 (2020), 270–299.

[3] Z. Ali, S. A. Chaudhry, M. S. Ramzan, and F. Al-Turjman. 2020. Securing smart city surveillance: a lightweight authentication mechanism for unmanned vehicles. *IEEE Access* 8 (2020), 43711–43724. https://doi.org/10.1109/ACCESS.2020.2977817

[4] Bander A. Alzahrani, Shehzad Ashraf Chaudhry, Ahmed Barnawi, Abdullah Al-Barakati, and Mohammed H. Alsharif. 2020. A privacy preserving authentication scheme for roaming in IoT-Based wireless mobile networks. *Symmetry* 12, 2 (2020), 287.

[5] M. N. Aman, M. H. Basheer, S. Dash, J. W. Wong, J. Xu, H. W. Lim, and B. Sikdar. 2020. HAtt: hybrid remote attestation for the internet of things with high availability. *IEEE IoT J.* 7, 8 (2020), 7220–7233. https://doi.org/10.1109/JIOT.2020.2983655

[6] R. Amin, S. H. Islam, G. Biswas, M. K. Khan, L. Leng, and N. Kumar. 2016. Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks. *Comput. Netw.* 101 (2016), 42–62.

[7] R. Amin, N. Kumar, G. P. Biswas, R. Iqbal, and V. Chang. 2018. A light weight authentication protocol for IoT-enabled devices in distributed Cloud Computing environment, *Fut. Gener. Comput. Syst.* 78 (2018), 1005–1019.

[8] S. Banerjee, V. Odelu, A. K. Das, J. Srinivas, N. Kumar, S. Chattopadhyay, and K. K. R. Choo. 2019. A provably secure and lightweight anonymous user authenticated session key exchange scheme for internet of things deployment. *IEEE IoT J.* 6, 5 (2019), 8739–8752.

[9] Rajanpreet Kaur Chahal, Neeraj Kumar, and Shalini Batra. 2020. Trust management in social Internet of Things: A taxonomy, open issues, and challenges. *Comput. Commun.* 150 (2020), 13–46.

[10] Sravani Challa, Ashok Kumar Das, Prosanta Gope, Neeraj Kumar, Fan Wu, and Athanasios V. Vasilakos. 2020. Design and Analysis of Authenticated Key Agreement Scheme in Cloud-assisted Cyber–physical Systems. *Future Generation Computer Systems* 108 (2020), 1267–1286.

[11] S. Challa, M. Wazid, A. K. Das, N. Kumar, A. Goutham Reddy, E. Yoon, and K. Yoo. 2017. Secure signature-based authenticated key establishment scheme for future iot applications. *IEEE Access* 5 (2017), 3028–3043. https://doi.org/10.1109/ACCESS.2017.2676119

[12] C. C. Chang, H. D. Le, and A. Provably Secure. 2016. Efficient and flexible authentication scheme for ad hoc wireless sensor networks, *IEEE Trans. Wireless Commun.* 15, 1 (2016), 357–366.
[13] Rajat Chaudhary. 2019. Gagangeet Singh Aujla, Neeraj Kumar, Sherali Zeadally, lattice-based public key cryptosystem for internet of things environment: Challenges and solutions. *IEEE IoT J.* 6, 3 (2019), 4897–4909.
[14] S. A. Chaudhry, H. Alhakami, A. Baz, and F. Al-Turjman. 2020. Securing demand response management: A certificate-based access control in smart grid edge computing infrastructure. *IEEE Access* 8 (2020), 101235–101243. https://doi.org/10.1109/ACCESS.2020.2996093
[15] Shehzad Ashraf Chaudhry, Taeshik Shon, Fadi Al-Turjman, and Mohammed H. Alsharif. 2020. Correcting design flaws: An improved and cloud assisted key agreement scheme in cyber physical systems. *Comput. Commun.* 153 (2020), 527–537. https://doi.org/10.1016/j.comcom.2020.02.025
[16] Ashok Kumar Das, Mohammad Wazid, Neeraj Kumar, Athanasios V. Vasilakos, and Joel J. P. C. Rodrigues. 2018. Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial internet of things deployment. *IEEE IoT J.* 5, 6 (2018), 4900–4913.
[17] M. L. Das. 2009. Two-factor user authentication in wireless sensor networks. *IEEE Trans. Wireless Commun.* 8, 3 (2009), 1086–1090.
[18] D. Dolev and A. Yao. 1983. On the security of public key protocols. *IEEE Trans. Inf. Theory* 29, 2 (1983), 198–208.
[19] M. S. Farash, M. Turkanovic, S. Kumari, and M. Holbl. 2016. An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the internet of things environment. *Ad Hoc Netw.* 36 (2016), 152.
[20] Anwar Ghani, Khwaja Mansoor, Shahid Mehmood, Shehzad Ashraf Chaudhry, Arif Ur Rahman, and Malik Najmus Saqib. 2019. Security and Key Management in IoT based wireless sensor networks: an authentication protocol using symmetric key. *Int. J. Commun. Syst.* 32 (2019), 16. https://doi.org/10.1002/dac.4139
[21] Rajesh Gupta, Sudeep Tanwar, Sudhanshu Tyagi, and Neeraj Kumar. 2019. Tactile internet and its applications in 5G era: A comprehensive review. *Int. J. Commun. Syst.* 32 (2019), 14.
[22] Rajesh Gupta, Sudeep Tanwar, Sudhanshu Tyagi, and Neeraj Kumar. 2020. Machine learning models for secure data analytics: a taxonomy and threat model. *Comput. Commun.* 153 (2020), 406–440.
[23] Mahmood Ul Hassan, Shehzad Ashraf Chaudhry, and Azeem Irshad. 2020. An improved SIP authenticated key agreement based on Dongqing et al. *Wireless Pers. Commun.* 110, 4 (2020), 2087–2107.
[24] W. B. Hsieh, J. S. Leu, and A. Robust. 2014. User authentication scheme using dynamic identity in wireless sensor networks. *Wireless Pers. Commun.* 77, 2 (2014), 979–989.
[25] Sajid Hussain and Shehzad Ashraf Chaudhry. 2019. Comments on "biometrics-based privacy-preserving user authentication scheme for cloud-based industrial internet of things deployment." *IEEE IoT J.* 6, 6 (2019), 10936–10940.
[26] A. Irshad, S. A. Chaudhry, O. A. Alomari, K. Yahya, and N. Kumar. 2020. A novel pairing-free lightweight authentication protocol for mobile cloud computing framework. *IEEE Syst. J.* (2020), 1–9. https://doi.org/10.1109/JSYST.2020.2998721
[27] A. Irshad, M. Usman, S. A. Chaudhry, H. Naqvi, and M. Shafiq. 2020. A provably secure and efficient authenticated key agreement scheme for energy internet-based vehicle-to-grid technology framework. *IEEE Trans. Industr. Appl.* 56, 4 (2020), 4425–4435. https://doi.org/10.1109/TIA.2020.2966160
[28] U. Javaid, M. N. Aman, and B. Sikdar. 2020. A scalable protocol for driving trust management in internet of vehicles with blockchain. *IEEE IoT J.* 7, 12 (2020), 11815–11829. https://doi.org/10.1109/JIOT.2020.3002711
[29] Q. Jiang, S. Zeadally, J. Ma, and D. He. 2017. Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks. *IEEE Access* 5 (2017), 3376–3392.
[30] N. Khalil, M. R. Abid, D. Benhaddou, and M. Gerndt. 2014. Wireless sensors networks for internet of things. In *Proceedings of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP'14)*, 1–6.
[31] M. K. Khan and K. Alghathbar. 2010. Cryptanalysis and security improvements of a two-factor user authentication in wireless sensor networks. *Sensors* 10, 3 (2010), 2450–2459.
[32] H. H. Kilinc and A T. Yanik. 2014. survey of sip authentication and key agreement schemes. *IEEE Commun. Surv. Tutor.* 16, 2 (2014), 1005–1023.
[33] Zhihan Lv and Neeraj Kumar. 2020. Software defined solutions for sensors in 6G/IoE, computer communications. *Comput. Commun.* 153 (2020), 42–47.
[34] Khalid Mahmood, Jehangir Arshad, Shehzad Ashraf Chaudhry, and Saru Kumari. 2019. An enhanced anonymous identity-based key agreement protocol for smart grid advanced metering infrastructure. *Int. J. Commun. Syst.* 32 (2019), 16. https://doi.org/10.1002/dac.4137
[35] Khalid Mahmood, Xiong Li, Shehzad Ashraf Chaudhry, Husnain Naqvi, Saru Kumari, Arun Kumar Sangaiah, and Joel J. P. C. Rodrigues. 2018. Pairing based anonymous and secure key agreement protocol for smart grid edge computing infrastructure. *Fut. Gener. Comput. Syst.* 88 (2018), 491 – 500. https://doi.org/10.1016/j.future.2018.06.004

[36] Khwaja Mansoor, Anwar Ghani, Shehzad Ashraf Chaudhry, and Shahaboddin Shamshirband. 2019. Securing iot based rfid systems: A robust authentication protocol using symmetric cryptography. *Sensors* 19 (2019), 21. https://doi.org/10.3390/s19214752

[37] T. S. Messerges, E. A. Dabbish, and R. H. Sloan. 2002. Examining smart-card security under the threat of power analysis attacks. *IEEE Trans. Comput.* 51, 5 (2002), 541–552.

[38] Arzoo Miglani and Neeraj Kumar. 2019. Deep learning models for traffic flow prediction in autonomous vehicles: A review, solutions, and challenges. *Vehic. Commun.* 20 (2019).

[39] J. Ni, K. Zhang, X. Lin, and X. S. Shen. 2018. Securing fog computing for internet of things applications: challenges and solutions. *IEEE Commun. Surv. Tutor.* 20, 1 (2018), 601–628.

[40] Divya Prerna, Rajkumar Tekchandani, and Neeraj Kumar. 2020. Device-to-device content caching techniques in 5G: A taxonomy, solutions, and challenges. *Comput. Commun.* 153 (2020), 48–84.

[41] Sandeep Saharan, Seema Bawa, and Neeraj Kumar. 2020. Dynamic pricing techniques for intelligent transportation system in smart cities: a systematic review. *Comput. Commun.* 150 (2020), 603–625.

[42] W. Shi and P. Gong. 2013. A new user authentication protocol for wireless sensor networks using elliptic curves cryptography. *Int. J. Distrib. Sens. Netw.* 2013 (2013).

[43] Deepika Sirohi, Neeraj Kumar, and Prashant Singh Rana. 2020. Convolutional neural networks for 5G-enabled intelligent transportation system: A systematic review. *Comput. Commun.* 153 (2020), 459–498.

[44] Jangirala Srinivas, Ashok Kumar Das, Neeraj Kumar, and Joel J. P. C. Rodrigues. 2019. TCALAS: temporal credential-based anonymous lightweight authentication scheme for internet of drones environment. *IEEE Trans. Vehic. Technol.* 68, 7 (2019), 6903–6916.

[45] J. Srinivas, S. Mukhopadhyay, and D. Mishra. 2017. Secure and efficient user authentication scheme for multi-gateway wireless sensor networks. *Ad Hoc Netw.* 54 (2017), 147–169.

[46] M. Turkanovic, B. Brumen, and M. Holbl. 2014. A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks. *Ad Hoc Netw.* 20 (2014), 96–112.

[47] B. Vaidya, D. Makrakis, and H. T. Mouftah. 2010. Improved two-factor user authentication in wireless sensor networks. In *Proceedings of the 2nd International Workshop on Network Assurance and Security Services in Ubiquitous Environments*, 600–606.

[48] M. Wazid, A. K. Das, V. Bhat, and A. V. Vasilakos. 2020. LAM-CIoT: lightweight authentication mechanism in cloud-based IoT environment. *J. Netw. Comput. Appl.* 150, 10249 (2020), 6.

[49] M. Wolf and D. Serpanos. 2018. Safety and security in cyber-physical systems and internet-of-things systems. In *Proc. IEEE* 106, 1 (2018).

[50] F. Wu, L. Xu, S. Kumari, and X. Li. 2017. A privacy-preserving and provable user authentication scheme for wireless sensor networks based on internet of things security. *J. Ambient Intell. Human. Comput.* 8, 1 (2017), 101–116.

[51] H. L. Yeh, T. H. Chen, P. C. Liu, T. H. Kim, and H. W. Wei. 2011. A secured authentication protocol for wireless sensor networks using elliptic curves cryptography. *Sensors* 11, 5 (2011), 4767–4779.

Received June 2020; revised August 2020; accepted September 2020