Journal of Ambient Intelligence and Humanized Computing (2024) 15:3331–3345
https://doi.org/10.1007/s12652-024-04810-1

ORIGINAL RESEARCH

Lightweight and privacy-preserving device-to-device authentication to enable secure transitive communication in IoT-based smart healthcare systems

Sangjukta Das[1] · Maheshwari Prasad Singh[1] · Suyel Namasudra[2]

Received: 12 June 2023 / Accepted: 30 April 2024 / Published online: 21 June 2024
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2024

Abstract

Internet of Things (IoT) devices are often directly authenticated by the gateways within the network. In complex and large systems, IoT devices may be connected to the gateway through another device in the network. In such a scenario, new device should be authenticated with the gateway through the intermediate device. To address this issue, an authentication process is proposed in this paper for IoT-enabled healthcare systems. This approach performs a privacy-preserving mutual authentication between the gateway and an IoT device through intermediate devices, which are already authenticated by the gateway. The proposed approach relies on the session key established during gateway-intermediate device authentication. To emphasizes lightweight and efficient system, the proposed approach employs lightweight cryptographic operations, such as XOR, concatenation, and hash functions within IoT networks. This approach goes beyond the traditional device-to-device authentication, allowing authentication to propagate across multiple devices or nodes in the network. The proposed work establishes a secure session between an authorized device and a gateway, preventing unauthorized devices from accessing healthcare systems. The security of the protocol is validated through a thorough analysis using the AVISPA tool, and its performance is evaluated against existing schemes, demonstrating significantly lower communication and computation costs.

Keywords Untraceability · Key agreement · Anonymity · Integrity

1 Introduction

Technological advancements have brought about positive changes in all aspects of modern life, particularly in the field of Internet of Things (IoT)-enabled medical or healthcare systems. The benefits of it are being utilized through the use of wearables like smartwatches and wristbands, smartphones, and many more. Nowadays, people are using wearable gadgets to connect with online medical systems. It has provided convenience to the users, who can now receive medical advice without having to physically visit a medical facility or hospital (Sowjanya et al. 2021). However, it is crucial to protect users' information from illegitimate access and keep all exchanged messages between the user and server confidential. In any application, it is necessary to ensure the authenticity of user and server to provide security and privacy for both parties. Therefore, the reliability of these applications depends on authentication.

In a typical IoT-based network, user authentication involves three entities: gateway, smart device, and user. Based on the message exchange patterns among these actors, the existing authentication mechanisms in the literature can be categorize into many different categories (Krishnasrija et al. 2023). Out of those, one approach involves the gateway directly authenticating the user device, while the second approach involves the gateway authenticating the user device via a smart device. In this approach, the user communicates with the devices directly. For example, Farash et al. (2016) proposed a user authentication and key establishment model for an IoT environment. In this technique,

✉ Sangjukta Das
sangjukta24@gmail.com

[1] Department of Computer Science and Engineering, National Institute of Technology Patna, Bihar, India

[2] Department of Computer Science and Engineering, National Institute of Technology Agartala, Tripura, India

a user connects with a specific sensor device and reads its specific data without having to establish a connection with the gateway first and simply receive aggregated information. In another approach, the user authentication protocol does not allow the user direct access to the device. Instead, the device undergoes authentication by directly interacting with the gateway. For example, Shuai et al. (2019) proposed an authentication mechanism for smart homes. This system permits three different mutual authentications. The first is between the user and the gateway, the second is between the gateway and the smart device, and the third is between the user and the smart device. The majority of protocols in the above two categories focus only on user authentication. However, for large-scale IoT applications, there is a need for mechanisms that enable device-to-device (i.e. between the gateway and smart device) and device-to-gateway authentications (Zhou et al. 2019). These mechanisms facilitate mutual authentication between devices within an IoT network and enable the formation of an IoT device group with other devices.

Consider a IoT-based healthcare scenario, where multiple gateways and devices are interconnected. The gateway can authenticate the registered device directly, as shown in connection (A) of Fig. 1, or using another device that is directly linked to and authenticated by the gateway as shown in connection (B) of Fig. 1. Here, devices are divided into two categories: already authenticated devices and new devices trying to connect. This requires handling two types of requests for connection: device to gateway and device to device. To address this, the authentication process must support two separate schemes. The first scheme involves authentication for devices directly connecting to gateways. The second scheme enables authentication for devices connecting to trusted devices already authenticated by the gateway, ensuring secure and trusted communication.

By implementing these authentication schemes, the IoT network can effectively manage connections from both already authenticated devices and new devices, maintaining security and integrity within the network. The proposed mechanism, enabling mutual and transitive authentication in IoT networks, emphasizes the use of lightweight techniques to establish secure communication between IoT devices due to the computational limitations of these devices. The term "mutual" implies that both parties involved in the

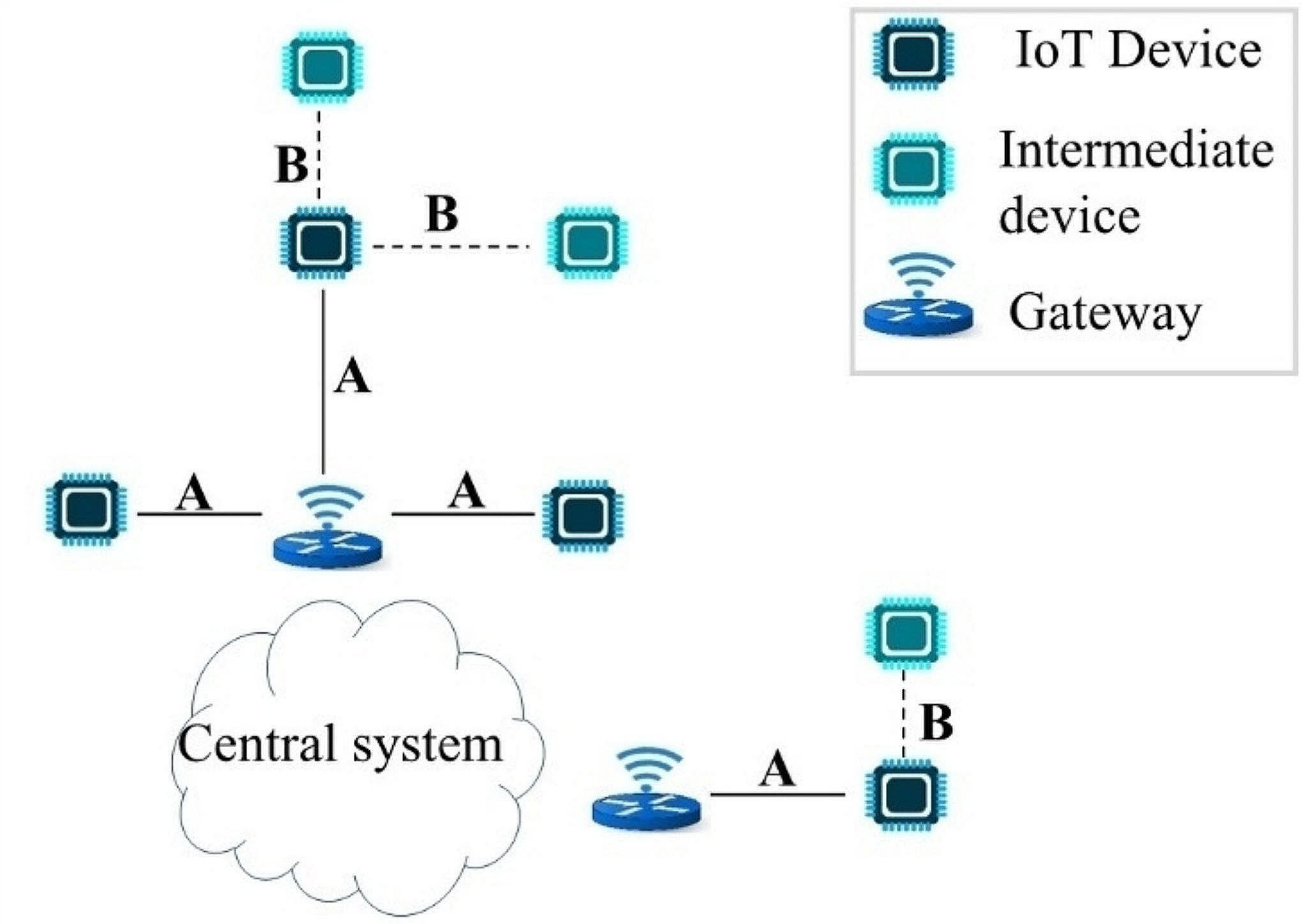


Fig. 1 Different types of devices in an IoT network

authentication process authenticate each other, ensuring trust and integrity. The term "transitive" suggests that the authentication mechanism extends beyond individual device-to-device authentication, allowing authentication to propagate across multiple devices or nodes in the network. Overall, the scheme focuses on lightweight and efficient authentication for IoT networks. During the registration phase, the necessary parameters and credentials for authentication are shared with the devices. These shared credentials are used for both authentication processes. The key contributions of this work are outlined as follows:

1) Development of two lightweight mutual authentication schemes: The work proposes novel authentication schemes for both device-to-gateway and device-to-device communication in IoT networks. These schemes are designed to be lightweight, ensuring efficient authentication while maintaining security.
2) Secure session establishment: The proposed schemes enable secure session establishment for registered devices. Successful authentication is a prerequisite for initiating a secure session, and each session utilizes a unique session key for data communication. This enhances the overall security and confidentiality of the communication.
3) Efficient credential management: The work introduces a credential updating process, ensuring the regular update of session IDs and parameters at the end of each session. This contributes to better credential management and helps maintain the integrity and freshness of authentication credentials.
4) Resistance to security threats: The proposed work is designed to withstand various security threats, including man-in-the-middle attacks, replay attacks, capturing, and eavesdropping. By incorporating security features, such as privacy preservation, untraceability, forward secrecy, credential updates, and session key secrecy, the proposed schemes enhance the overall security posture of the IoT-based smart healthcare system.

The remainder of the article is arranged in the following sections. Section 2 discusses some existing works that are related to the proposed scheme. In Sect. 3, the proposed authentication mechanism is introduced, by giving an overview of the design goals and system model of the proposed work. The construction of the proposed authentication mechanism is discussed in detail in Sect. 4. A detailed security analysis is carried out in Sect. 5, whereas in Sect. 6, the performance of the proposed scheme is compared with similar existing schemes. Finally, in Sect. 7, the work is concluded with future goals.

2 Related work

In literature, a few articles have recently covered the authentication and key agreement process because it is still a relatively new concept. In this section, some authentication and key agreement techniques which are presented in the past to ensure secure communication are discussed in brief.

Shuai et al. (2019) proposed an authentication mechanism based on Elliptic Curve Cryptography (ECC) for smart homes. This system permits three different mutual authentications. The first is between the user and the gateway, the second is between the gateway and the smart device, and the third is between the user and the smart device. Lastly, to facilitate secure interaction, a symmetric session key is also established between the user and the smart device. However, the performance of this approach is not satisfactory in terms of computational cost and communication expenses. In addition, several vulnerabilities, such as susceptibility to session key disclosure, replay attacks, and privileged insider attacks are also present in this scheme.

A user authentication scheme for a heterogeneous wireless sensor network was proposed by Turkanovic et al. (2014). This scheme offers secure authentication and generates a secure session key through the use of simple XOR and hash computation. However, this authentication model cannot provide mutual authentication between remote users and the gateway and cannot fulfill the requirements like anonymity and privacy in the network. Additionally, several security flaws were later revealed by Chang et al. (2016), including user impersonation and stolen smart card attacks.

Gope et al. (2016) proposed another lightweight authentication protocol to transfer real-time data in a wireless sensor network. This approach, however, asks for the gateway to save additional user data, which is impractical and often leads to failed authentication. This method does not support anonymity and untraceability because the sensor node's identity is revealed during data transmission over a public channel. Jolfaie et al. (2017) also revealed some security weaknesses in this protocol, specifically regarding the disclosure of the session key.

Li et al. (2017) designed another lightweight protocol for anonymous mutual authentication in WBAN. This protocol offers perfect backward and forward secrecy and is resilient to several attacks. However, upon further analysis, it is discovered that this protocol is also susceptible to intermediate node capture, sensor node impersonation, and hub node impersonation attacks. This protocol also suffers from a key-escrow problem.

Wazid et al. (2020) suggested a novel lightweight authentication approach for a cloud-based IoT system. The authors asserted that the scheme is effective, scalable, and capable of giving IoT sensors real-time data access via corresponding gateways which are placed in various clusters. However, while more gateway nodes are added, the system becomes unstable, thus, arises scalability issues. In this regard, Chaudhry et al. (2021) have found that the scheme cannot achieve mutual authentication between system entities in the situation of numerous registered users.

Jan et al. (2021) put forward a proposal for a lightweight mutual authentication and secure session generation scheme that relies on a client–server model. The IoT device acts as a client system and registers with the server anonymously in this scheme. Following this, both the server and client authenticate each other to establish a secure session for data transmission. During the registration and authentication phases, this scheme employs a lightweight symmetric encryption technique to exchange messages. However, the scheme is unable to handle the issue of server failure.

Izza et al. (2021) have proposed another user authentication scheme, where by using a trusted gateway device, the authentication starts between a user and a device. This technique makes advantage of straightforward symmetric cryptography to make the whole process lightweight. However, there are still potential vulnerabilities that could be exploited by an attacker. For example, the use of a single trusted gateway device to facilitate the authentication process introduces a potential single point of failure, and if the gateway device is compromised, an attacker could potentially impersonate the sensor device and gain unauthorized access to the network.

Banerjee et al. (2019) proposed a user authentication scheme suitable for symmetric key cryptosystem, where a user can connect to its preferred IoT sensor node. In this scheme, the gateway node serves as a third party and does not play any role once the user is connected to a sensor node. The key agreement protocol effectively negotiates a session key for authentication with the user's preferred sensor node, thus establishing a secure connection.

Masud et al. (2022) introduced an authentication scheme designed for IoT-enabled healthcare systems, featuring four distinct phases: device registration, user registration, mutual authentication between device and user, and key generation. This lightweight scheme is adept at establishing a secure session between the user and the device, thereby thwarting unauthorized access to data or resources. Nevertheless, it exhibits vulnerability to several types of attacks, including stolen verification attacks, privilege escalation attacks, and sensor node capture attacks.

Kumar et al. (2023) suggested a robust authentication mechanism that merges a password-protected biometric with physically unclonable functions (PUF). PUF device generates a distinct cryptographic key, which serves as the basis for authenticating the device, facilitating access to protected resources. This work addresses shortcomings observed in existing systems, particularly mitigating concerns related to key compromise impersonation attacks, wrong login, and server registration complexity.

Besides the above-discussed approaches, a variety of other approaches are proposed by many researchers in the literature for a wide range of application sectors, including mobile cloud environment (Gomaa et al. 2016; Gutub 2022), fog computing (Wang et al. 2022), healthcare sector (Khasim and Basha 2022), and vehicular technology (Namasudra and Sharma 2022; Wang et al. 2016). Yet, most of those techniques do not focus on the processing capability of resource-constraint devices (Moqurrab et al. 2022). Hence, novel schemes need to be designed that can match the characteristics of IoT and yet, offer a robust security mechanism (Khasim and Basha 2022; Das and Namasudra 2023; Chen 2022; Huang 2023; Kumar and Priyanka 2023; Ali et al. 2022; Gaur et al. 2023).

3 Overview of the proposed scheme

This study proposes a novel lightweight IoT device authentication approach for smart healthcare system. This section provides an overview of the proposed scheme, including design goals, system model, and network model of the proposed scheme.

3.1 System model

An IoT network can consist of several gateway devices, and there may be some IoT devices connected to it either directly or indirectly. The proposed approach involves two entities, namely the IoT devices and gateway devices that are defined below:

1) IoT Device: An IoTD is a limited-resourced device that gathers and sends a patient's health data to a gateway with which it is associated.
2) Gateway: A gateway is a device that connects various sensors, IoTD, and users to the internet and serves as a bridge between them. It acts as an intermediary between IoTD and the user, allowing them to communicate with each other. The gateway is not a resource-constraint device.

As shown in Fig. 1, IoT devices can be divided into two categories: devices directly connected to the gateway and device connected to the gateway via another device. The network needs to provide two different authentication and network connections; one is to connect a device directly to the gateway and the other is to connect a device to another device already authenticated by the gateway. A few assumptions that are considered in this work are mentioned below:

- It is assumed that the gateway is not a resource-constraint device while IoT device (IoTD) is a limited-resourced device.
- The IoT device gets registered with the system in an offline mode before the network gets operational.

3.2 Design goals

The objective of this study is to develop a new method for authenticating devices in smart healthcare environments with limited resources to access medical equipment. Therefore, the proposed authentication method for medical devices should ensure the following design goals:

1) Secure device authentication and session key establishment
2) Resistance to various cryptographic attacks
3) Minimizing storage, number of messages exchanged, and execution time.
4) Use of lightweight or simple cryptographic operations.

4 Construction of the proposed scheme

In this section, the major phases of the proposed scheme, namely registration, authentication, and credential update phase, are presented. Here, G_j and D_i represent gateway and IoT device, respectively, where $j = \{1, 2, \ldots, J\}$, $i = \{1, 2, \ldots, I\}$, and $I > J$. All the symbols used in this paper are described in Table 1.

Table 1 Descriptions of symbols

Symbols	Description
D_{ID},AID	Device identity and anonymous identity
R_{REQ}	Registration request
S_{REQ}	Session initiation request
$\oplus, \vert\vert, H()$	XOR, concatenation, and hash operation
n_1’ g_1’ m_1’g_2	Randomly chosen secret key values
T_Δ	Valid time range
R_{Device}	Registered device list
D_i and G_j	Device and gateway
T_c	Current time
SK	Session key
$SKgd$	Session key between the D_{new} and G_j
$SKdd$	Session key between the D_{new} and D_i
G_{ch}	Gateway challenge
D_{res}	Device response

4.1 Registration process

During the registration process, a secure channel is utilized for communication between the device and gateway. The steps of this procedure are sequentially discussed below:

Step 1: Each device D_i in the network contains a unique D_i's identity D_{ID}. The D_i generates a fresh Registration Request (R_{Req}) and sends it to a gateway G_j.

Step 2: After receiving the request, G_j retrieves D_{ID} from R_{Req} and selects a random x and a pre-shared key (PK_i) for D_i. Then, the G_j computes D_i's anonymous identity $AID = D_{ID} \oplus x$ and $S_{Hash} = h(AID||PK_i)$. After completing the registration process, the G_i stores D_i 's AID and PK_i in its registered device list (R_{Device}) and also sends these parameters to the D_i.

Step 3: After receiving AID and PK_i from G_i, D_i stores them in its secure memory.

4.2 Authentication process between a gateway and an immediate IoT device

The gateway can authenticate a registered device by utilizing its unique anonymous identity and pre-shared key. The first mutual authentication process is initiated by a device D_i which is directly connected to a G_j. The authentication procedure, outlined in Fig. 1, includes the following sequential steps:

Step 1: Initially, the D_i generates a number n_1 randomly, and using n_1, it calculates S_{Req} $=AID \oplus PK_i \oplus n_1$ and $S_{Hash} = H(AID||PK_i)$. Then, D_i sends the authentication request $\{S_{Req}$,S_{Hash}, $T_1\}$ at time T_1 to G_j through a public channel.

Step 2: The G_j searches for AID using S_{Hash} in its R_{Device} list. If the AID is not recorded in this R_{Device} list, then, the S_{Req} is declined by the G_j. Otherwise, G_j continues the authentication process. The G_j retrieves n_1 from S_{Req} as it contains D_i's AID, PK_i, and n_1. Further, G_j chooses a random nonce g_1 and calculates a challenge $G_{ch} = n_1 \oplus g_1 \oplus PK_i$. Finally, the G_j sends the $\{G_{ch}$,$T_2\}$ to the D_i at time T_2.

Step 3: On receiving $\{G_{ch}$,$T_2\}$, the D_i, at first, checks the validity of T_2 as $T_c - T_2 <= T_\Delta$. Then, the D_i generates g_1 from G_{ch} by computing $g_1 = G_{ch} \oplus n_1 \oplus PK_i$ and calculates response $D_{res} = g_1 \oplus n_1 \oplus g_2$ after choosing a random nonce g_1. The D_i sends $\{D_{res}$,$T_3\}$ to the G_j at time T_3. The D_i also computes the session key $SK = H(g_1||g_2)$.

Step 4: After receiving $\{D_{res}$,$T_3\}$ the G_j checks the validity of T_3 as $T_c - T_3 <= T_\Delta$. Then, the G_j computes $g_2 = D_{res} \oplus n_1 \oplus g_1$ to generate the session key $SK = H(g_1||g_2)$.

Thus, the mutual authentication of the device and the concerned gateway completes. Both the D_i and G_j can exchange data through a secure session using SK. Once data transmission completes, the session is terminated by discarding the current session key. At this stage, the anonymous identity and pre-shared key can be securely updated as AID_new and PK_i_new before terminating the ongoing session. The complete process is defined in the below subsection. Figure 2 shows the entire authentication technique discussed above along with the credential update phase.

4.3 Credential update phase

This section provides a process to update IoT device's credentials, such as anonymous identity and pre-shared key at the end of each session.

Step 1: The G_j selects fresh random nonce x_new and pre-shared key PK_i_new for D_i and computes $AID\_new = H(AID \oplus x\_new)$ and $S_{Hash} = H(AID\_new||PK_i\_new)$. Then, the G_j stores S_{Hash} and PK_i_new in its R_{Device} list and sends AID_new and PK_i_new to the device through the secure channel.

Step 2: On receiving $\{AID\_new, PK_i\_new\}$, the D_i updates its AID_new and PK_i_new. Then, the D_i ends the ongoing session by discarding the session key.

4.4 Authentication between a gateway and a new device through an intermediate device

As previously discussed, a new device can be connected to another authenticated device in addition to the gateway device. This authenticated device serves as an authentication facilitator for the new device. This authentication procedure is shown in Fig. 3 and also described using the below-mentioned steps:

Step 1: Initially, the new device D_{new} generates a number m_1 randomly. By using m_1, it calculates S_{Req} $=AID \oplus PK_j \oplus m_1$ and $S_{Hash} = h(AID||PK_j)$. Then, D_{new} sends the authentication request $\{S_{Req}, S_{Hash}, T_1\}$ at time T_1 to the authenticated device D_i through a public channel.

Step 2: After D_i receives $\{S_{Req}, S_{Hash}\}$, it adds its AID_{Di} and sends the message $\{S_{Req}, S_{Hash}, AID_{Di}\}$ to the concerned gateway device G_j. D_i sends the message through the secure channel that was established during the direct authentication process performed between D_i and G_j.

Step 3: Once the gateway receives D_{new}'s authentication request message from D_i, at first, it verifies the authenticated device identity AID_{Di}. Then, by following the same procedure as in the direct mutual authentication process the G_j computes challenge $G_{ch} = m_1 \oplus g_1 \oplus PK_j$ for D_{new}. The G_j also computes $D_{ch} = m_1 \oplus x \oplus n_1$ for D_i where x is a random nonce. Finally, the G_j sends the $\{G_{ch}, D_{ch}\}$ to the D_i through the same secure channel.

Step 4: Authenticated device D_i receives $\{G_{ch}, D_{ch}\}$ from G_j at time T_4 and checks if $(T_c - T_4) \leq T_\Delta$. D_i computes $D_{ch}\prime = D_{ch} \oplus n_1$, which provides $D_{ch}\prime = m_1 \oplus x$. Here, n_1 is known to the D_i. Then, the D_i selects another random nonce d^{old} to compute $D_{ch}^{new} = m_1 \oplus x \oplus d^{old}$. The D_i sends the message containing two challenges to generate secret session keys $\{G_{ch}, D_{ch}^{new}\}$ to the D_{new} through the public channel.

Step 5: At this stage, the D_{new} retrieves g_1 from G_{ch} by computing $g_1 = G_{ch} \oplus m_1 \oplus PK_j$. Once g_1 is retrieved, then, the D_{new} selects random nonce g_2 and calculates the session key $SKgd$ between G_j and D_{new} as $H(g_1||g_2)$ and a response $D_{res} = g_1 \oplus m_1 \oplus g_2$ for G_j. The D_{new} again selects random parameter d^{new} and calculates the session key SK_{dd} $=H(D_{ch}^{new} \oplus m_1 \oplus d^{new})$ $=H(x \oplus d^{old} \oplus d^{new})$ between D_i and D_{new}, and response for $D_i D_{res}^{new} = m_1 \oplus d^{new}$. Finally, D_{new} sends both D_{res} and D_{res}^{new} to the D_i.

Step 6: On receiving the response messages the D_i combines the D_{ch}^{new} and D_{res}^{new} to get the session key between D_i and D_{new} as $SK_{dd} = H(D_{ch}^{new} \oplus D_{res}^{new}) = H(x \oplus d^{old} \oplus d^{new})$ and the authentication ends. Next, the D_i sends $\{\mathrm{D_{res}}\}_{\mathrm{SK}}$ to the gateway.

Step 7: After the gateway G_j receives the response message D_{res} from D_i, it computes $g_2 = D_{res} \oplus m_1 \oplus g_1$. Then, it computes the session key $SKgd = H(g_1||g_2)$ between the D_i and G_j and the authentication ends here. Both the D_i and G_j can communicate with each other by using this key $SKgd$.

5 Security analysis

This section presents a security analysis of the proposed scheme to demonstrate its resilience and effectiveness. Initially, several potential threat scenarios are examined, and it is demonstrated that the proposed scheme is capable of withstanding these scenarios.

Theorem 1 A session can only be initiated by a registered device D_i with a concerned gateway G_j by sending a valid request S_{Req}.

Proof During the device registration phase, each valid device D_i is assigned an anonymous identity AID and a pre-shared key PK_i by the corresponding gateway G_j. G_j maintains a record of device identity (D_{ID}), AID, PK_i, and S_{Hash} in its R_{Device} list for every registered device. Suppose an intruder I_k tries to start a session by sending a S_{Req} to the G_j. G_j checks the sender's authenticity by

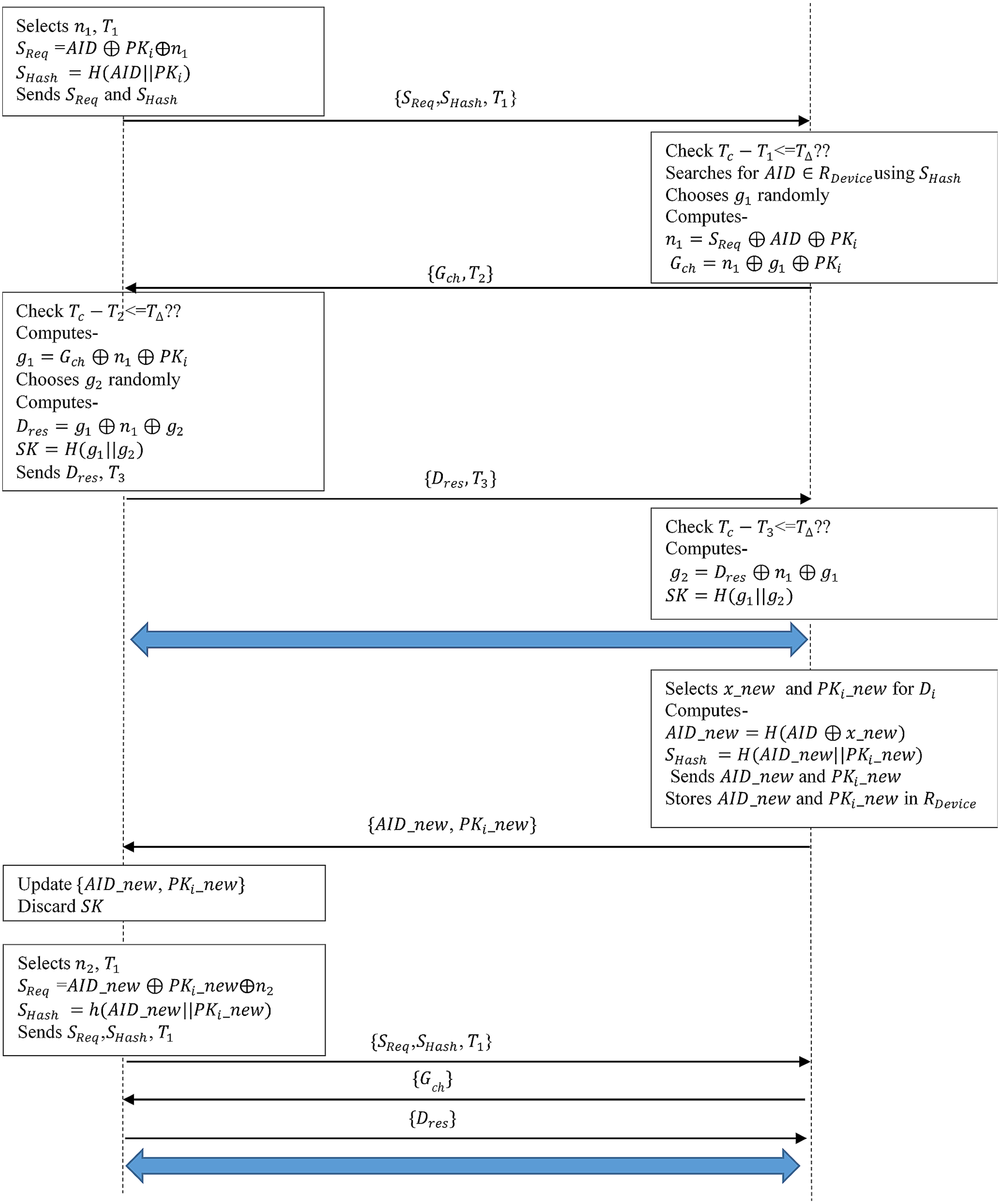

Selects n_1, T_1
$S_{Req} = AID \oplus PK_i \oplus n_1$
$S_{Hash} = H(AID||PK_i)$
Sends S_{Req} and S_{Hash}
$\{S_{Req}, S_{Hash}, T_1\}$
Check $T_c - T_1 <= T_\Delta$??
Searches for $AID \in R_{Device}$ using S_{Hash}
Chooses g_1 randomly
Computes-
$n_1 = S_{Req} \oplus AID \oplus PK_i$
$G_{ch} = n_1 \oplus g_1 \oplus PK_i$
$\{G_{ch}, T_2\}$
Check $T_c - T_2 <= T_\Delta$??
Computes-
$g_1 = G_{ch} \oplus n_1 \oplus PK_i$
Chooses g_2 randomly
Computes-
$D_{res} = g_1 \oplus n_1 \oplus g_2$
$SK = H(g_1||g_2)$
Sends D_{res}, T_3
$\{D_{res}, T_3\}$
Check $T_c - T_3 <= T_\Delta$??
Computes-
$g_2 = D_{res} \oplus n_1 \oplus g_1$
$SK = H(g_1||g_2)$
Selects x_new and PK_i_new for D_i
Computes-
$AID\_new = H(AID \oplus x\_new)$
$S_{Hash} = H(AID\_new||PK_i\_new)$
Sends AID_new and PK_i_new
Stores AID_new and PK_i_new in R_{Device}
$\{AID\_new, PK_i\_new\}$
Update $\{AID\_new, PK_i\_new\}$
Discard SK
Selects n_2, T_1
$S_{Req} = AID\_new \oplus PK_i\_new \oplus n_2$
$S_{Hash} = h(AID\_new||PK_i\_new)$
Sends S_{Req}, S_{Hash}, T_1
$\{S_{Req}, S_{Hash}, T_1\}$
$\{G_{ch}\}$
$\{D_{res}\}$

Fig. 2 The authentication between a gateway and directly connected IoT device and credential update phase

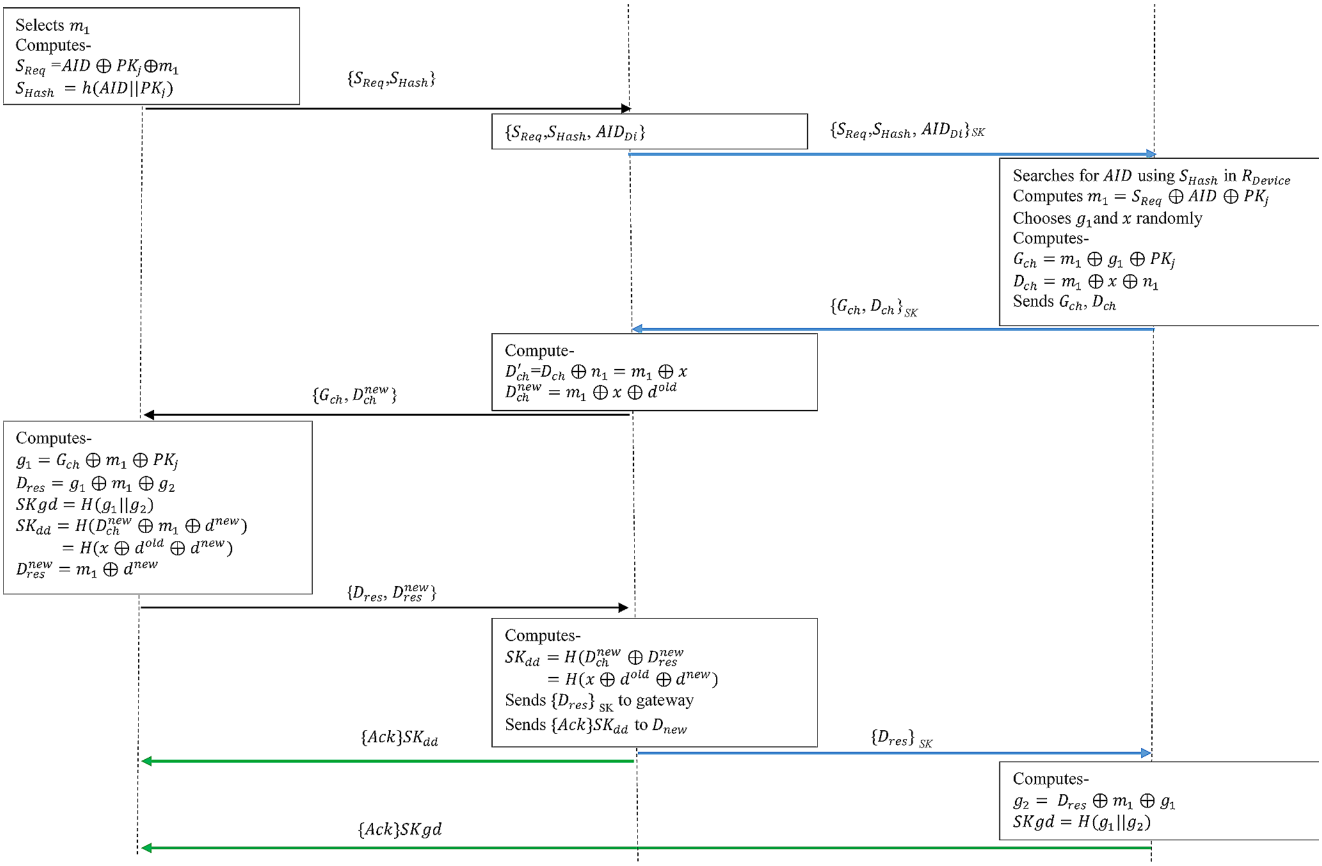

Selects m_1
Computes-
$S_{Req} = AID \oplus PK_j \oplus m_1$
$S_{Hash} = h(AID||PK_j)$
$\{S_{Req}, S_{Hash}\}$
$\{S_{Req}, S_{Hash}, AID_{Di}\}$
$\{S_{Req}, S_{Hash}, AID_{Di}\}_{SK}$
Searches for AID using S_{Hash} in R_{Device}
Computes $m_1 = S_{Req} \oplus AID \oplus PK_j$
Chooses g_1 and x randomly
Computes-
$G_{ch} = m_1 \oplus g_1 \oplus PK_j$
$D_{ch} = m_1 \oplus x \oplus n_1$
Sends G_{ch}, D_{ch}
$\{G_{ch}, D_{ch}\}_{SK}$
Compute-
$D'_{ch} = D_{ch} \oplus n_1 = m_1 \oplus x$
$D^{new}_{ch} = m_1 \oplus x \oplus d^{old}$
$\{G_{ch}, D^{new}_{ch}\}$
Computes-
$g_1 = G_{ch} \oplus m_1 \oplus PK_j$
$D_{res} = g_1 \oplus m_1 \oplus g_2$
$SKgd = H(g_1||g_2)$
$SK_{dd} = H(D^{new}_{ch} \oplus m_1 \oplus d^{new})$
$= H(x \oplus d^{old} \oplus d^{new})$
$D^{new}_{res} = m_1 \oplus d^{new}$
$\{D_{res}, D^{new}_{res}\}$
Computes-
$SK_{dd} = H(D^{new}_{ch} \oplus D^{new}_{res}$
$= H(x \oplus d^{old} \oplus d^{new})$
Sends $\{D_{res}\}_{SK}$ to gateway
Sends $\{Ack\}SK_{dd}$ to D_{new}
$\{Ack\}SK_{dd}$
$\{D_{res}\}_{SK}$
Computes-
$g_2 = D_{res} \oplus m_1 \oplus g_1$
$SKgd = H(g_1||g_2)$
$\{Ack\}SKgd$

Fig. 3 Authentication of a new device with a gateway through another authenticated device

checking S_{Hash} in its R_{Device} and also in R_{Update} list. If S_{Hash} is not found, it implies that AID is not registered with G_j, i.e., $AID \notin \{AID_1, AID_2, \ldots, AID_I\}$. In such a case, G_j rejects the S_{Req} and marks it as an intruder. Conversely, if a registered device D_i initiates a session by sending S_{Req}, the G_j verifies D_i's authenticity by checking S_{Hash} in R_{Device} and R_{Update} list. Therefore, a secure sessions with the gateway G_j can be initiated only by a registered device D_i. This completes the proof.

Theorem 2 Only the intended gateway G_j, and not any intruder I_k, can process device D_i's session initiation request S_{Req}.

Proof During the registration phase,D_i's registration details, including D_{ID}, AID, PK_i, and S_{Hash} are stored in G_j's R_{Device} list. Only D_i and the concerned G_j know these details. Suppose I_k receives the S_{Req} sent by the D_i. Since I_k does not know AID and PK_i, it is unable to get n_1 from S_{Req}. Hence, I_k cannot calculate a valid G_{ch} as $\{n_1 \oplus g_1 \oplus PK_i\}$ for D_i. Even if I_k manages to send G_{ch} to D_i, D_i can identify it due to the presence of the incorrect n_1 value. However, by using correct AID and PK_i the intended G_j can retrieve n_1 from S_{Req}. Therefore, only the correct G_j is able to process S_{Req} by calculating a valid G_{ch}. This concludes the proof.

Theorem 3 Only the designated entities D_i and G_j have the ability to decrypt the encrypted G_{ch} and D_{Res}, not any intruder I_k.

Proof To decrypt G_{ch} sent by G_j, any device D_i or intruder I_k needs to have correct PK_i and n_1. Since I_k does not know PK_i, it cannot get g_1 from G_{ch}. Therefore, I_k cannot compute the response message (D_{Res}) using the correct n_1 and g_1. Even if I_k attempts to guess the PK_i value, the probability of success is $\frac{1}{2^{128}}$. In contrast, D_i can decrypt G_{ch} with the correct PK_i and n_1 value. In the same way, I_k cannot decrypt D_{Res} sent by D_i. Therefore, only the designated D_i and G_j can decrypt the G_{ch} and D_{Res}. This concludes the proof.

5.1 Formal security analysis based on ROR model

In subsection, the security analysis of the proposed scheme is explained by employing the ROR model (Abdalla et al. 2005). The ROR model, recognized for its probabilistic nature, serves as a tool to validate the security of the session key implemented in the proposed scheme. Initially, the fundamentals of the ROR model, encompassing its definition and operational principles, are explored. Subsequently, the presentation explores the mathematical proofs in detail. The ROR model incorporates the following components.

The network involves three key entities: (1) new IoTD device, (2) intermediate node or device, and (3) gateway. It is assumed that D_{new}, D_i, and G_j, represent instances of new IoTD, intermediate node or device, and gateway, respectively, with the parameter ξ denoting the tuple $\{D_{new}$, D_i, $G_j\}$. These instances are referred to as oracles. As per the threat model, the adversary $\mathcal{A}$ holds complete control over the communication network. $\mathcal{A}$ is empowered to capture, eavesdrop, modify, delete, or reconstruct new messages within the network where legitimate devices communicate with each other. The adversary $\mathcal{A}$ can ascertain various queries to gain insight into the security of the protocol. Additionally, $\mathcal{A}$ can execute the following queries:

Exec (ξ): Through this query, $\mathcal{A}$ can conduct an eavesdropping attack, acquiring all the messages exchanged among the legitimate entities.

Send (ξ,M): $\mathcal{A}$ has the capability to send a message $\Updownarrow$ to ξ and receive a reply message $\Updownarrow'$ from ξ.

Capt (ξ): $\mathcal{A}$ can perform a capture attack by executing this query and capture all the parameters stored in the memories of the IoT device and gateway. Notably, $\mathcal{A}$ is constrained to a limited number of Capt() queries.

Hash (S): By executing this query, the adversary can obtain a fixed-length hash value from an input string S.

Tst (ξ): This query is designed to assess the semantic security of the session key. Before the experiment's commencement, adversary $\mathcal{A}$ tosses an unbiased coin c and only $\mathcal{A}$ is aware of the result, which dictates the outcome of the test query. If c = 1 and SK is fresh, the adversary obtains the correct SK from ξ. The adversary receives a random number if c = 0; otherwise, it returns a null value (⊥). Importantly, $\mathcal{A}$ can execute an unlimited number of Tst() queries.

Here, the event for $\mathcal{A}$ to win a game is considered as Scc, and the advantage of $\mathbb{A}$ to break the proposed scheme is denoted as $Adv^{\xi}_{\mathcal{A}} = |2Prob\,(Scc) - 1|$. The proposed scheme is secured if $Adv^{\xi}_{\mathcal{A}} \leq \varepsilon$ for sufficiently small $\varepsilon > 0$.

Semantic security of the session-key: Ensuring the semantic security of the session key within the ROR model entails the requirement that adversary $\mathcal{A}$ should be incapable of distinguishing between the genuine session key of an instance and a randomly generated key. $\mathcal{A}$ can conduct multiple *Tst*() queries to either IoT device, Intermediate Node, or gateway, and the output must consistently align with or be uniformly distributed for the random bit *c*. After the process, $\mathcal{A}$ provides a guessed bit *c*′ and succeeds if *c*′ = *c*. Denoting the event for $\mathcal{A}$ winning a game as *Scc*, the advantage of adversary $\mathcal{A}$ attempting to compromise the semantic security of the protocol is defined as $Adv^{\xi}_{\mathcal{A}} = |2Prob\,(Scc) - 1|$. The protocol is considered secure if $Adv^{\xi}_{\mathcal{A}} \leq \varepsilon$ where $\varepsilon > 0$ is a sufficiently small positive value, for the runtime *t*.

Theorem Let $\mathcal{A}$ be an adversary operating within polynomial time *t* against the proposed scheme in the random oracle model. The variables q_{snd}, |H*ash*|, q_{snd}, |*PD*|, and $Adv^{\xi}_{\mathcal{A}}$ represent the number of hash queries, the range space of hash, the number of send queries, the size of the uniformly distributed password dictionary, and $\mathcal{A}$'s advantage in breaking the proposed scheme in time *t*, respectively. The estimated advantage of *A* in deriving the session key between the IoTD Node and gateway is expressed as follows:

$$Adv^{\xi}_{\mathcal{A}} \leq \frac{q_{snd}}{|PD|} + \frac{q^2_{hash}}{|Hash|}$$

Proof G_i is a sequence of four games, where i ranges from 0 to 3. The success of each game G_i is denoted as $Scc(G_i)$ under the rules defined for each game.

G_0: $\mathcal{A}$ does not execute any query. Thus, the probability of $\mathcal{A}$ breaking the proposed scheme is $Adv^{\xi}_{\mathcal{A}}(n) = |2prob\left[Scc^{\xi}_{\mathcal{A}}(n) - 1\right]|$.

G_1: The game G_1 introduces $Exec(\xi)$ query to the first game G_0 to perform an eavesdropping attack. In this scenario, $\mathcal{A}$ can send all messages through the common channel and attempt to obtain the session key by executing $Tst\,(\xi)$ query. The adversary remains incapable of accessing the random secret values used in the session key. Consequently, the probability of success for G_1 is equivalent to that G_0, denoted as $Prob\left[Scc^{G_1}_{\mathcal{A}}\right] = Prob\left[Scc^{G_0}_{\mathcal{A}}\right]$.

G_2: $\mathcal{A}$ performs this game by sending $Send(\xi, \mathcal{M})$ and $Hash(S)$ queries over G_1 to mislead a legitimate device into accepting an illicit message. However, all messages incorporate current timestamps and random secret numbers, making it impractical to achieve hash collisions within polynomial time through the execution of send and hash queries. According birthday paradox, the following outcome is derived:

$$Prob\left[Scc^{G_2}_{\mathcal{A}}\right] - Prob\left[Scc^{G_1}_{\mathcal{A}}\right] | \leq \frac{q^2_{hash}}{2.\,|Hash|}$$

G_3: The adversary executes $Capt(\xi)$ query to gain access to all the confidential parameters stored in the memory of each node. $\mathcal{A}$ captures an intermediate node and gets all the information from it. However, since the intermediate node doesn't store any parameters of either the new IoTD device or gateway, no additional information is acquired. $\mathcal{A}$

captures new IoTD to get the parameters stored in it. D_{ID} , not being stored in the device's memory, $\mathcal{A}$ cannot use a node capture attack to obtain the true identity. Even if $\mathcal{A}$ has the parameter AID , the secret key value x prevents it from tracking the device. Therefore, the following can be derived

$$Prob\left[Scc_{\mathcal{A}}^{G_3}\right] - Prob\left[Scc_{\mathcal{A}}^{G_2}\right] | \leq \frac{q_{snd}}{2.\,|PD|}$$

$\mathcal{A}$ executes all the oracle queries to break the semantic security of the proposed protocol. A can win the game only after accurately guessing the c bit after executing $Tst()$ query. This gives $Prob\left[Scc_{\mathcal{A}}^{G_3}\right] = 1/2$ and from the above equations, it can be noted that

$$\begin{aligned} Adv_{\mathcal{A}}^{\xi} &= \left|2.Prob\left[Scc_{\mathcal{A}}^{G_0}\right] - 1\right| \\ &= \left|2.Prob\left[Scc_{\mathcal{A}}^{G_1}\right] - 1\right| \\ &= \left|2.Prob\left[Scc_{\mathcal{A}}^{G_1}\right] - 1/2\right| \\ &=2.\left|Prob\left[Scc_{\mathcal{A}}^{G_1}\right] - Prob\left[Scc_{\mathcal{A}}^{G_3}\right]\right| \end{aligned}$$

By using triangular inequality, it can be derived

$$\begin{aligned} \left|Prob\left[Scc_{\mathcal{A}}^{G_1}\right] - Prob\left[Scc_{\mathcal{A}}^{G_3}\right]\right| \leq \sum_{i=0}^{3} \Big|Prob\left[Scc_{\mathcal{A}}^{G_{i+1}}\right] \\ - Prob\left[Scc_{\mathcal{A}}^{G_i}\right]\Big| \end{aligned}$$

Table 2 Comparison of security features

Security features	Turkanović et al. (2014)	Fasash et al. (2016)	Banerjee et al. (2019)	Li et al. (2017)	Proposed scheme
Mutual authentication	√	√	√	√	√
Key agreement	√	√	√	√	√
Device anonymity	×	×	×	×	√
User anonymity	√	×	√	×	√
Traceability protection	×	×	×	×	√
Credential change	√	×	–	×	√
Resilience against					
Forward secrecy	√	×	×	√	√
MITM attack	√	√	√	√	√
IoTD impersonation attack	√	×	×	×	√
Evasdroping attack	×	√	–	√	√
Replay attack	√	√	×	√	√

$$\frac{1}{2}Adv_{\mathcal{A}}^{\xi} \leq \frac{q_{hash}^2}{2.\,|Hash|} + \frac{q_{snd}}{2.\,|PD|}$$

$$Adv_{\mathcal{A}}^{\xi} \leq \frac{q_{snd}}{|PD|} + \frac{q_{hash}^2}{|Hash|}\,(\text{multiplied by } 2)$$

5.2 Informal analysis

In this section, various potential security attacks, such as MITM, replay, eavesdropping, and impersonation against the proposed protocol are examined. The aim is to demonstrate that the devised scheme is resistant to such attacks, as described in the following subsections. In addition, this scheme also provides untraceability and anonymity features. A security feature comparison is given in Table 2.

1) MITM Attack: During a MITM attack, messages exchanged between two parties are intercepted by intruders who may modify them as they require. If the alterations are performed flawlessly, the parties communicating are not alerted to the changes. However, the proposed scheme requires the attacker to be aware of all secret parameters and pre-shared keys of the device to alter original messages sent from gateway or device. As attackers don't possess this knowledge, a MITM attack can be thwarted.
2) Replay Attack: In a replay attack, an unauthorized user resends previously captured secure information in an attempt to deceive the recipient. In a mutual authentication scheme, the gateway ignores a message if $|T_c - T_s$ $|> T_{\Delta}$, where T_{Δ} is the maximum transmission delay, preventing an attacker from intercepting and replaying communications. Similar to the first mutual authentication, if $|T_c - T_s| > T_{\Delta}$is true, the already authenticated device and gateway device also ignore the message. Assuming the adversary captured the messages and tried to impersonate a legitimate new device by replaying them, Dt would reject the request because of the invalid nonce in the replayed messages. In the same way, if the adversary impersonates valid Dt and replays the messages, the gateway would reject the request as it notices the invalid random numbers. Similarly, during all phases of authentication, additional checks are made to ensure the freshness of messages that contain the random nonce.
3) Identity Anonymity and Untraceability: An adversary shouldn't learn the device D_i 's true identity, D_{ID} , or be able to track D_i by eavesdropping on any communication channel. The D_{ID} in the proposed approach is concealed by utilizing a secret key value x, and each device is given an associated AID . D_{ID} not being

stored in the device's memory the adversary cannot use a node capture attack to obtain the true identity. Even if the attacker has the parameter AID, the secret key value x prevents it from tracking the device. Thus, the proposed technique maintains the features of anonymity and untraceability.

4) Eavesdropping Attack: Assuming that an adversary can access all the messages exchanged among the different entities. This means that adversary may have access to $S_{Req}, S_{Hash}, G_{ch},$ and D_{res}. However, adversary cannot get the session key or any other confidential information because the PK_i is not available to the adversary. As a result, the session key is protected against eavesdropping attack.

5) Known Session Key Secrecy: This security guarantee ensures that even if an attacker possesses knowledge of the session key used for message exchange, the communication remains secure. The proposed scheme utilizes a randomly generated session key $SKgd = H(g_1||g_2)$, where g_1 and g_2 are freshly generated at each new session. This approach ensures that the session key cannot be successfully predicted by an attacker, even if they have compromised a previous session key. Similarly, the session key between two devices is calculated as $SK_{dd} = H(D^{new}_{ch} \oplus D^{new}_{res}) = H(x \oplus d^{old} \oplus d^{new})$, where d^{old} and d^{new} are freshly generated for each new session. As a result, an attacker cannot access any sensitive data by predicting future session keys. Thus, the proposed scheme securely establishes the session keys with known session key secrecy.

5.3 Formal security analysis using AVISPA tool

This subsection provides a brief introduction to AVISPA, a widely used tool that evaluates the security of cryptographic protocols against known attacks and determines their safety status. AVISPA utilizes high-level protocol specification language (HLPSL) codes to specify any security model. The HLPSL2IF translator is employed to convert the HLPSL code into an intermediary form (IF), which is then processed by one of the four back-ends of the AVISPA tool. These back-ends include TA4SP, CL-AtSe, SATMC, and OFMC for security analysis. Overall, AVISPA is a powerful tool that can help researchers and practitioners evaluate the security of cryptographic protocols and identify potential vulnerabilities in their designs.

To analyze the HLPSL specification of the proposed protocol, the Security Protocol Animator (SPAN) of AVISPA simulation tool is utilized on an Ubuntu 10.10 (32-bit) operating system. The HLPSL specification of the proposed scheme outlines the roles of two entities, namely the device and gateway. The SUMMARY of both back-ends (CL-AtSe and OFMC) are obtained as output and shown in Fig. 4. The results demonstrate that this protocol is SAFE against a number of well-known attack, including impersonation, replay, and MITM attacks.

6 Performance analysis

The experimental evaluation of the proposed authentication methodology is presented in this section. The proposed methodology is implemented in Java1.8 on a Windows 11 computer with an 8th generation Intel Core i7 processor and 16 GB of RAM, along with analogous methods that have been identified in the literature. Memory utilization (storage cost), communication costs, and computation cost are some key IoT authentication metrics used in the analysis of experimental outcomes.

6.1 Storage requirement comparison

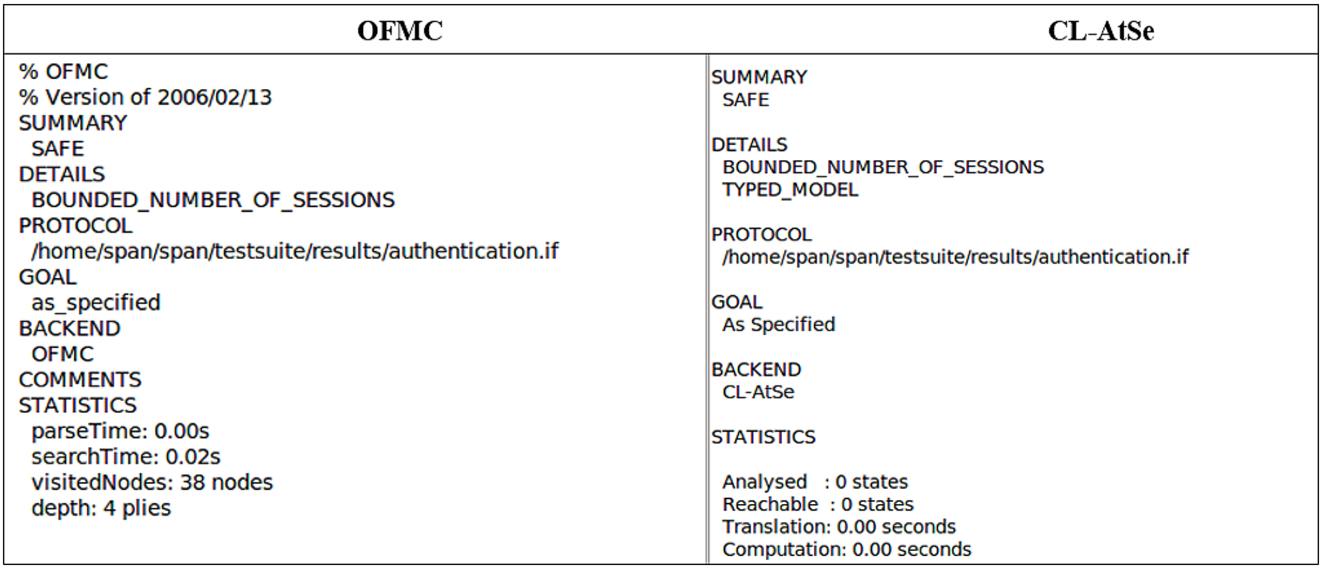
Table 3 compares the storage requirement of proposed scheme to that of the existing schemes (Li et al. 2017; Jan et al. 2021). AID and PK_i are stored in the memory of each device while a few parameters are stored in gateway and CA's memory in a similar manner. However, the storage requirement of the gateway and CA rises with the number of registered devices in the system. The size of the output of the hash function, identity parameters, and secret key values are 160, 128, and 128 bits, respectively. The overall storage cost of each entity in the suggested scheme is determined based on the size of these parameters.

6.2 Communication cost

Table 4 compares the communication cost of the proposed scheme with the existing schemes in terms of the number of messages and bits sent and received by the communicating entities. The proposed authentication process exchanges three messages which takes a total of 544 bits for these messages. As compared to other existing methods, the proposed scheme transfers noticeably less messages and bits.

6.3 Computation cost

To assess the computational cost of all existing authentication schemes, including the proposed one, XOR, concatenation (||), and hash operations are computed. Table 5 shows the number of operations used by both IoT device and the gateway in the authentication phase. H_T and X_T represent the time it takes to execute hash and XOR operations, respectively. It should be noted that the computational



OFMC

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/authentication.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.02s
  visitedNodes: 38 nodes
  depth: 4 plies
```

CL-AtSe

```
SUMMARY
 SAFE

DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL

PROTOCOL
 /home/span/span/testsuite/results/authentication.if

GOAL
 As Specified

BACKEND
 CL-AtSe

STATISTICS

 Analysed   : 0 states
 Reachable  : 0 states
 Translation: 0.00 seconds
 Computation: 0.00 seconds
```

Fig. 4 Output summary produced by the AVISPA tool's two back ends (OFMC and CL_AtSe)

Table 3 Storage requirement of the proposed scheme in bits

Scheme	Device	Gateway	CA
Li et al. (2017)	512b	512b	480b
Jan et al. (2021)	256b	256b	1792b
Proposed scheme	256b	416b	416b

Table 4 Communicational cost of the authentication process between gateway and device

Schemes	Total messages exchanged	Total bits
Li et al. (2017)	4	4672
Jan et al. (2021)	4	896
Proposed scheme	3	544

cost of a scheme is mostly caused by the cryptographic operations used. Therefore, for simplicity, it is possible to approximate the efficiency of a scheme by examining the execution time of these operations. Here, the execution time of hash operation, i.e., H_T is 32 ms and X_T is negligible as compared to H_T. As depicted in Table 5 the computation time for each protocol is calculated by considering the cryptographic operations and their execution time. Figure 5 illustrates the corresponding graphical representation.

Again the proposed indirect mutual authentication protocol is compared to other similar authentication protocols that are detailed in the literature to evaluate its effectiveness. From the perspective of message exchange, all the existing methods (Farash et al. 2016; Turkanović et al. 2014; Banerjee et al. 2019) are comparable to the proposed protocol because they use an IoT device linked to the gateway to authenticate the new device. To evaluate the proposed protocol, the methods discussed in Farash et al. (2016); Turkanović et al. 2014; Banerjee et al. 2019), are also implemented in this context. The experimental findings show that the proposed protocol significantly reduces computation time at the new IoT device. The experimental results are shown in Table 6 and Fig. 6.

7 Conclusion and future work

In any IoT-enabled healthcare system, and it is crucial to authenticate each entity before beginning a secure session. The anonymity and untraceability of each device must also be maintained. This article proposes an IoT-enabled healthcare system with a simple anonymous device authentication method. The existing session key established during authentication between the gateway and the immediate device is utilized by the authentication protocol for new devices.

Table 5 Computation cost of the authentication process between gateway and device

Scheme	IoTD	Gateway	Total
Jan et al. (2021)	$2H_T + 2X_T$	$2H_T + 2X_T$	$4H_T + 4X_T \approx 4H_T$
Li et al. (2017)	$3H_T + 7X_T$	$4H_T + 12X_T$	$7H_T + 19X_T \approx 7H_T$
Proposed scheme	$1H_T + 3X_T$	$1H_T + 3X_T$	$2H_T + 6X_T \approx 2H_T$

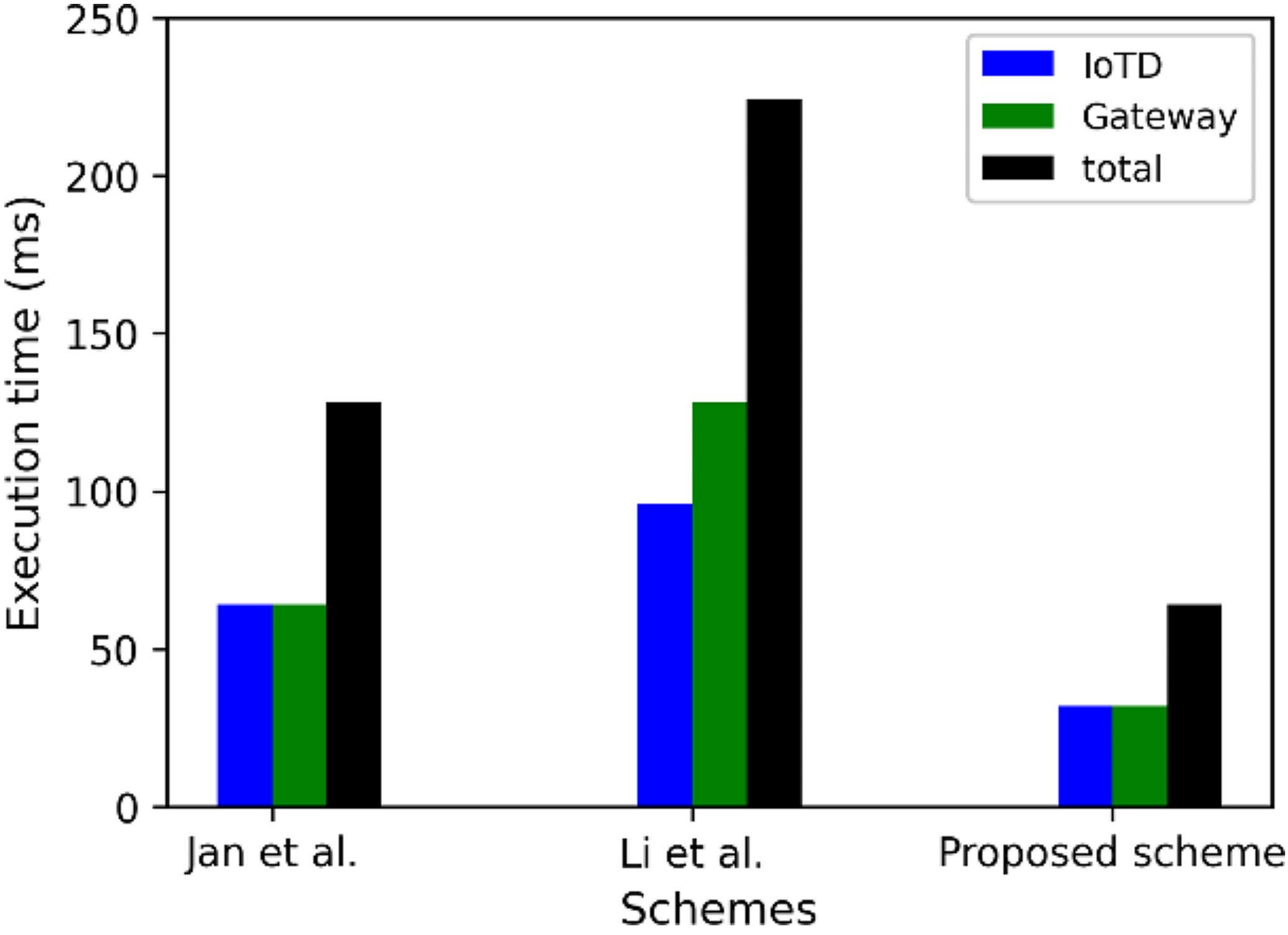


Fig. 5 Execution time taken by each scheme

Table 6 Computation cost of the authentication process between new device and gateway through already authenticated device

Scheme	New device	Auth_device	Gateway	Total
Turkanovic et al. (2014)	$7H_T$	$5H_T$	$7H_T$	19
Farash et al. (2016)	$11H_T$	$7H_T$	$14H_T$	32
Banerjee et al. (2019)	$8H_T$	$6H_T$	$8H_T$	22
Proposed scheme	$2H_T$	$2H_T$	$1H_T$	5

This means that the intermediate device simply facilitates the establishment of the session and securely transfers messages to the gateway. This method uses lightweight symmetric cryptographic operations, such as XOR, concatenation, and hashing, to perform mutual authentication between the device and gateway. Furthermore, the proposed authentication process exhibits a smaller message size compared to existing approaches, resulting in lower energy consumption. This makes it well-suited for IoT devices with limited resources. The lightweight nature of the process ensures efficient utilization of device capabilities, making it energy-efficient and suitable for resource-constrained IoT environments.

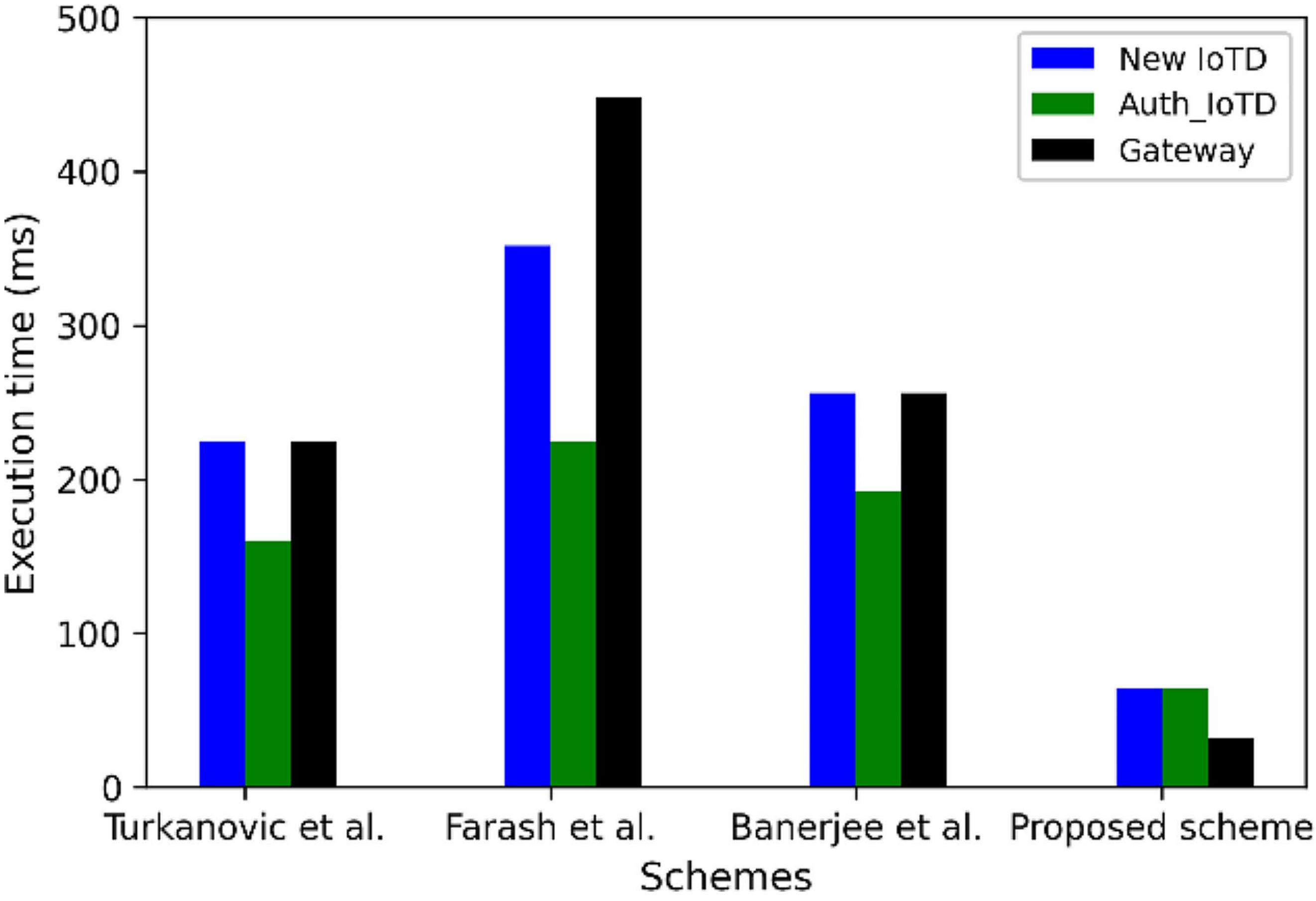


Fig. 6 Execution time taken by each scheme

References

Abdalla M, Fouque PA, Pointcheval D (2005) "Password-based authenticated key exchange in the three-party setting", in International Workshop on Public Key Cryptography. Springer, Berlin, pp 65–84

Ali HM et al (2022) Planning a secure and reliable IoT-enabled FOG-assisted computing infrastructure for healthcare. Clust Comput 25:2143–2161

Banerjee S, Chunka C, Sen S, Goswami RS (2019) An enhanced and secure biometric based user authentication scheme in wireless sensor networks using smart cards. Wireless Pers Commun 107(1):243–270

Chang C, Le H (2016) A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks. IEEE Trans Wireless Commun 15(1):357–366

Chaudhry SA, Irshad A, Yahya K, Kumar N, Alazab M, Zikria YB (2021) Rotating behind privacy: an improved lightweight authentication scheme for cloud-based IoT environment. ACM Trans Internet Technol 21(3):1–19

Chen Z (2022) Research on internet security situation awareness prediction technology based on improved RBF neural network algorithm. J Comput Cognit Eng 1(3):103–108

Das S, Namasudra S (2023) Lightweight and efficient privacy-preserving mutual authentication scheme to secure Internet of Things-based smart healthcare. Trans Emerg Telecommun Technol. https://doi.org/10.1002/ett.4716

Farash MS, Turkanovi´c M, Kumari S, Holbl M (2016) An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the internet of things environment. Ad Hoc Networks 36:152–176

Gaur VS, Sharma V, McAllister J (2023) Abusive adversarial agents and attack strategies in cyber-physical systems. CAAI Trans Intell Technol 8(1):149–165

Gomaa IA, Elrahman EA, Abid M (2016) Virtual identity approaches evaluation for anonymous communication in cloud environments. Int J Adv Comput Sci Appl 7(2):267–276

Gope P, Hwang T (2016) A realistic lightweight anonymous authentication protocol for securing real-time application data access in wireless sensor networks. IEEE Trans Industr Electron 63(11):7124–7132

Gutub A (2022) Boosting image watermarking authenticity spreading secrecy from counting-based secret-sharing. CAAI Trans Intell Technol. https://doi.org/10.1049/cit2.12093

Huang YK (2023) Design of a smart cabin lighting system based on internet of things. Cloud Comput Data Sci 4(2):112–121

Izza S, Benssalah M, Drouiche K (2021) An enhanced scalable and secure RFID authentication protocol for WBAN within an IoT environment. J Inform Secur Appl 58:1–15

Jan MA et al (2021) Lightweight mutual authentication and privacy-preservation scheme for intelligent wearable devices in industrial-CPS. IEEE Trans Industr Inf 17(8):5829–5839

Jolfaei AA, Talouki MA, Aghili SF (2017) Lightweight and anonymous three-factor authentication and access control scheme for real-time applications in wireless sensor networks. Peer Peer Netw Appl. https://doi.org/10.1007/s12083-017-0627-8

Khasim S, Basha SS (2022) An improved fast and secure CAMEL based authenticated key in smart health care system. Cloud Comput Data Sci 3(2):77–91

Krishnasrija R, Mandal AK, Cortesi A (2023) A lightweight mutual and transitive authentication mechanism for IoT network. Ad Hoc Networks 138:1–17

Kumar EP, Priyanka S (2023) A password less authentication protocol for multi-server environment using physical unclonable function. J Supercomput 79:21474–21506

Li X, Ibrahim MH, Kumari S, Sangaiah AK, Gupta V, Choo KKR (2017) Anonymous mutual authentication and key agreement scheme for wearable sensors in wireless body area networks. Comput Netw 129:429–443

Masud M, Gaba GS, Choudhary K, Hossain MS, Alhamid MF, Muhammad G (2022) Lightweight and anonymity-preserving user authentication scheme for IoT-based healthcare. IEEE Internet Things J 9(4):2649–2656

Moqurrab SA, Anjum A, Tariq N, Srivastava G (2022) Instant_Anonymity: a lightweight semantic privacy guarantee for 5g-enabled IIoT. IEEE Trans Industr Inf. https://doi.org/10.1109/TII.2022.3179536

Namasudra S, Sharma P (2022) Achieving a decentralized and secure cab sharing system using blockchain technology. IEEE Trans Intell Transp Syst. https://doi.org/10.1109/TITS.2022.3186361

Shuai M, Yu N, Wang H, Xiong L (2019) Anonymous authentication scheme for smart home environment with provable security. Comput Secur 86:132–146

Sowjanya K, Dasgupta M, Ray S (2021) A lightweight key management scheme for key-escrow-free ECC-based CP-ABE for IoT healthcare systems. J Syst Architect 117:1–10

Turkanović M, Brumen B, Hölbl M (2014) A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the internet of things notion. Ad Hoc Netw 20:96–112

Wang F, Xu Y, Zhang H, Zhang Y, Zhu L (2016) 2FLIP: A two-factor lightweight privacy-preserving authentication scheme for VANET. IEEE Trans Veh Technol 65(2):896–911

Wang H, Meng J, Du X, Cao T, Xie Y (2022) Lightweight and anonymous mutual authentication protocol for edge iot nodes with physical unclonable function. Secur Commun Netw 2022:1–11

Wazid M, Das AK, Bhat V, Vasilakos AV (2020) LAM-CIoT: Lightweight authentication mechanism in cloud-based IoT environment. J Netw Comput Appl 150:1–16

Zhou L, Li X, Yeh KH, Su C, Chiu W (2019) Lightweight IoT-based authentication scheme in cloud computing circumstance. Future Gener Comput Syst 91:244–251

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.