Cluster Computing
https://doi.org/10.1007/s10586-022-03733-w

SUACC-IoT: secure unified authentication and access control system based on capability for IoT

N. Sivaselvan[1,2] · K. Vivekananda Bhat[3] · Muttukrishnan Rajarajan[1] · Ashok Kumar Das[4] · Joel J. P. C. Rodrigues[5,6]

Received: 19 April 2021 / Revised: 2 July 2022 / Accepted: 4 July 2022

Abstract
With the widespread use of Internet of Things (IoT) in various applications and several security vulnerabilities reported in them, the security requirements have become an integral part of an IoT system. Authentication and access control are the two principal security requirements for ensuring authorized and restricted accesses to limited and essential resources in IoT. The built-in authentication mechanism in IoT devices is not reliable, because several security vulnerabilities are revealed in the firmware implementation of authentication protocols in IoT. On the other hand, the current authentication approaches for IoT that are not firmware are vulnerable to some security attacks prevalent in IoT. Moreover, the recent access control approaches for IoT have limitations in context-awareness, scalability, interoperability, and security. To mitigate these limitations, there is a need for a robust authentication and access control system to safeguard the rapidly growing number of IoT devices. Consequently, in this paper, we propose a new secure unified authentication and access control system for IoT, called SUACC-IoT. The proposed system is based around the notion of capability, where a capability is considered as a token containing the access rights for authorized entities in the network. In the proposed system, the capability token is used to ensure authorized and controlled access to limited resources in IoT. The system uses only lightweight Elliptic Curve Diffie-Hellman Ephemeral (ECDHE), symmetric key encryption/decryption, message authentication code and cryptographic hash primitives. SUACC-IoT is proved to be secure against probabilistic polynomial-time adversaries and various attacks prevalent in IoT. The experimental results demonstrate that the proposed protocol's maximum CPU usage is 29.35%, maximum memory usage is 2.79% and computational overhead is 744.5 ms which are quite acceptable. Additionally, in SUACC-IoT, a reasonable communication cost of 872 bits is incurred for the longest message exchanged.

Keywords Internet of Things (IoT) · Authentication · Access control · Capability · Security

1 Introduction

The recent years witnessed the widespread use of Internet of Things (IoT) paradigm in many applications such as smart home, smart healthcare, smart grid, smart transport, smart logistics, supply chain in industries, and so on. A study reveals the number of IoT devices worldwide would be more than 75 billion by the year 2025 [1]. Meanwhile, various security attacks are reported in the IoT devices of different applications [2, 3]. Thus, security in the field of IoT is an indispensable and crucial requirement. Authentication and access control are the two main security requirements to ensure authorized and restricted accesses to limited and pivotal resources in IoT. In an attempt to partially fulfill these requirements, some IoT device manufacturers made IoT device products with built-in authentication mechanism. However, several security vulnerabilities are disclosed in the firmware implementation of authentication in IoT such as weak, guessable, or hardcoded passwords leading to unauthorized access, insecure ecosystem interfaces resulting to lack of authentication/authorization or weak encryption (broken authentication), lack of firmware validation on device, insecure network services, insecure default settings that may allow the operators to modify the configurations, and so on [4]. Hence, the built-in authentication mechanism in IoT

Extended author information available on the last page of the article

Published online: 16 September 2022

devices is not reliable. On the other hand, the current authentication approaches for IoT [2, 3, 5–10] which are not firmware, are also vulnerable to certain security attacks among the prevalent ones namely Man-in-the-Middle (MITM), replay, traceability, session-key computation, secret disclosure, impersonation, gateway bypass, Denial-of-Service (DoS), and dictionary attacks in IoT. Moreover, some present access control approaches for IoT [11, 12] show limitations in terms of context-awareness, scalability, interoperability, and security. Therefore, there is a need for a robust authentication and access control system to safeguard the fast growing number of IoT devices.

In this paper, we propose a secure, unified authentication and access control system based on capability for IoT, called SUACC-IoT. The system is based on the concept of capability token which holds the access rights granted to the entity holding it. In the proposed system, the capability token is generated in the authentication stage. The generated token is used in mutual authentication and access control to ensure authorized and restricted access to limited resources in IoT. The system uses only lightweight Elliptic Curve Diffie-Hellman Ephemeral (ECDHE) performed using a highly performance optimized and fast elliptic curve, symmetric encryption/decryption, message authentication code and cryptographic hash primitives.

The proposed SUACC-IoT can be applied in a cloud-enabled IoT healthcare system, where the health-related information collected from smart IoT devices (wearable devices) is typically outsourced to the cloud in order to facilitate the timely sharing of health information with the healthcare service providers as well as the medical practitioners [13, 14]. The data security and privacy are very important in such environment as the health-related information is confidential and private. In addition, there are the challenges for handling the expensive computational time and energy consumption for the resource-limited IoT wearable devices that are deployed in the patient and doctor side in a smart healthcare system [15]. To handle these issues, we have applied the capability tokens that can be used in mutual authentication and access control in order to ensure authorized and restricted access to the limited resources in IoT-enabled healthcare system.

1.1 Research contributions

Various security vulnerabilities are reported in the built-in authentication mechanism in IoT devices such as weak, guessable, or hardcoded passwords, insecure ecosystem interfaces, lack of firmware validation on device, insecure network services, insecure default settings, and so forth [4]. Hence, the built-in mechanism is not reliable. For instance, recent IoT smart devices, such as fitness tracker and smartwatch often rely on the "Bluetooth Low Energy (BLE)" for transmission of the data. Wang et al. [16] designed the BlueDoor method that can obtain illegal information from the IoT smart devices via the BLE vulnerability. Michalevsky et al. [17] suggested some applications for the purpose of cryptographic secret handshakes among the mobile devices on the top of "Bluetooth Low-Energy (LE)". The present authentication approaches for IoT [2, 3, 5–10] that are not firmware are vulnerable to some security attacks among the prevalent ones viz., MITM, replay, traceability, session-key computation, secret disclosure, impersonation, gateway bypass, DoS, and dictionary in IoT. Some recent access control approaches for IoT [11, 12] show limitations in terms of context-awareness, scalability, interoperability, and security.

The following are the major contributions in this research work:

- We propose a secure unified authentication and access control system based on capability for IoT, called SUACC-IoT to address the limitations in the current authentication and access control approaches.
- We assess the security strength of the proposed protocol to computationally bounded probabilistic polynomial-time (PPT) adversaries using the universal Real-Or-Random (ROR) model [18].
- We carry out the security analysis of the proposed protocol for various attack vectors predominant in IoT namely MITM, replay, traceability, session key computation, secret disclosure, device impersonation, gateway impersonation, gateway bypass, offline dictionary, and DoS using the widely accepted Scyther automated software validation tool [19] and by intuitive reasoning.
- We also evaluate the proposed protocol for key performance parameters namely CPU usage, memory usage, computational overhead and communication cost in our IoT testbed involving Raspberry Pi.

1.2 Structure of the paper

The remainder of the paper is structured as follows: The closely related authentication and access control schemes are discussed in Sect. 2. The network model of the proposed protocol is presented in Sect. 3. The proposed system, SUACC-IoT, is discussed in detail in Sect. 4. The security strength of the proposed system to PPT adversaries and multiple attack vectors in IoT is demonstrated in Sect. 5. The performance/features of the proposed system are compared with the closely related existing schemes in Sect. 6. Section 7 concludes the paper.

2 Related work

In this section, the authentication and access control schemes that are closely related to our work are discussed.

A "user authentication scheme for multi-gateway Wireless Sensor Network (WSN)" was presented by Srinivas et al. [5]. Their scheme offers mutual authentication and key agreement. It is secure against MITM, replay, session key computation, traceability, device impersonation, gateway node impersonation, offline dictionary, and DoS attacks. It also supports anonymity. However, it is vulnerable to secret disclosure, and gateway bypass attacks. Also, it requires offline device registration with a system administrator which introduces some security threats.

Aman et al. [6] designed a "mutual authentication protocol for IoT using Physical Unclonable Functions (PUF)". It enables secret key establishment among the IoT devices. Their scheme is resistant to eavesdropping, replay, MITM, tampering, secret disclosure, and cloning attacks. However, it does not analyze anonymity, traceability, session key computation, gateway impersonation and bypass, offline dictionary, and DoS attacks.

Alotaibi [7] devised an "anonymous user authentication scheme for WSN". The scheme provides key agreement along with mutual authentication. It supports security features like user anonymity and resistance to replay, MITM, session key computation, user impersonation, gateway impersonation, DoS attacks. But, traceability, secret disclosure, gateway bypass, and offline dictionary attacks are not examined. Also, the scheme requires an additional hardware since biometric is used in user authentication.

Gope et al. [8] introduced a "privacy-preserving two-factor authentication scheme" for IoT devices. The scheme considers PUFs as one of the authentication factors. The scheme provides security features such as support for anonymity and resilience to replay, tampering, traceability, secret disclosure, cloning, impersonation attacks. But, MITM, session key computation, gateway impersonation and bypass, offline dictionary and DoS attacks are not investigated in the work.

A "lightweight and secure authentication scheme for IoT" was presented by Adeel et al. [9]. Their scheme provides mutual authentication and session key agreement. It is resistant to replay, MITM, session key computation, forgery, impersonation, and DoS attacks. However, it is vulnerable to traceability, secret disclosure, gateway impersonation and bypass, and offline dictionary attacks. Also, offline device registration with an authentication server is required in the scheme which poses security threats.

Aghili et al. [2] designed a "lightweight authentication, access control and access permissions transfer scheme for the e-health systems in IoT". It supports anonymity and exhibits resilience to MITM, replay, traceability, session key computation, impersonation, offline dictionary, and DoS attacks. But, it is not secure against secret disclosure, and gateway impersonation and bypass attacks. Their scheme's access permission transfer phase lacks scalability feature.

Feng et al. [20] pointed out that the serial computing mode is the primary concern for "slow decryption speed of the outsourced decryption" as well as the "parallel computing mode of outsourced decryption". To mitigate these issues, they designed an attribute-based encryption (ABE) model that relies on the parallel outsourced decryption for edge intelligent Internet of Vehicles (IoV) paradigm. Their scheme is suitable for all the ABE schemes with the tree access structures. Yin et al. [21] proposed a method for hybrid privacy preservation which is based on both the "functional encryption" and "Bayesian differential privacy" techniques. For the federated learning, they suggested a new function that can ensure that the server cannot extract the gradient parameters of each user's local training model as well as the weights of users' datasets. Moreover, they applied a local quantification mechanism for privacy loss in Bayesian differential privacy, that can permit the users to adapt the privacy budget based on the "data distribution of the datasets".

Bao et al. [22] suggested an intrusion-resilient server-aided attribute-based signature (ABS) scheme for an industrial IoT environment. In their approach, an adversary cannot forge a legitimate signature of the previous and future time period even if both the helper device and the server are compromised by the adversary.

Mohajer et al. [23] suggested a reputation based routing protocol that is based on CDS Connected Dominating Set (CDS) for mobile ad-hoc networks (MANETs). They also suggested a weight heuristic that can be applied to each node in MANET in order to choose CDS based on the use of the reputation value. It helps in achieving the selective forwarders' detection.

Kumar et al. [24] designed an energy efficient smart building architecture using the IoT technology. In their approach, the "Datagram Transport Layer Protection (DTLS)" and "Secure Hash Algorithm (SHA-256)" are integrated along with the optimizations from the "Certificate Authority (CA)" for improving security of their proposed architecture.

A "user authenticated key establishment protocol for smart home environment" was introduced by Wazid et al. [3]. Their scheme offers both mutual authentication and key agreement. The scheme provides support for anonymity and security against MITM, replay, traceability,

session key computation, user and device impersonation, gateway impersonation, gateway bypass, and offline dictionary attacks. But, it does not investigate secret disclosure and DoS attacks. The scheme assumes the gateway node is fully trusted and compromise of this node would compromise everything. Moreover, the scheme requires the devices to do offline registration with a registration authority which invites further security threats. Kim et al. [10] also designed an "authentication scheme based on lightweight signcryption protocol for IoT environment". It is resistant to MITM, replay, session key computation, and impersonation attacks. However, traceability, anonymity, secret disclosure, gateway, offline dictionary, and DoS attacks are not studied in the scheme.

Kurniawan and Kyas [25] proposed a trust-based access control mechanism that is based Bayesian decision theory for a large scale IoT environment. Their mechanism is applied for access control on uncertainty environment where the identities are not known in priori. Imani and Ghoreishi [26] suggested a framework that relies on the combination of the graphical model, the "Bayesian optimization", and the "mean objective cost of uncertainty (MOCU)". Their proposed framework satisfies scalability, fast decision making as well as efficiency.

Xu et al. [11] proposed a "capability-based access control framework for federated IoT environment". The framework considers two IoT domains. The framework is lightweight, context-aware, and fine grained. However, it is not scalable because there is only one coordinator who creates the capability token for all the devices in a particular IoT domain. Besides, interoperability and security are not examined in the framework. A attribute-based access control scheme using blockchain for IoT was presented by Yang et al. [12]. The scheme is context-aware, fine-grained, scalable, and secure. However, interoperability between the different parties in the system model is not examined.

Bao et al. [15] devised a "secure and lightweight fine-grained searchable data sharing for IoT-oriented and cloud-assisted smart healthcare system". The scheme realizes fine-grained access control and ciphertext search concurrently. It significantly reduces the computational time of IoT devices in the data user and patient side. The scheme's security is formally analyzed. Other authentication schemes in IoT-related environments have been also suggested in [27, 28].

The above discussion demonstrates that many authentication approaches in the literature are vulnerable to some security attacks among the prevalent ones namely MITM, replay, traceability, session-key computation, secret disclosure, impersonation, gateway bypass, DoS, and dictionary in IoT. The recent access control approaches have limitations in context-awareness, scalability, interoperability, and security. Thus, there is a need for a robust authentication and access control system that safeguards rapidly growing number of IoT devices. This has motivated us to design a secure unified authentication and access control system that fulfills the aforesaid criteria in this research work.

3 Network model

The proposed system's network model is depicted in Fig. 1. It presents an IoT environment where a device such as clinical smartphone would want to access a resource, for instance the file containing the readings of patient's biological parameter, in a healthcare device say glucometer or heart rate monitor or blood pressure monitor or spirometer or others to present the patient's health status to the

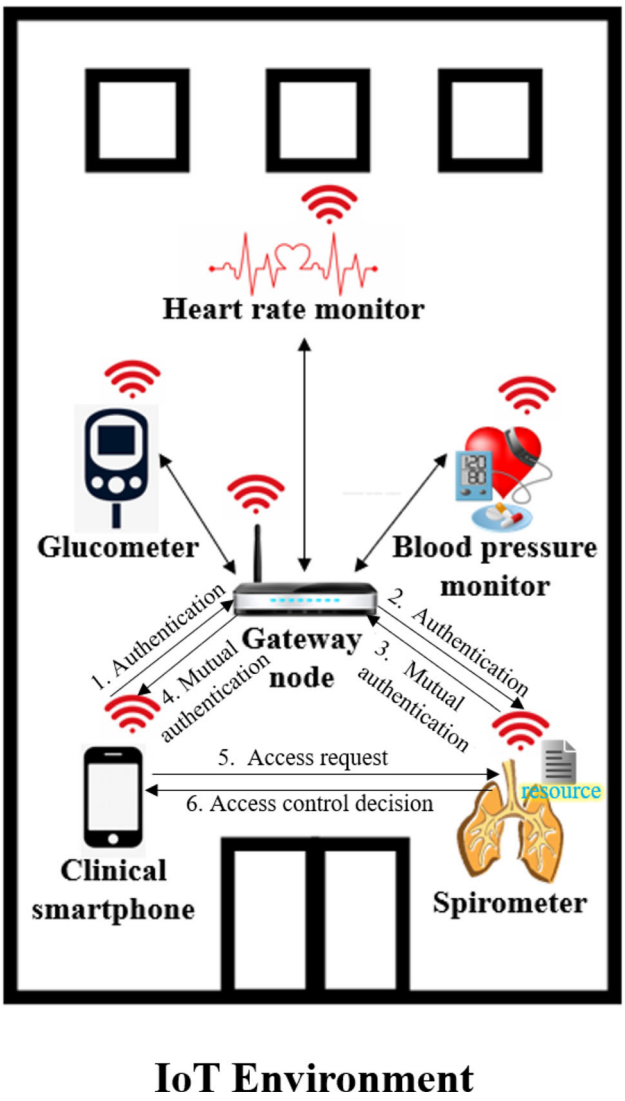

Fig. 1 Network model

physician for necessary action. The communication between the clinical smartphone and healthcare device happens via the gateway node. The gateway node acts as the protocol bridge to ensure protocol compatibility across the devices. Thus, the gateway node takes the responsibility of ensuring device interoperability.

As the major step to achieve the necessary security level, the two devices (clinical smartphone and healthcare device) and gateway node undergo mutual authentication. Two types of mutual authentication take place: 1) between the device and gateway node and 2) between the devices. The device (clinical smartphone) requests access to the resource in the healthcare device. The healthcare device then grants or denies access based on the access control rule.

4 Proposed SUACC-IoT system

In this section, the proposed authentication and access control system based on capability for IoT environment, SUACC-IoT, is presented. The system is based on the concept of capability token which holds the access rights granted to the entity holding it. In the proposed system, the capability token is generated in the authentication stage. The generated token is used in mutual authentication and access control to ensure authorized and restricted access to limited resources in IoT. The system uses only ECDHE performed using a highly performance optimized and fast elliptic curve, symmetric encryption/decryption, message authentication code and cryptographic hash primitives. The proposed protocol is split into three stages: (1) setup, (2) authentication, and (3) access control. Table 1 provides the description for the notations used in the protocol.

Table 1 Table of notations and their descriptions

Notation	Description
pr_k_{D1}, pu_k_{D1}	Public, private keys of $D1$
pr_k_{GWN}, pu_k_{GWN}	Public, private keys of GWN
pr_k_{D2}, pu_k_{D2}	Public, private keys of $D2$
s_1	Shared secret key of $D1$ and GWN
s_2	Shared secret key of GWN and $D2$
dk_1,dk_2	Keys derived from s_1,s_2
uid_{D1}, uid_{D2}	Universally unique identifiers of $D1$ and $D2$
η_1 to η_6	Random nonce values
$M1$ to $M20$	Computed messages
r	Resource requested by $D1$
id_{GWN}	Identifier of GWN
AR	Access rights assigned for $D1$
$ctxt$	Context awareness parameter assigned for $D1$
Cap_{D1}	Capability token of $D1$
Rnd	Cryptographically strong random number
sk	Session key of $D1$ and $D2$
r_A	Access requested by $D1$ regarding $'r'$
$E(\cdot)$	Symmetric-key encryption
$D(\cdot)$	Symmetric-key decryption
$MAC(\cdot)$	Message authentication code
$h(\cdot)$	One-way cryptographic hash
$\|$	Concatenation operator

4.1 Setup phase

In this stage, the device $D1$ and gateway node GWN agree upon a secret key dk_1 using lightweight ECDHE which they use in the authentication stage. Similarly, the device $D2$ and GWN agree upon a secret key dk_2. In the proposed system, GWN stores several secrets and sensitive information. Hence, GWN is equipped with tamper-proof device so that all the sensitive information stored in its secure databased is protected from the adversary.

- **Step 1.** The participating entities $D1$, GWN, and $D2$ generate their EllipticCurveCryptography (ECC) private and public keys $\{pr\_k_{D1}, pu\_k_{D1}\}$, $\{pr\_k_{GWN}, pu\_k_{GWN}\}$, and $\{pr\_k_{D2}, pu\_k_{D2}\}$ using a highly performance optimized, fast, and safe elliptic curve. These keys are ephemeral keys generated freshly in every key exchange. $D1$ and GWN exchange their public keys pu_k_{D1} and pu_k_{GWN}. In the same way, GWN and $D2$ exchange their public keys pu_k_{GWN} and pu_k_{D2}. This is the only instance in the proposed protocol, where the messages are exchanged through secure channels (established using the Transport Layer Security (TLS) protocol) to prevent possible MITM attacks. This is acceptable because the setup phase is only one-time process. However, the remaining messages in the proposed protocol are exchanged via open channels. In this manner, the proposed protocol makes very limited use of TLS.
- **Step 2.** $D1$ and GWN individually compute the shared secret key, s_1, as in $s_1 = pr\_k_{D1} \times pu\_k_{GWN}$ and $s_1 = pr\_k_{GWN} \times pu\_k_{D1}$. In a similar manner, GWN and $D2$ independently compute the shared secret key, s_2, using $s_2 = pr\_k_{GWN} \times pu\_k_{D2}$ and $s_2 = pr\_k_{D2} \times pu\_k_{GWN}$.
- **Step 3.** Subsequently, $D1$ and GWN independently derive another key, dk_1, from s_1 via $dk_1 = h(s_1, pu\_k_{D1}, pu\_k_{GWN})$. GWN and $D2$ individually derive key, dk_2, from s_2 through $dk_2 = h(s_2, pu\_k_{GWN}, pu\_k_{D2})$. The keys dk_1 and dk_2

are used in the authentication process of $D1$, GWN, and $D2$.

4.2 Authentication phase

In this stage, $D1$ and GWN, $D2$ and GWN, and eventually $D1$ and $D2$ are mutually authenticated. Besides, $D1$ and $D2$ establish a session key $'sk'$ at the end of authentication as outlined in Figs. 2 and 3. Consider $D1$ requests access to resource $'r'$ in $D2$.

- **Step 1.** $D1$ generates its universally unique identifier uid_{D1} and a random, one-time nonce η_1. Universally unique identifiers are used to identify the IoT devices, large in number, because they are unique, random and collision-resistant. $D1$ computes the messages $M1 = E(uid_{D1}\|r, dk_1)$ and $M2 = MAC(dk_1, M1\|\eta_1)$. $D1$, then, sends $M1, M2, \eta_1$ to GWN.
- **Step 2.** GWN computes the message $M3 = MAC(dk_1, M1\|\eta_1)$ and checks if $M3 == M2$. This verification ensures that $M1, M2, \eta_1$ is not replayed and the sender, $D1$, is a legitimate holder of the key dk_1. If the verification is successful, $D1$ is authenticated to GWN. GWN generates its identifier id_{GWN} and η_2. Next, GWN computes the messages $M4 = D(M1, dk_1), M5 = E(M4, dk_2)$, and $M6 = MAC(dk_2, M5\|\eta_2)$ and sends $M5, M6, \eta_2$ to $D2$.

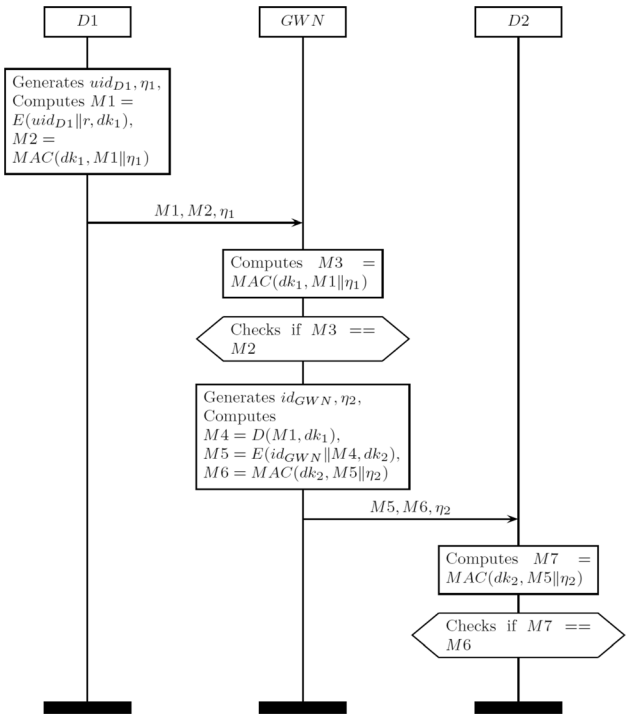

Fig. 2 Summary of authentication phase 1

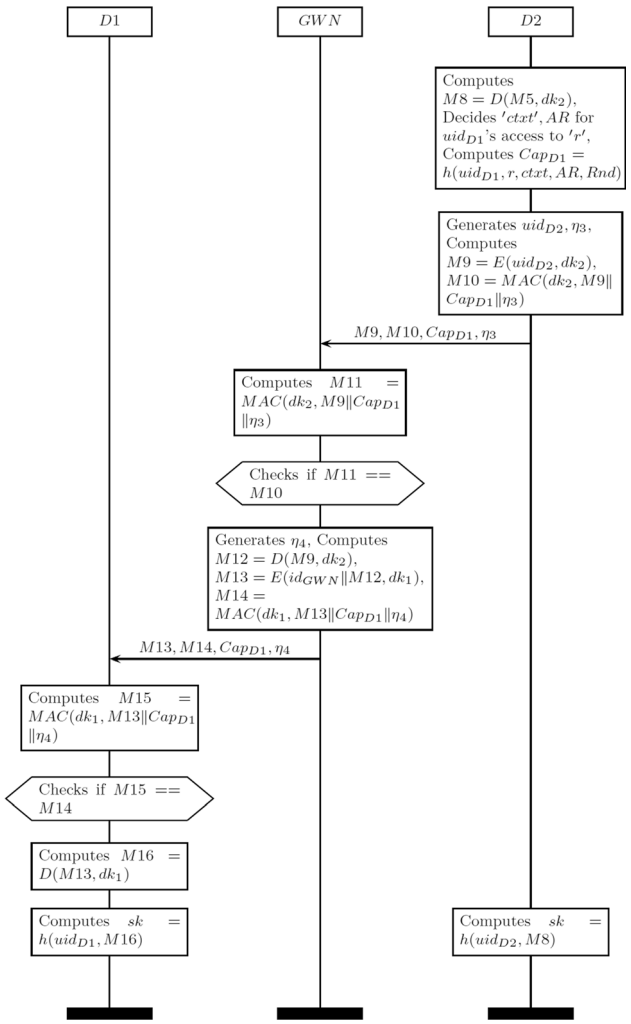

Fig. 3 Summary of authentication phase 2 (mutual authentication)

- **Step 3.** $D2$ computes the message $M7 = MAC(dk_2, M5\|\eta_2)$ and checks if $M7 == M6$. If the verification is successful, GWN is authenticated to $D2$.
- **Step 4.** $D2$ computes the message $M8 = D(M5, dk_2)$. Consequently, $D2$ obtains id_{GWN}, uid_{D1}, and $'r'$. $D2$ decides the context $'ctxt'$ and access rights AR for uid_{D1}'s access to its resource $'r'$. $ctxt \in \{null, \{ctxt1, ctxt2\}\}$ where $'ctxt1'$ and $'ctxt2'$ are context-awareness information like time, location and $AR \in \{null, read, write, \{read, write\}\}$. Then, $D2$ computes the capability token of $D1$ as in $Cap_{D1} = h(uid_{D1}, r, ctxt, AR, Rnd)$.
- **Step 5.** $D2$ generates its universally unique identifier uid_{D2} and η_3. Thereafter, it computes the messages $M9 = E(uid_{D2,dk_2})$ and $M10 = MAC(dk_2, M9\|Cap_{D1}\|\eta_3)$. $D2$ sends $M9, M10, Cap_{D1}, \eta_3$ to GWN.
- **Step 6.** GWN computes the message $M11 = MAC(dk_2, M9\|Cap_{D1}\|\eta_3)$ and verifies if

$M11 == M10$. If the verification is successful, $D2$ is authenticated to GWN. GWN generates η_4 and computes the messages $M12 = D(M9, dk_2), M13 = E(id_{GWN} \| M12, dk_1)$, and $M14 = MAC(dk_1, M13 \| Cap_{D1} \| \eta_4)$. It sends $M13, M14, Cap_{D1}, \eta_4$ to $D1$.

- **Step 7.** $D1$ computes the message $M15 = MAC(dk_1, M13 \| Cap_{D1} \| \eta_4)$ and checks if $M15 == M14$. If this verification succeeds, GWN is authenticated to $D1$. Thus, the mutual authentication of $D1$ and GWN, $D2$ and GWN and $D1$ and $D2$ is achieved. $D1$ computes $M16 = D(M13, dk_1)$ to get id_{GWN}, uid_{D2}. Lastly, $D1$ and $D2$ independently compute the session key sk using $sk = h(uid_{D1}, M16)$ and $sk = h(uid_{D2}, M8)$ respectively. The session key so established is used for secure communication in the access control process.

4.3 Access control phase

The capability token Cap_{D1} generated in the authentication process is used to control the access of $D1$ to resource $'r'$ of $D2$ at this instance. Figure 4 presents the steps in the access control process. The sequence charts in this section are drawn using the msc package [29].

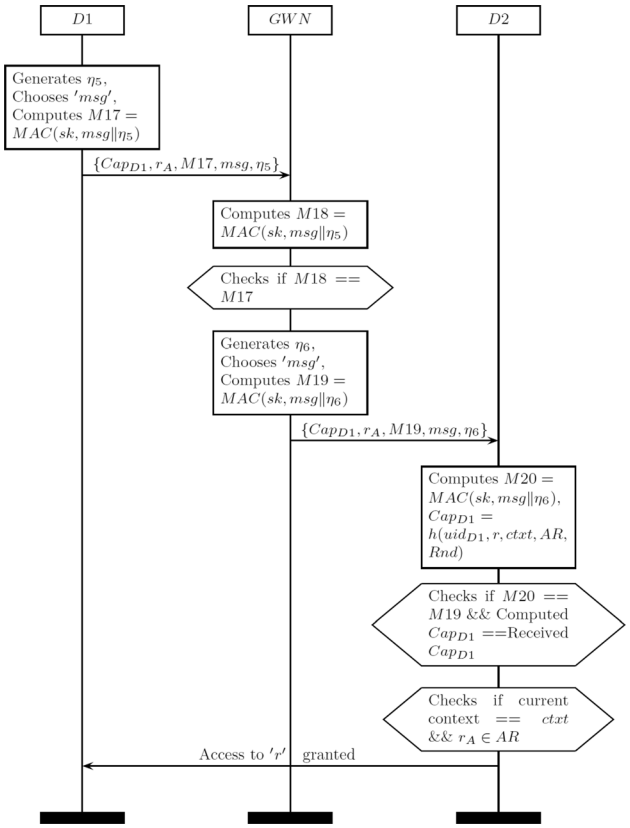

Fig. 4 Summary of access control phase

- **Step 1.** $D1$ sends GWN, the generated η_5, randomly chosen message $'msg'$ and computed message $M17 = MAC(sk, msg \| \eta_5)$ along with Cap_{D1}, r_A to prevent possible replay attack in this communication.
- **Step 2.** GWN computes the message $M18 = MAC(sk, msg \| \eta_5)$ and checks if $M18 == M17$. If this verification succeeds, GWN generates η_6, randomly chooses $'msg'$ and computes the message $M19 = MAC(sk, msg \| \eta_6)$. It then sends $Cap_{D1}, r_A, M19, msg, \eta_6$ to $D2$.
- **Step 3.** $D2$ computes the message $M20 = MAC(sk, msg \| \eta_6)$ and $Cap_{D1} = h(uid_{D1}, r, ctxt, AR, Rnd)$. It checks if $M20 = = M19$ and and computed $Cap_{D1} ==$ received Cap_{D1}. If the verification is successful, $D2$ checks if the current context $== ctxt$ and and the access requested $r_A \in AR$. If this verification also succeeds, the requested access to $'r'$ is granted to $D1$.

5 Security analysis

In this section, we rigorously analyze the SUACC-IoT system in terms of its security. We perform the formal security analysis using the widely-recognized Real-Or-Random (ROR) random oracle model [18] and the formal security verification using the broadly-accepted automated software validation tool, known as the Scyther tool [19]. We also carry out security analysis by intuitive reasoning through the non-mathematical (heuristic) approaches. Wang et al. [30] in their seminal work stated that the widely-used formal security methods, such as the "random oracle model" and "Burrows–Abadi–Needham (BAN) logic" [31] can not always capture some structural mistakes in the analyzed authentication protocols, and thus, ensuring the soundness of authentication protocols still remains an open problem. Due to this, we need to analyze the proposed protocol using all the possible security methods (formal analysis, formal security verification and informal analysis) to show that it is robust against various potential attacks with high probability.

5.1 Formal security analysis using ROR model

The security strength of the proposed protocol to computationally bounded PPT adversaries is evaluated in this section using the universal ROR model.

5.1.1 ROR model

In the proposed protocol, there are three participants namely $D1$, GWN, and $D2$.

- **Instances:** Let $\Pi_{D1}^{w_1}$, $\Pi_{GWN}^{w_2}$, and $\Pi_{D2}^{w_3}$ be the instances of $D1$, GWN, and $D2$ respectively. w_1, w_2, and w_3 are called oracles.
- **Accepted state:** An instance Π^w is known to be accepted, if it goes into an accepted state on receiving the last protocol message. All the messages sent and received by Π^w, concatenated in order, is the session-identification (s-id) of Π^w for the ongoing session.
- **Partnering:** Two instances Π^{w_1} and Π^{w_2} are said to be partnered, if they meet the following three conditions simultaneously: (i) Π^{w_1} and Π^{w_2} are in accepted state, (ii) Π^{w_1} and Π^{w_2} undergo mutual authentication and share an identical s-id, and (iii) Π^{w_1} and Π^{w_2} are mutual partners.
- **Instance freshness:** The instance $\Pi_{D1}^{w_1}$ or $\Pi_{D2}^{w_3}$ is considered fresh, if the session key between $D1$ and $D2$ is not revealed to $\mathcal{A}$ via $Reveal(\Pi^w)$ query.
- **Adversary:** The adversary $\mathcal{A}$ is a PPT Turing machine which has the ability to read, modify, intercept, delay, delete the protocol messages, fabricate new messages and inject them into the network. In addition, it can ask an instance to reveal the session key. These abilities of $\mathcal{A}$ are modeled using a predetermined set of oracles. These oracles are accessible to $\mathcal{A}$ and all the participants. They are:
  - *Enc*: This oracle represents the symmetric key encryption $E(\cdot)$ of the proposed protocol.
  - *Gen*: This oracle corresponds to the code generation part of message authentication code $MAC(\cdot)$ of the proposed protocol.
  - *Ver*: This oracle represents the code verification part of message authentication code $MAC(\cdot)$ of the proposed protocol.
  - *h*: This oracle corresponds to the cryptographic hash function $h(\cdot)$ of the proposed protocol.

  In addition, $\mathcal{A}$ has access to the following queries:
  - $Execute(\Pi_{D1}^{w_1}, \Pi_{GWN}^{w_2} and \Pi_{D2}^{w_3})$: This query represents a passive eavesdropping attack on the protocol messages. $\mathcal{A}$ runs this query to acquire the messages exchanged between $D1$, GWN and $D2$.
  - $Send(\Pi_E^w, message)$: The *Send* query models active attacks on the protocol. It sends *message* to an instance Π_E^w. On receiving *message*, Π_E^w advances as per the specifications of the protocol. Any message generated by Π_E^w is regarded as the output and given to $\mathcal{A}$.
  - $Reveal(\Pi_E^w)$: An instance Π_E^w, on receiving this query, reveals the session key that it has established with its partner to $\mathcal{A}$.
  - $TestAKE(\Pi_E^w)$: This query represents the indistinguishability-based semantic security of the session key $'sk'$ between Π_E^w ($D1$) and its partner ($D2$). The *TestAKE* oracle chooses value for the random bit b_r. If $b_r = 1$, the actual session key is returned as response to the query, Otherwise, a random key chosen from the session key sample space is returned.

5.1.2 Cryptographic preliminaries

The following cryptographic preliminaries are used in the security proof in the subsequent section.

(1) Elliptic Curve Computational Diffie Hellman Problem (*ECCDHP*): Consider G be an elliptic curve of prime order $'q'$ and Q be a base point on the elliptic curve G. Let $'u'$ and $'v'$ be the private keys of the two communicating parties chosen randomly from Z_q^*, where Z_q^* is the set of integers over $'q'$. The *ECCDHP* for G, when given two elements $(uQ, vQ) \in G^2$, is to compute the shared secret key viz., $uvQ \in G$. The advantage of adversary $\mathcal{A}$ to find the solution to the *ECCDHP* for G is given by $Adv_G^{ECCDHP}(\mathcal{A}) = P[\mathcal{A}(G, Q, uQ, vQ) = uvQ]$. The *ECCDHP* assumption holds in G if for all the PPT adversaries, $Adv_G^{ECCDHP}(\mathcal{A})$ is negligible.

(2) Hash collision (*HASH*): Consider $h : \{0,1\}^* \rightarrow \{0,1\}^l$ be a one-way cryptographic hash function which inputs a string of random length, say, $x \in \{0,1\}^*$ and outputs a fixed length $l - bit$ hash, say, $h(x) \in \{0,1\}^l$. Let $'x1'$ and $'x2'$ be strings of random length randomly chosen. The advantage of $\mathcal{A}$ in finding *HASH* for $h(\cdot)$ is given by $Adv_{h(\cdot)}^{HASH}(\mathcal{A}) = P[(x1, x2) \leftarrow \mathcal{A} : x1 \neq x2 \text{ and } h(x1) = h(x2)]$. $h(\cdot)$ is collision-resistant if for all the PPT adversaries, $Adv_{h(\cdot)}^{HASH}(\mathcal{A})$ is negligible.

(3) Indistinguishability of encryption under Chosen Plaintext Attack ($IND - CPA$): Let Ω denote a symmetric key encryption scheme. Let the encryption key be ek_1. Let $'p_i'$ denote a plaintext input and ϕ indicate the choice of choosing 0 or 1 for $'i'$. The advantage of $\mathcal{A}$ in carrying out $IND - CPA$, as a single eavesdropper, is given by

$$Adv_{\Omega}^{IND-CPA}(\mathcal{A}) = |2.P[\mathcal{A} \leftarrow Enc_{ek_1}; (p_0, p_1) \leftarrow \mathcal{A}; \phi \leftarrow \{0,1\}; \beta \leftarrow Enc_{ek_1}(p_\phi) : \mathcal{A}(\beta) = \phi] - 1|$$

. The scheme Ω is $IND - CPA$ secure if for all the PPT adversaries, $Adv_{\Omega}^{IND-CPA}(\mathcal{A})$ is negligible.

(4) Existential Unforgeability under Chosen Plaintext Attack ($EU - CPA$): Let Δ denote a MessageAuthenticationCode (MAC) scheme. Δ involves

the message authentication code generation and verification processes. *Gen* oracle inputs a $l-bit$ key $'k'$ and a message $'msg'$, and generates a code δ. *Ver* oracle inputs key $'k'$, message $'msg'$ and code δ, and verifies if δ is the correct code for $'msg'$ under $'k'$. If yes, it outputs 1 else outputs 0. Let the advantage of $\mathcal{A}$ in performing $EU-CPA$ on Δ be $Adv_{\Delta}^{EU-CPA}(\mathcal{A})$. It is given by $Adv_{\Delta}^{EU-CPA}(\mathcal{A}) = P[(msg,\delta) \leftarrow \mathcal{A} : Ver_k(msg,\delta)=1$ and $\delta \neq \{\delta_1,..,\delta_n\}]$, where $\delta_1,..,\delta_n$ are the previously outputted δs by Gen_k. The scheme Δ is $EU-CPA$ secure if for all the PPT adversaries, $Adv_{\Delta}^{EU-CPA}(\mathcal{A})$ is negligible.

5.1.3 Security proof

In this section, we present the formal security proof for the proposed protocol. The main goal of such a proof is to prove that the proposed protocol is robust against the sesssion-key (SK) security against PPT adversaries. If $\mathcal{A}$ be a PPT adversary running in polynomial-time t_p against the proposed protocol and the advantage (success probability) of $\mathcal{A}$ in breaking the proposed protocol in time t_p is negligible, we call the proposed scheme offers the SK-security.

Theorem 1 *Let $\mathcal{A}$ be a PPT adversary running in polynomial-time t_p against the proposed protocol ψ in the random oracle. The advantage of $\mathcal{A}$ in breaking the proposed authenticated key exchange (AKE) protocol, ψ's security is* $Adv_{\psi}^{AKE}(\mathcal{A}) \leq 2.[Adv_{G}^{ECCDHP}(t_p) + q_{Send}.Adv_{h(\cdot)}^{HASH}(t_p) + q_{Send}.Adv_{\Delta}^{EU-CPA}(t_p) + Adv^{IND-CPA}\ \Omega(t_p)]$, *where* $Adv_{G}^{ECCDHP}(t_p)$, $Adv_{h(\cdot)}^{HASH}(t_p)$, $Adv_{\Omega}^{IND-CPA}(t_p)$, q_{Send}, $Adv_{\Delta}^{EU-CPA}(t_p)$ *are the advantage of $\mathcal{A}$ in solving ECCDHP for G, the advantage of $\mathcal{A}$ in finding the hash collision in $h(\cdot)$, the advantage of $\mathcal{A}$ in breaking the IND − CPA security of Ω, the number of Send queries, and the advantage of $\mathcal{A}$ in violating the $EU-CPA$ of Δ, respectively.*

Proof In this proof, we consider a sequence of six games namely $Game_0$ - $Game_5$. $Game_0$ is the basic game. The other games viz., $Game_1$, $Game_2$, $Game_3$, $Game_4$, and $Game_5$ are built upon their preceding game(s). Let $success_i$ be an event wherein $\mathcal{A}$ succeeds in the game $Game_i$ in choosing the random bit b_r correctly. The difference in the success probabilities between the previous and current games is studied every time.

- $Game_0$: $\mathcal{A}$ may ask any oracle queries except the following: i) $\mathcal{A}$ is not permitted to ask a $TestAKE(\Pi_E^w)$ query if the instance Π_E^w is no more fresh, and ii) $\mathcal{A}$ is not permitted to ask a $Reveal(\Pi_E^w)$ query if Π_E^w or its partner has already been asked a $TestAKE(\Pi_E^w)$ query. As a result, $\mathcal{A}$ produces a random output as its guess for the random bit b_r. $\mathcal{A}$ succeeds if its random output== b_r chosen by the *TestAKE* oracle. At this point, $\mathcal{A}$'s advantage in breaking the *AKE* security of ψ is given by,

$$Adv_{\psi}^{AKE}(\mathcal{A}) = 2.P_{\psi,\mathcal{A}}[success_0] - 1 \tag{1}$$

- $Game_1$: The first modified game $Game_1$ represents a passive eavesdropping attack wherein $\mathcal{A}$ can run the $Execute(\Pi_{D1}^{w_1}, \Pi_{GWN}^{w_2}, \Pi_{D2}^{w_3})$ oracle query. $\mathcal{A}$ gets all the messages exchanged between the three participants viz., $M1, M2, \eta_1, M5, M6, \eta_2, M9, M10, Cap_{D1}$, $\eta_3, M13,\ M14, Cap_{D1}, \eta_4$. However, the session key $'sk'$ cannot be computed by $\mathcal{A}$ because dk_1, dk_2 are not known to $\mathcal{A}$ and the identities uid_{D1}, id_{GWN}, and uid_{D2}, therefore, cannot be extracted from the acquired messages. Thus, $\mathcal{A}$'s probability of succeeding in $Game_1$ is not increased. This is given by,

$$P_{\psi,\mathcal{A}}[success_0] = P_{\psi,\mathcal{A}}[success_1] \tag{2}$$

- $Game_2$: In this game, the computations of pu_k_{D1} and pu_k_{GWN} are modified as given below:
  - The simulator picks a random point $Y \in G$.
  - For every fresh instance, the simulator chooses the random secrets $r_1, r_2 \in Z_q^*$ and then it sets $pu\_k_{D1} = r_1 Y, pu\_k_{GWN} = r_2 Y$. The simulator computes pu_k_{D1} and pu_k_{GWN} as in $Game_1$ for the other instances.

  Due to the modification in the computations of pu_k_{D1} and pu_k_{GWN}, the simulator is not aware of the ephemeral secrets pr_k_{D1} and pr_k_{GWN}. Hence, it cannot compute the shared secret s_1. Therefore, the simulator cannot compute the secret dk_1. In the same manner, when the computations of pu_k_{D2} and pu_k_{GWN} are modified, the simulator cannot compute the secrets s_2 and dk_2. Due to this, it cannot obtain $uid_{D1}, id_{GWN}, uid_{D2}$ and simply sets the session key $'sk'$ to a random $l-bit$ string. The difference in the success probabilities of $\mathcal{A}$ between $Game_1$ and $Game_2$ is upper bounded by the below equation.

$$\begin{aligned} &|P_{\psi,\mathcal{A}}[success_1] - P_{\psi,\mathcal{A}}[success_2]| \\ &\leq Adv_{G}^{ECCDHP}(t_p) \end{aligned} \tag{3}$$

- $Game_3$: $Game_2$ is modified into $Game_3$ by adding the simulation of $'h'$ oracle and *Send* query. For a query to the $'h'$ oracle on a string $'x'$, the simulator first checks if an entry of the kind (x, str) is present in the LList. It is a list that stores the input-output pairs of $'h'$ oracle. If present, the simulator responds the query by producing the string $'str'$. If not present, the simulator responds the

query by producing a random $l-bit$ string $'str'$ and adds (x, str) to the LList. $Game_3$ models an active attack. In this game, the objective of $\mathcal{A}$ is to trick a participant in accepting a modified message. $\mathcal{A}$ is permitted to use *Send* query and query $'h'$ oracle any number of times for this purpose. $\mathcal{A}$ queries $'h'$ oracle to find the presence of hash collisions. The exchanged messages obtained by $\mathcal{A}$ in $Game_1$ as a result of *Execute* query include $M1, M2, \eta_1, M5, M6, \eta_2, M9, M10, Cap_{D1}, \eta_3, M13, M14, Cap_{D1}, \eta_4$. In case of Cap_{D1}, tricking the participant requires finding a hash collision which is very hard. Besides, Cap_{D1} is used in the messages $M10$, $M14$. Therefore, tricking the participant through these messages is computationally infeasible for $\mathcal{A}$. Hence, the difference in the success probabilities between $Game_2$ and $Game_3$ follows the result of birthday paradox and is given by,

$$|P_{\psi,\mathcal{A}}[success_2] - P_{\psi,\mathcal{A}}[success_3]| \leq q_{Send} \cdot \frac{q_h^2}{(2|h|)}$$

where q_{Send}, q_h, and $|h|$ are the number of *Send* queries, number of $'h'$ oracle queries and range space of $'h'$ respectively. Therefore,

$$|P_{\psi,\mathcal{A}}[success_2] - P_{\psi,\mathcal{A}}[success_3]| \leq q_{Send}.Adv_{h(\cdot)}^{HASH}(t_p) \quad (4)$$

– $Game_4$: Let *Forge* be an event, wherein, $\mathcal{A}$ forwards a query of the type $Send(\Pi_E^w, E'||message)$ such that *message* holds a MAC forgery. In this game, the objective of $\mathcal{A}$ is to output a MAC pair (msg, δ) such that $Ver_k(msg, \delta) = 1$ and this δ was not previously outputted by $Gen_k(msg)$. The secrets dk_1, dk_2 are used as MAC keys in ψ. Let k_n denote the number of MAC keys used for the forgery attempt. It is very clear that $k_n \leq q_{Send}$. The oracles Gen_k, Ver_k are accessible to all the participants and $\mathcal{A}$. $\mathcal{A}$ begins the game by picking a random key k_i from the key space k_n. It then accesses the Gen_{k_i} oracle to generate δ for $'msg'$ and sends the MAC pair (msg, δ) to an instance. The process is repeated. If the event *Forge* occurs against an instance holding the key k_i, $\mathcal{A}$ declares the MAC pair as its forgery. The most crucial thing for $\mathcal{A}$ to win this game is to guess the key correctly. However, guessing the key k_i such that $k_i = dk_1$ or dk_2 is very hard because the shared secrets s_1, s_2 which are based on ECDHE are not known to $\mathcal{A}$. As a result, the session key $'sk'$ cannot be computed. The difference in the success probabilities between $Game_3$ and $Game_4$ is given by,

$$\begin{aligned} &Adv_{\Delta}^{EU-CPA}(\mathcal{A}) = \frac{P[Forge]}{k_n} \\ &P[Forge] \leq q_{Send}.Adv_{\Delta}^{EU-CPA}(\mathcal{A}) \\ &|P_{\psi,\mathcal{A}}[success_3] - P_{\psi,\mathcal{A}}[success_4]| \\ &\leq q_{Send}.Adv_{\Delta}^{EU-CPA}(t_p) \end{aligned} \quad (5)$$

– $Game_5$: In this game, the objective of $\mathcal{A}$ is to identify the correct plaintext in the plaintext pair for a given ciphertext. In this game, $\mathcal{A}$ has access to all the oracles in $Game_4$ in addition to the encryption oracle $'Enc'$. The indistinguishability game is explained below: For each device, $\mathcal{A}$ produces the true identity *uid* and random identity *uidr* as its plaintext pair and forwards it to the challenger. The challenger randomly picks a plaintext from the plaintext pair and encrypts it using $'Enc'$ oracle. Then, the challenger sends the ciphertext to $\mathcal{A}$. $\mathcal{A}$ tries to identify the correct plaintext, *uid* or *uidr*, for the ciphertext. It could not succeed by mere guessing. In the proposed protocol, for the ciphertext messages namely $M1$, $M5$, $M9$, $M13$, $\mathcal{A}$ has no choice other than guessing the correct plaintext due to the use of stateless symmetric cipher for encryption. As a result, it loses the indistinguishability game. $\mathcal{A}$ would not know $uid_{D1}, id_{GWN}, uid_{D2}$ and hence, cannot compute the session key $'sk'$. Therefore, it just guesses the random bit b_r chosen by the *TestAKE* oracle. The difference in the success probabilities between $Game_4$ and the indistinguishability game $Game_5$ is given by the following equation:

$$|P_{\psi,\mathcal{A}}[success_4] - P_{\psi,\mathcal{A}}[success_5]| \leq Adv_{\Omega}^{IND-CPA}(t_p) \quad (6)$$

All the six games are simulated. After querying the *TestAKE* oracle for the session key $'sk'$, $\mathcal{A}$ has no choice other than guessing the random bit b_r to win the game. Hence,

$$P_{\psi,\mathcal{A}}[success_5] = \frac{1}{2} \quad (7)$$

Equation (1) is adjusted to obtain the following equation.

$$\frac{1}{2}.Adv_{\psi}^{AKE}(\mathcal{A}) = |P_{\psi,\mathcal{A}}[success_0] - \frac{1}{2}| \quad (8)$$

According to the triangular inequality,

$$\begin{aligned} &|P_{\psi,\mathcal{A}}[success_1] - P_{\psi,\mathcal{A}}[success_5]| \\ &\leq |P_{\psi,\mathcal{A}}[success_1] - P_{\psi,\mathcal{A}}[success_2]| \\ &+ |P_{\psi,\mathcal{A}}[success_2] - P_{\psi,\mathcal{A}}[success_3]| \\ &+ |P_{\psi,\mathcal{A}}[success_3] - P_{\psi,\mathcal{A}}[success_4]| \\ &+ |P_{\psi,\mathcal{A}}[success_4] - P_{\psi,\mathcal{A}}[success_5]| \end{aligned}$$

Using Eq. (3) through (6), we obtain

$$
\begin{aligned}
&|P_{\psi,\mathcal{A}}[success_1] - P_{\psi,\mathcal{A}}[success_5]| \\
&\quad \leq Adv_G^{ECCDHP}(t_p) + q_{Send}.Adv_{h(\cdot)}^{HASH}(t_p) \\
&\quad + q_{Send}.Adv_{\Delta}^{EU-CPA}(t_p) + Adv_{\Omega}^{IND-CPA}(t_p)
\end{aligned} \tag{9}
$$

Substituting Eqs. (2) and (7) in the L.H.S of Eq. (9), we have

$$
\begin{aligned}
&|P_{\psi,\mathcal{A}}[success_0] - \frac{1}{2}| \\
&\quad \leq Adv_G^{ECCDHP}(t_p) + q_{Send}.Adv_{h(\cdot)}^{HASH}(t_p) \\
&\quad + q_{Send}.Adv_{\Delta}^{EU-CPA}(t_p) + Adv_{\Omega}^{IND-CPA}(t_p)
\end{aligned}
$$

Using Eq. (8), we get the final equation as follows.

$$
\begin{aligned}
&Adv_{\psi}^{AKE}(\mathcal{A}) \leq 2.[Adv_G^{ECCDHP}(t_p) + \\
&q_{Send}.Adv_{h(\cdot)}^{HASH}(t_p) + \\
&q_{Send}.Adv_{\Delta}^{EU-CPA}(t_p) + \\
&Adv_{\Omega}^{IND-CPA}(t_p)]
\end{aligned} \tag{10}
$$

From Eq. (10), it is evident that the advantage of $\mathcal{A}$ in breaking the *AKE* security of ψ is negligible. Thus, the proposed protocol is secure against the PPT adversaries.

5.2 Formal security verification: simulation study using Scyther tool

In this section, the proposed protocol's resilience to different attack vectors in IoT is assessed using the widely-accepted automated software validation tool, known as the Scyther tool. Through the simulation study using the Scyther tool, we show that the proposed scheme is safe against other types of attacks, such as passive secret disclosure, impersonation, traceability and session key computation attacks.

Scyther [19] is a security tool which can be used for verification, falsification, and analysis of security protocols. It uses a pattern refinement algorithm to produce infinite set of traces. The protocol to be verified is provided to the scyther tool in the form of protocol description written using the "Security Protocol Description Language (SPDL)". The protocol description comprises a set of roles. Each role consists of a sequence of events. The events can be send or receive of terms (security parameters).

The entities $D1$, *GWN* and $D2$ are communicating with one another in the proposed protocol. They are modeled as roles $D1$, *GWN* and $D2$ as shown in Figs. 5, 6 and 7, respectively. A role begins with the declaration of the sending and receiving terms, then the exchange of such terms followed by the security claims. The security claims are used to model the protocol's security properties. These claims are crucial part of the protocol description without which scyther would not know what is to be verified. Fig. 8

```
role D1 {
fresh uidD1,r,n1,rA,msg,n5:Nonce;
var idGWN,uidD2,n4:Nonce;
send_1(D1,GWN,{uidD1,r}dk1,M2,n1);
recv_4(GWN,D1,{idGWN,uidD2}dk1,M14,CapD1,n4);
match(M15,M14);
send_5(D1,GWN,CapD1,rA,M17,msg,n5);
claim_D11(D1,Secret,dk1);
claim_D12(D1,Secret,uidD2);
claim_D13(D1,Secret,idGWN);
claim_D14(D1,SKR,sk);
claim_D15(D1,Alive);
claim_D16(D1,Weakagree);}
```

Fig. 5 Role for IoT device $D1$ in SPDL

```
role GWN {
fresh idGWN,n2,n4,msg,n6:Nonce;
var uidD1,uidD2,r,n1,n3,n5,rA,msg:Nonce;
recv_1(D1,GWN,{uidD1,r}dk1,M2,n1);
match(M3,M2);
send_2(GWN,D2,{idGWN,uidD1}dk2,M6,n2);
recv_3(D2,GWN,{uidD2}dk2,M10,CapD1,n3);
match(M11,M10);
send_4(GWN,D1,{idGWN,uidD2}dk1,M14,CapD1,n4);
recv_5(D1,GWN,CapD1,rA,M17,msg,n5);
match(M18,M17);
send_6(GWN,D2,CapD1,rA,M19,msg,n6);
claim_GWN1(GWN,Secret,dk1);
claim_GWN2(GWN,Secret,dk2);
claim_GWN3(GWN,Secret,uidD1);
claim_GWN4(GWN,Secret,uidD2);
claim_GWN5(GWN,SKR,sk);
claim_GWN6(GWN,Alive);
claim_GWN7(GWN,Weakagree);}
```

Fig. 6 Role for gateway node *GWN* in SPDL

```
role D2 {
fresh uidD2,n3:Nonce;
var idGWN,uidD1,n2,n6,rA,msg:Nonce;
recv_2(GWN,D2,{idGWN,uidD1}dk2,M6,n2);
match(M7,M6);
send_3(D2,GWN,{uidD2}dk2,M10,CapD1,n3);
recv_6(GWN,D2,CapD1,rA,M19,msg,n6);
match(M20,M19);
claim_D21(D2,Secret,dk2);
claim_D22(D2,Secret,uidD1);
claim_D23(D2,Secret,idGWN);
claim_D24(D2,SKR,sk);
claim_D25(D2,Alive);
claim_D26(D2,Weakagree);}
```

Fig. 7 Role for IoT device $D2$ in SPDL

confirms that the roles $D1$, *GWN*, and $D2$ are reachable. This ensures that there is no obvious weakness in the protocol description.

Our claims on role $D1$ include (i) key dk_1 is secret, (ii) uid_{D2} is secret, (iii) id_{GWN} is secret, (iv) session key $'sk'$ is secret, (v) aliveness, and vi) weak agreement. Secondly,

Scyther results : characterize				
Claim			**Status**	
Proposed	D1	Reachable	Ok	Verified
	GWN	Reachable	Ok	Verified
	D2	Reachable	Ok	Verified

Fig. 8 Scyther results: verification of reachability of the roles for $D1$, GWN and $D2$

our claims on role GWN comprise i) dk_1 is secret, ii) key dk_2 is secret, iii) uid_{D1} is secret, iv) uid_{D2} is secret, V) $'sk'$ is secret, vi) aliveness, and vii) weak agreement. Thirdly, our claims on role $D2$ are i) dk_2 is secret, ii) uid_{D1} is secret, iii) id_{GWN} is secret, iv) $'sk'$ is secret, v) aliveness, and vi) weak agreement.

From the scyther verification results in Fig. 9, we have drawn useful insights which are summarized in Table 2. Firstly, the keys dk_1, dk_2 derived from s_1, s_2 are secret. As a result, the passive secret disclosure attack is prevented in the system. Secondly, the system is secure against impersonation attack because the identities uid_{D1}, id_{GWN} and uid_{D2} are declared secret. Thirdly, the system is protected from traceability attack since no attacks are reported on uid_{D1}, uid_{D2}. In fact, these identities are freshly generated every time. Lastly, the system is resistant to session key computation attack since the identities required to compute $'sk'$ are secret.

5.3 Informal security analysis

In this section, we show that the proposed protocol is secure against various attack vectors in IoT environment by intuitive reasoning. It is worth noticing that we follow the informal (non-mathematical heuristic) security analysis to show the proposed protocol is secure against other attacks that are not covered so far in Sects. 5.1 and 5.2.

Proposition 1 *SUACC-IoT prevents brute-force attack.*

Scyther results : verify					
Claim				**Status**	**Comments**
Proposed	D1	Proposed,D11	Secret h1(k(D1,GWN),pk(D1),pk(GWN))	Ok	No attacks within bounds.
		Proposed,D12	Secret uidD2	Ok	No attacks within bounds.
		Proposed,D13	Secret idGWN	Ok	No attacks within bounds.
		Proposed,D14	SKR h1(uidD1,uidD2,idGWN)	Ok	No attacks within bounds.
		Proposed,D15	Alive	Ok	No attacks within bounds.
		Proposed,D16	Weakagree	Ok	No attacks within bounds.

(a) Verifying $D1$'s claims.

Scyther results : verify					
Claim				**Status**	**Comments**
Proposed	GWN	Proposed,GWN1	Secret h1(k(D1,GWN),pk(D1),pk(GWN))	Ok	No attacks within bounds.
		Proposed,GWN2	Secret h1(k(D2,GWN),pk(D2),pk(GWN))	Ok	No attacks within bounds.
		Proposed,GWN3	Secret uidD1	Ok	No attacks within bounds.
		Proposed,GWN4	Secret uidD2	Ok	No attacks within bounds.
		Proposed,GWN5	SKR h1(idGWN,uidD1,uidD2)	Ok	No attacks within bounds.
		Proposed,GWN6	Alive	Ok	No attacks within bounds.
		Proposed,GWN7	Weakagree	Ok	No attacks within bounds.

(b) Verifying GWN's claims.

Scyther results : verify					
Claim				**Status**	**Comments**
Proposed	D2	Proposed,D21	Secret h1(k(D2,GWN),pk(D2),pk(GWN))	Ok	No attacks within bounds.
		Proposed,D22	Secret uidD1	Ok	No attacks within bounds.
		Proposed,D23	Secret idGWN	Ok	No attacks within bounds.
		Proposed,D24	SKR h1(uidD2,uidD1,idGWN)	Ok	No attacks within bounds.
		Proposed,D25	Alive	Ok	No attacks within bounds.
		Proposed,D26	Weakagree	Ok	No attacks within bounds.

(c) Verifying $D2$'s claims.

Fig. 9 Scyther results: verification of claims

Table 2 Inferences from Scyther results

System parameter(s) verified	Scyther comment	Attack prevented in the system
dk_1, dk_2	No attacks within bounds	Passive secret disclosure
uid_{D1}, id_{GWN}, uid_{D2}	No attacks within Bounds	Impersonation
uid_{D1}, uid_{D2}	No attacks within bounds	Traceability
sk	No attacks within bounds	Session key computation

Proof The private keys pr_k_{D1} and pr_k_{GWN} of $D1$ and GWN respectively are of at least $k-1$ bits, where k is large. Now, even if $\gcd(pr\_k_{D1}, pr\_k_{GWN}) = pr\_k_{D1}$, pu_k_{GWN} cannot be expressed in term of pu_k_{D1} by brute-forcing approach, where $\gcd(x, y)$ represents the greatest common divisor of two numbers x and y. In the same manner, considering $D2$ and GWN, pu_k_{GWN} cannot be also expressed in term of pu_k_{D2} with $\gcd(pr\_k_{D2}, pr\_k_{GWN}) = pr\_k_{D2}$. Thus, the brute-force attacks based on the key sizes are prevented in the proposed protocol.

Proposition 2 *SUACC-IoT is resilient to man-in-the-middle (MITM) and replay attacks.*

Proof Suppose an adversary $\mathcal{A}$ intercepts the exchanges messages during the communication among the various entities in the network and tries to modify the messages on the fly so that the recipients will not be aware of the modified messages and the adversary $\mathcal{A}$ will force the recipients to believe that the messages are genuine. In order to do so, in the authentication stage, if $\mathcal{A}$ carries out an active MITM attack like "intercept and modify" on the exchanged parameters exchanged, such as $M1$, η_1, $M5$, η_2, $M9$, Cap_{D1}, η_3, $M13$, η_4, he would not succeed because the integrity of these parameters is guaranteed by the respective message authentication codes $M2 = MAC(dk_1, M1\|\eta_1)$, $M6 = MAC(dk_2, M5\|\eta_2)$, $M10 = MAC(dk_2, M9\|Cap_{D1}\ \|\eta_3)$, $M14 = MAC(dk_1, M13\|\ Cap_{D1}\|\ \eta_4)$, and the adversary needs to know the secret credentials. On the other hand, the replay attack is also prevented by random one-time nonces $\eta_1, \eta_2, \eta_3, \eta_4$ and the respective message authentication codes. Therefore, in the access control stage, the replay attack is prevented using the random nonces η_5 and η_6, and the corresponding message authentication codes. As a result, the proposed protocol resists both replay and MITM attacks.

Proposition 3 *SUACC-IoT prevents traceability attack.*

Proof Traceability attack is prevented in the system by the use of universally unique identifiers for IoT devices and dynamic messages created by the entities during the communication. The universally unique identifier is a cryptographically strong, random, and unique identifier, which is collision-resistant. In order to have a collision with a probability of 0.5, 2.71 quintillion identifiers are to be generated which is computationally infeasible. Due to the above reasons, uid_{D1} and uid_{D2} generated in different sessions would be different and unique. Consequently, the capability token, such as $Cap_{D1} = h(uid_{D1}, r, ctxt, AR, Rnd)$ and MACs: $M2$, $M6$, $M10$, $M14$ in different sessions would be different. Moreover, in each session, the exchanged messages are dynamic and unique due to usage of the random nonces. Therefore, the traceability is prevented in the proposed SUACC-IoT.

Proposition 4 *SUACC-IoT preserves anonymity property.*

Proof During the authentication stage of the proposed SUACC-IoT, whether it is uid_{D1} or uid_{D2}, the identity of a device is protected by the symmetric key encryption and the corresponding key as $M1 = E(uid_{D1}\|r, dk_1)$ and $M9 = E(uid_{D2}, dk_2)$. In addition, when a device $D2$ sends Cap_{D1} to another device $D1$, uid_{D1} is hidden as in $Cap_{D1} = h(uid_{D1}, r, ctxt, AR, Rnd)$. Moreover, when MACs, such as $M10$ and $M14$ are sent, Cap_{D1} is then hidden; thereby, uid_{D1} is also hidden. During the access control stage of SUACC-IoT, $D1$ submits Cap_{D1} to $D2$ for granting the requested access where uid_{D1} is hidden. In this way, anonymity is preserved in the SUACC-IoT. □

Proposition 5 *SUACC-IoT prevents session-key computation attack.*

Proof In the proposed SUACC-IoT, the identities uid_{D1}, id_{GWN} and uid_{D2} required to compute the session key sk that are protected by symmetric key encryption and the corresponding keys dk_1, dk_2 in $M1 = E(uid_{D1}\|r, dk_1)$, $M5 = E(id_{GWN}\|M4, dk_2)$, $M9 = E(uid_{D2}, dk_2)$ and $M13 = E(id_{GWN}\|M12, dk_1)$. Thus, the session key computation is computationally infeasible for an adversary without having the knowledge of the secret credentials dk_1 and dk_2. Hence, the session-key computation attack is prevented in the proposed SUACC-IoT. □

Proposition 6 *SUACC-IoT protects the system parameters from passive secret disclosure attack.*

Proof If an adversary $\mathcal{A}$ eavesdrops or intercepts on the messages during the communication to read the parameters $M1$, $M5$, $M9$, $M13$, Cap_{D1} containing in the exchanged messages, he would not be able to disclose the secrets of

the entities in the network due to the following reasons. Firstly, the adversary would need the permanent (long-term) secrets dk_1 and dk_2 which are derived from the keys based on ECDHE to disclose the secrets uid_{D1}, r, id_{GWN}, and uid_{D2} in the $M1$, $M5$, $M9$ and $M13$. Secondly, the secrets contained in Cap_{D1}, such as uid_{D1}, r, $ctxt$ and AR cannot be disclosed, because Cap_{D1} uses collision-resistant one way cryptographic hash function. Thus, the system parameters are protected from passive secret disclosure attack in the SUACC-IoT due to the hardness of computational ECDLP and collision-resistant property of one way cryptographic hash function. □

Proposition 7 *SUACC-IoT is secure against device impersonation attack.*

Proof An impersonation attack allows an adversary to attempt to falsify a unauthenticated message to defraud other recipient parties in IoT network on behalf of a sending party. In such a scenario, the receiver will be forced to believe that the message has come from a genuine entity [32]. Suppose an adversary $\mathcal{A}$ intercepts the exchanged messages and gets $M1$, $M2$, $M9$, $M10$, Cap_{D1}. After that $\mathcal{A}$ tries to create valid $M1'$, $M2'$, $M9'$, $M10'$, Cap'_{D1} on behalf of $D1$ and $D2$. In that case, $\mathcal{A}$ would be unsuccessful due to the following reasons. Firstly, in order to compute valid ciphertexts $M1$ and $M9$, he would need the secrets uid_{D1}, r, uid_{D2} and dk_1, dk_2 that are not known. Secondly, he would require the unknown (long-term) secret credentials dk_1 and dk_2 to compute valid MACs, such as $M2$ and $M10$. Thirdly, for compute valid capability token Cap_{D1}, he would need uid_{D1}, r, $ctxt$, AR, Rnd which are not known to $\mathcal{A}$. From these discussions, it is clear that without having the secret credentials, the adversary $\mathcal{A}$ can not create valid messages on behalf of any IoT devices $D1$ and $D2$, and send the messages on behalf of $D1$ and $D2$. Thus, SUACC-IoT is secure against device impersonation attack. □

Proposition 8 *SUACC-IoT is resistant to gateway impersonation attack.*

Proof Similar to Proposition 7, assume that $\mathcal{A}$ intercepts the exchanged messages to know $M5$, $M6$, $M13$ and $M14$ in order to create valid $M5'$, $M6'$, $M13'$, $M14'$ on behalf of the gateway node, GWN. However, $\mathcal{A}$ would not succeed, because of the following: (i) to compute valid ciphertexts $M5$ and $M13$ the adversary $\mathcal{A}$ would require the secret credentials id_{GWN}, uid_{D1}, uid_{D2} and dk_1, dk_2 which are not known to $\mathcal{A}$, and (ii) to compute valid MACs, such as $M6$ and $M14$ he would need the unknown secret credentials dk_1 and dk_2. We then see that without having the secret credentials, it is also infeasible task for the adversary $\mathcal{A}$ to create legitimate messages on behalf of the GWN. As a result, the proposed SUACC-IoT is resistant to gateway impersonation attack. □

Proposition 9 *SUACC-IoT provides protection from gateway bypass attack.*

Proof In the proposed SUACC-IoT, both the devices $D1$ and $D2$ establish a session key sk via the gateway node GWN. $D1$ cannot compute $M5$ and $M6$ as it does not know the secret key dk_2. In a similar manner, $D2$ cannot also compute $M13$ and $M14$ since it is not aware of the secret key dk_1. However, the GWN knows both the secrets dk_1 and dk_2. Moreover, $D1$ receives Cap_{D1} from $D2$ via GWN. In addition, during the access control stage, $D1$ submits Cap_{D1} to $D2$ through the GWN. Hence, neither $D1$ nor $D2$ could bypass the GWN. As a result, the GWN bypass attack is protected in SUACC-IoT. □

Proposition 10 *SUACC-IoT prevents Denial-of-Service (DoS) attack.*

Proof When $\mathcal{A}$ accesses a particular resource over and over again in the network, DoS can take place. In the proposed SUACC-IoT, DoS attack using single identity is prevented because it restricts access to a resource for an identity to only one session at a time. Moreover, even if an adversary mounts the replay attacks to send old messages to the recipients, due to the lightweight cryptographic primitives used in the proposed SUACC-IoT the adversary can not consume more resource from the recipient side. Thus, DoS attack is resisted in the proposed SUACC-IoT. □

Proposition 11 *SUACC-IoT is resilient to dictionary attacks.*

Proof Suppose an adversary $\mathcal{A}$ carries out a dictionary attack to determine the decryption keys dk_1 and dk_2 so as to decipher the ciphertexts in the system (for example, to compute the parameters like $M1$, $M5$, $M9$ and $M13$. However, $\mathcal{A}$ would not be successful, because the secret keys dk_1 and dk_2 are derived from the shared secrets s_1 and s_2 generated using the ECDHE. Besides, the stateless Cipher Block Chaining (CBC) mode of the applied symmetric encryption is used for symmetric key encryption operations in the system as in [3]. Hence, SUACC-IoT is resilient to dictionary attack due to hardness of the computational ECDLP and "indistinguishability under chosen plaintext attack (IND-CPA) security" of stateless CBC mode of symmetric cipher. □

6 Testbed results and discussions

In this section, the proposed SUACC-IoT is evaluated for key performance parameters, namely CPU and memory usage, and computational overhead in an IoT testbed setup using Raspberry PI 3 [33]. Besides, the communication cost and security features of the system are assessed. Furthermore, the proposed system is also compared with the existing competing schemes.

6.1 Cryptographic standards used in testbed

The proposed system is implemented in Java using Java Cryptography Architecture [34] and BouncyCastle [35] libraries. The cryptographic standards mentioned in Table 3 are used in the implementation of the proposed system. The reasons for the choice of the cryptographic standards are provided below: i) Curve25519 [36] is used for secret key generation using ECDHE since it is a highly performance optimized, fast, and safe elliptic curve, ii) Advanced Encryption Standard (AES-256) [37], which has a key length of 256 bits, in stateless CBC cipher mode is used for symmetric encryption/decryption so that the resultant ciphertext is different every time and satisfies IND-CPA security, iii) Secure Hash Algorithm (SHA-256) [38], which produces 256 bits hash output, is used for finding cryptographic hash so that the generated hash is collision-resistant, and iv) CBC-MAC based on AES is used for finding message authentication code so that EU-CPA is fulfilled.

6.2 IoT testbed experiments

In this section, we provide the IoT testbed experiments using Raspberry PI 3 setting [33] for measuring CPU and memory usage as well as computational time based on the cryptographic standards used in Sect. 6.1.

Table 3 Cryptographic standards used

Cryptographic primitive	Standard used
ECDHE	Curve25519
Symmetric encryption/decryption	AES-256 stateless CBC Cipher mode
Hash function	SHA-256
Message authentication code	CBC-MAC based on AES

6.2.1 CPU and memory usage

To the best of our knowledge, the existing schemes did not consider CPU and memory usage parameters in their performance evaluation. We conducted four measurement runs in our IoT testbed to quantify the CPU and memory usage of the proposed system. The experimental results in Fig. 10 indicate that the SUACC-IoT system's maximum CPU usage is 29.35% and maximum memory usage is 2.79% which are quite acceptable.

6.2.2 Computational cost

The third essential parameter we considered in the performance evaluation is computational overhead. We conducted four measurement runs in our IoT testbed. Fig. 11 presents the overhead in the different measurement runs. The average computational overhead computed from the different overheads is 744.5 ms which is fairly acceptable.

The theoretical computational overhead of the SUACC-IoT system and the different authentication approaches under consideration are presented in Table 4. Let T_H, T_{MAC}, $T_{ENC/DEC}$, T_{FE}, T_{PUF}, T_O, and $T_{K.GEN}$ denote the overhead to execute hash, MAC, symmetric encryption/decryption, fuzzy extractor generation and reproduction, physical unclonable function, addition and multiplication, and key generation using ECDHE computations respectively. The computational overhead of SUACC-IoT is $2T_H + 1T_{K.GEN} + 4T_{ENC/DEC} + 4T_{MAC}$. The time complexities of hash, key generation, symmetric encryption/decryption and MAC are deemed $\mathcal{O}(n)$ for a n-bit message. Therefore, the overall time complexity of the proposed system is $\mathcal{O}(n)$. The system involves key generation and symmetric encryption/decryption operations which are avoided in the schemes [2, 5, 9]. However, this is complemented by the security features of the proposed system.

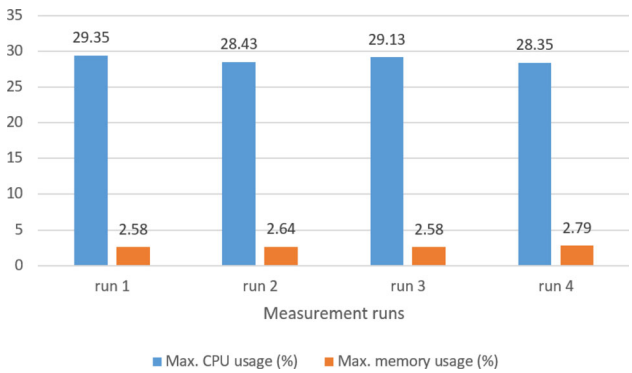

Fig. 10 CPU and memory usage

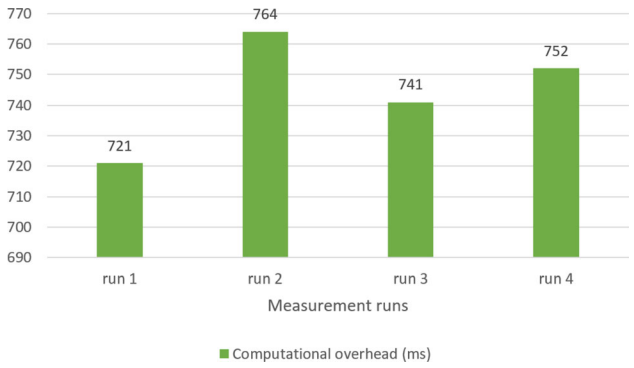

Fig. 11 Computational overheads

Table 4 Comparison of computational costs among different authentication approaches and SUACC-IoT

Approach	Computational cost
Srinivas et al. [5]	$35T_H$
Aman et al. [6]	$1T_H + 4T_{MAC} + 3T_{ENC}$
Alotaibi [7]	$14T_H + 4T_{ENC}$
Gope et al. [8]	$5T_H + 1T_{FE} + 2T_{PUF}$
Adeel et al. [9]	$23T_H$
Aghili et al. [2]	$29T_H$
Wazid et al. [3]	$22T_H + 4T_{ENC/DEC} + 1T_{FE}$
Kim et al. [10]	$6T_O + 6T_H + 2T_{ENC/DEC}$
SUACC-IoT	$2T_H + 1T_{K.GEN} + 4T_{ENC/DEC} + 4T_{MAC}$

6.3 Communication cost

Communication cost is yet another key parameter. Table 5 shows that the number of messages exchanged in SUACC-IoT is 6 which is reasonably acceptable. Besides, the size of the longest message exchanged $\{M9, M10, Cap_{D1}, \eta_3\}$ or $\{M13, M14, Cap_{D1}, \eta_4\}$ in SUACC-IoT is $(384 + 128 + 256 + 104) = 872$ bits. This size is less compared to those in the schemes [3, 5–7]. Thus, the communication cost incurred for the longest message exchanged in the proposed system is 872 bits which is reasonably acceptable.

6.4 Security and functionality features

Table 6 presents the comparison of security features between the different authentication approaches under consideration and SUACC-IoT. From the table, it can be observed that SUACC-IoT has better security features compared to the other approaches. The proposed system is secure against various attack vectors in IoT namely MITM, replay, traceability, session key computation, passive secret disclosure, device impersonation, gateway impersonation and bypass, offline dictionary, and DoS attacks.

In Table 7, the different access control approaches are compared with the SUACC-IoT system. Five features namely context-awareness, granularity, scalability, interoperability, and security are considered for comparison. SUACC-IoT supports all the features considered while the other approaches do not. In a nutshell, SUACC-IoT performs fairly well and has better security features compared to the closely related existing schemes.

7 Conclusion and future works

In this paper, we presented a new secure unified authentication and access control system based on capability for IoT, called SUACC-IoT. SUACC-IoT brings the following advantages to authentication and access control in IoT:

– Security: In the proposed protocol, an IoT device communicates with another device through the gateway node. The two devices and gateway node are mutually authenticated to one another. Two types of mutual authentication happen: (1) between the device and gateway node and (2) between the devices. Various attack vectors such as MITM, replay, traceability, session key computation, secret disclosure, device impersonation, gateway impersonation, gateway

Table 5 Comparison of communication costs among different authentication approaches and SUACC-IoT

Approach	No. of messages exchanged	Longest message exchanged (bits)
Srinivas et al. [5]	4	944
Aman et al. [6]	7	960
Alotaibi [7]	4	1146
Gope et al. [8]	3	–
Adeel et al. [9]	6	512
Aghili et al. [2]	4	832
Wazid et al. [3]	4	1280
Kim et al. [10]	4	864
SUACC-IoT	6	872

Table 6 Comparison of security features among different authentication approaches and SUACC-IoT

Security feature	[5]	[6]	[7]	[8]	[9]	[2]	[3]	[10]	SUACC-IoT
Resilience to MITM attack	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
Resistance to replay attack	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Prevents traceability attack	Yes	No	No	Yes	No	Yes	Yes	No	Yes
Assists anonymity	Yes	No	Yes	Yes	No	Yes	Yes	No	Yes
Prevents session-key computation attack	Yes	No	Yes	No	Yes	Yes	Yes	Yes	Yes
Protection against secret disclosure attack	No	Yes	No	Yes	No	No	No	No	Yes
Secure against device impersonation attack	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Resistant to gateway impersonation attack	Yes	No	Yes	No	No	No	Yes	No	Yes
Protection from gateway bypass attack	No	No	No	No	No	No	Yes	No	Yes
Prevents single-identity DoS attack	Yes	No	Yes	No	Yes	Yes	No	No	Yes
Resilience to dictionary attack	Yes	No	No	No	No	Yes	Yes	No	Yes
Refrain from offline device registration	No	Yes	Yes	No	No	Yes	No	Yes	Yes

Table 7 Comparison of features among various access control approaches and SUACC-IoT

Approach	Context-awareness	Granularity	Scalability	Interoperability	Security
Xu et al. [11]	Yes	Fine-grained	No	No	No
Aghili et al. [2]	No	Fine-grained	No	Yes	Yes
Yang et al. [12]	Yes	Fine-grained	Yes	No	Yes
Bao et al. [15]	No	Fine-grained	No	No	Yes
SUACC-IoT	Yes	Fine-grained	Yes	Yes	Yes

bypass, offline dictionary and DoS are addressed in the protocol. Security is also ensured during access control.

- Lightweight: The proposed system involves only lightweight cryptographic operations such as ECDHE using highly performance optimized and fast elliptic curve, symmetric key encryption/decryption, message authentication code, and hash. Furthermore, the proposed system uses TLS only during the setup stage. Thus, the proposed system is a lightweight system suitable for use in resource-constraint IoT.
- Scalability: The gateway node performs only limited number of lightweight ECDHE, symmetric key encryption/decryption and message authentication code operations in the protocol. The two communicating devices generate the required parameters on their own. The devices do not overload the gateway node. As a result, the proposed system performs fairly well with the increase in the number of devices.
- Interoperability: The gateway node acts as a protocol bridge to ensure protocol compatibility across various IoT devices. Thus, the gateway node ensures device interoperability.

SUACC-IoT shows promising results in the key performance parameters, namely CPU and memory usage, computational overhead, and communication cost. The protocol's maximum CPU usage is 29.35%, maximum memory usage is 2.79%, and computational overhead is 744.5 ms which are quite reasonable. The communication cost incurred for the longest message exchanged in the protocol is 872 bits which is fairly acceptable. Furthermore, SUACC-IoT has better security features compared to the closely related existing schemes. These make SUACC-IoT usable in various resource-constraint IoT environments.

Some future works are as follows. The first future work is to design a decentralized framework for the system to make the system more scalable. The mobility management among the heterogeneous network slices in a 5th generation mobile network (5G) network is an important issue [39]. Therefore, second future work may be on the mobility management task in IoT-enaled 5G environment where the subscription-based connectivity services for the end users should be granted based on access capability. Recently, the privacy-preserving bilateral access control with fine-granularity in an IoT-enabled healthcare has been suggested where only authorized counterparts will be able to access the health-related information [13]. As a result, another interesting future work may include to integrate privacy-preserving bilateral access control with fine-granularity with the proposed protocol.

Acknowledgements The authors would like to thank the anonymous reviewers, associate editor and editor-in-chief for their valuable feedback on the paper.

Author contributions Conceptualization, NS, KVB, MR; Methodology, NS, MR and AKD; Security analysis, NS and AKD; Investigation, NS, KVB, MR, AKD, JJPCR; Writing-original draft preparation and writing-review and editing, NS, KVB, MR; Supervision, NS, KVB, MR, AKD; Funding acquisition, MR and JJPCR.

Funding Open access funding provided by Manipal Academy of Higher Education, Manipal. Open access funding provided by Manipal Academy of Higher Education, Manipal. This work is partially funded by FCT/MCTES through national funds and when applicable co-funded EU funds under the Project UIDB/50008/2020; and by Brazilian National Council for Scientific and Technological Development - CNPq, via Grant No. 313036/2020-9.

Data availability Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose. The authors have no conflicts of interest to declare that are relevant to the content of this article. All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript. The authors have no financial or proprietary interests in any material discussed in this article.

References

1. Internet of Things - number of connected devices worldwide 2015-2025, https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/ (Statista Research Department. Accessed on March 2021)
2. Aghili, S.F., Mala, H., Shojafar, M., Peris-Lopez, P.: LACO: lightweight three-factor authentication, access control and ownership transfer scheme for E-health systems in IoT. Futur. Gener. Comput. Syst. **96**, 410–424 (2019)
3. Wazid, M., Das, A.K., Odelu, V., Kumar, N., Susilo, W.: Secure remote user authenticated key establishment protocol for smart home environment. IEEE Trans. Dependable Secure Comput. **17**(2), 391–406 (2020)
4. Ferrara, P., Mandal, A.K., Cortesi, A., Spoto, F.: Static analysis for discovering IoT vulnerabilities. Int. J. Softw. Tools Technol. Transfer (Springer) **23**, 71–88 (2021)
5. Srinivas, J., Mukhopadhyay, S., Mishra, D.: Secure and efficient user authentication scheme for multi-gateway wireless sensor networks. Ad-Hoc Netw. **54**, 147–169 (2017)
6. Aman, M.N., Chua, K.C., Sikdar, B.: Mutual authentication in IoT systems using physical unclonable functions. IEEE Internet Things J. **4**(5), 1327–1340 (2017)
7. Alotaibi, M.: An enhanced symmetric cryptosystem and biometric-based anonymous user authentication and session key establishment scheme for WSN. IEEE Access **6**, 70072–70087 (2018)
8. Gope, P., Sikdar, B.: Lightweight and privacy-preserving two-factor authentication scheme for IoT devices. IEEE Internet Things J. **6**(1), 580–589 (2018)
9. Adeel, A., Ali, M., Khan, A.N., Khalid, T., Rehman, F., Jararweh, Y., Shuja, J.: Trans. Emerg. Telecommun. Technol. A multi-attack resilient lightweight IoT authentication scheme **33**, e3676 (2019)
10. Kim, T.-H., Kumar, G., Saha, R., Buchanan, W.J., Devgun, T., Thomas, R.: LiSP-XK: extended light-weight signcryption for IoT in resource-constrained environments. IEEE Access **9**, 100972–100980 (2021)
11. Xu, R., Chen, Y., Blasch, E., Chenc, G.: A Federated Capability-based Access Control Mechanism for Internet of Things (IoTs), in: Proceedings of the Conference on Sensors and Systems for Space Applications, SPIE Defense and Commercial Sensing 2018 (DCS), (2018), pp. 1–17
12. Yang, Q., Zhang, M., Zhou, Y., Wang, T., Xia, Z., Yang, B.: A non-interactive attribute-based access control scheme by blockchain for IoT. Electronics **10**, 1–11 (2021)
13. Sun, J., Yuan, Y., Tang, M., Cheng, X., Nie, X., Aftab, M.U.: Privacy-preserving Bilateral Fine-grained Access Control for Cloud-enabled Industrial IoT Healthcare. IEEE Trans. Ind. Inf. (2021). https://doi.org/10.1109/TII.2021.3133345
14. Bao, Y., Qiu, W., Tang, P., Cheng, X.: Efficient, revocable and privacy-preserving fine-grained data sharing with keyword search for the cloud-assisted medical IoT system. IEEE J. Biomed. Health Inf. (2021). https://doi.org/10.1109/JBHI.2021.3100871
15. Bao, Y., Qiu, W., Cheng, X.: Secure and lightweight fine-grained searchable data sharing for IoT-oriented and cloud-assisted smart healthcare system. IEEE Internet Things J. (2021). https://doi.org/10.1109/JIOT.2021.3063846
16. Wang, J., Hu, F., Zhou, Y., Liu, Y., Zhang, H., Liu, Z.: BlueDoor: Breaking the Secure Information Flow via BLE Vulnerability, in: 18th International Conference on Mobile Systems, Applications, and Services (MobiSys '20), Toronto, Ontario, Canada, (2020), pp. 286–298
17. Michalevsky, Y., Nath, S., Liu, J.: MASHaBLE: Mobile Applications of Secret Handshakes over Bluetooth LE, in: 22nd Annual International Conference on Mobile Computing and Networking (MobiCom '16), New York City, New York, USA, (2016), pp. 387–400
18. Abdalla, M., Fouque, P. A., Pointcheval, D.: Password-based authenticated key exchange in the three-party setting, in: 8th International Workshop on Theory and Practice in Public Key Cryptography (PKC'05), Lecture Notes in Computer Science, Vol. 3386, Les Diablerets, Switzerland, (2005), pp. 65–84
19. Cremers, C.: The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols (Tool Paper), in: Proceedings of the 20th international conference on Computer Aided Verification, (2008), pp. 414–418
20. Feng, C., Yu, K., Aloqaily, M., Alazab, M., Lv, Z., Mumtaz, S.: Attribute-based encryption with parallel outsourced decryption for edge intelligent IoV. IEEE Trans. Veh. Technol. **69**(11), 13784–13795 (2020)

21. Yin, L., Feng, J., Xun, H., Sun, Z., Cheng, X.: A privacy-preserving federated learning for multiparty data sharing in social IoTs. IEEE Trans. Netw. Sci. Eng. **8**(3), 2706–2718 (2021)
22. Bao, Y., Qiu, W., Cheng, X.: Efficient and fine-grained signature for IIoT with resistance to key exposure. IEEE Internet Things J. **8**(11), 9189–9205 (2021)
23. Mohajer, A., Bavaghar, M., Saboor, R., Payandeh, A.: Secure dominating set-based routing protocol in MANET: Using reputation, in: 10th International ISC Conference on Information Security and Cryptology (ISCISC'13), Yazd, Iran, (2013), pp. 1–7
24. Kumar, A., Sharma, S., Goyal, N., Singh, A., Cheng, X., Singh, P.: Secure and energy-efficient smart building architecture with emerging technology IoT. Comput. Commun. **176**, 207–217 (2021)
25. Kurniawan, A., Kyas, M.: A trust model-based Bayesian decision theory in large scale Internet of Things, in: IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP'15), Singapore, (2015), pp. 1–5
26. Imani, M., Ghoreishi, S.F.: Graph-based Bayesian optimization for large-scale objective-based experimental design. IEEE Trans. Neural Netw. Learn. Syst. (2021). https://doi.org/10.1109/TNNLS.2021.3071958
27. Kang, J., Fan, K., Zhang, K., Cheng, X., Li, H., Yang, Y.: An ultra light weight and secure RFID batch authentication scheme for IoMT. Comput. Commun. **167**, 48–54 (2021)
28. Zhang, Q., Xu, D.: Security authentication technology based on dynamic Bayesian network in Internet of Things. J. Ambient. Intell. Humaniz. Comput. **11**(2), 573–580 (2020)
29. Mauw, S., Bos, V.: Drawing message sequence charts with LaTeX. TUGBoat **22**, 87–92 (2001)
30. Wang, D., He, D., Wang, P., Chu, C.-H.: Anonymous two-factor authentication in distributed systems: certain goals are beyond attainment. IEEE Trans. Dependable Secure Comput. **12**(4), 428–442 (2015)
31. Burrows, M., Abadi, M., Needham, R.: A logic of authentication. ACM Trans. Comput. Syst. **8**(1), 18–36 (1990)
32. Das, A.K., Zeadally, S., He, D.: Taxonomy and analysis of security protocols for Internet of Things. Futur. Gener. Comput. Syst. **89**, 110–125 (2018)
33. Raspberry Pi 3 Model B+, Accessed on April 2021 (2020). https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/
34. Java Cryptography Architecture, https://docs.oracle.com/javase/7/docs/technotes/guides/security/crypto/CryptoSpec.html. Accessed on March 2021
35. Bouncy Castle Cryptography Library 1.70, https://www.bouncycastle.org/docs/docs1.5on/index.html. Accessed on March 2021
36. Bernstein, D. J.: Curve25519: New Diffie-Hellman Speed Records, in: 9th International Workshop on Theory and Practice in Public Key Cryptography (PKC '06), New York, NY, USA, (2006), pp. 207–228
37. Advanced Encryption Standard (AES), FIPS PUB 197, National Institute of Standards and Technology (NIST), U.S. Department of Commerce, November 2001. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf. Accessed on Jun 2021
38. May, W. E.: Secure Hash Standard, http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf. Accessed on Feb 2021 (2015)
39. Balasubramanian, V., Zaman, F., Aloqaily, M., Ridhawi, I. A., Jararweh,Y., Salameh, H. B.: A Mobility Management Architecture for Seamless Delivery of 5G-IoT Services, in: IEEE International Conference on Communications (ICC'19), Shanghai, China, (2019), pp. 1–7

N. Sivaselvan received his M.E. degree from Anna University Tiruchirapalli. He is currently pursuing the Ph.D. degree in Information Engineering with the Department of Electrical and Computer Engineering, City, University of London, UK. He is working as an Assistant Professor with the Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal. His research interests include Blockchain and Internet-of-Things security. He has published papers in reputed International conferences and journals.

K. Vivekananda Bhat received the Ph.D. degree in computer science and engineering from IIT Kharagpur, India, and the M.Tech. degree in systems analysis and computer applications from the National Institute of Technology Karnataka, Surathkal, India. He is currently an Associate Professor with the Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India. He has published several articles in reputed Web of Science indexed international journals and conferences. His research interests include cryptology and cyber security.

Muttukrishnan Rajarajan is currently a Professor of security engineering with the City, University of London, U.K., where he currently leads the Information Security Group. He is the Director of Institute for Cyber Security at City. He is a Visiting Researcher with the British Telecommunication's Security Research and Innovation Laboratory. His research interests include privacy-preserving data analytics, cloud computing, the Internet of Things security, and wireless networks. He has published well over 300 articles and continues to be involved in the editorial boards and technical programme committees of several international security and privacy conferences and journals. He is an Advisory Board Member of the Institute of Information Security Professionals, U.K., and acts as an advisor to the U.K. Government's Identity Assurance Programme.

Ashok Kumar Das received a Ph.D. degree in computer science and engineering, an M.Tech. degree in computer science and data processing, and an M.Sc. degree in mathematics from IIT Kharagpur, India. He is currently an Associate Professor with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad, India. He also worked as a visiting faculty with the Virginia Modeling, Analysis and Simulation Center, Old Dominion University, Suffolk, VA 23435, USA. His current research interests include cryptography, system and network security, security in vehicular ad hoc networks, smart grids, smart homes, Internet of Things (IoT), Internet of Drones, Internet of Vehicles, Cyber-Physical Systems (CPS) and cloud computing, intrusion detection, blockchain and AI/ML security. He has authored over 310 papers in international journals and conferences in the above areas, including over 265 reputed journal papers. He was a recipient of the Institute Silver Medal from IIT Kharagpur. He has been included in the subject-wise ranking of top 2% scientists from India (all fields) in the area of Networking & Telecommunications, with Rank world-wide (by subject area): 321. He has been also listed in the top H-Index for Scientists in the World for Computer Science database maintained by Research.com (https://research.com/scientists-rankings/computer-science/in) with World rank: 1650 and National (India) rank: 8. He is/was on the editorial board of IEEE Systems Journal, Journal of Network and Computer Applications (Elsevier), Computer Communications (Elsevier), Journal of Cloud Computing (Springer), Cyber Security and Applications (Elsevier), IET Communications, KSII Transactions on Internet and Information Systems, and International Journal of Internet Technology and Secured Transactions (Inderscience), and has served as a Program Committee Member in many international conferences. He severed as one of the Technical Program Committee Chairs of the first International Congress on Blockchain and Applications (BLOCKCHAIN'19), Avila, Spain, June 2019, International Conference on Applied Soft Computing and Communication Networks (ACN'20), October 2020, Chennai, India, and second International Congress on Blockchain and Applications (BLOCKCHAIN'20), L'Aquila, Italy, October 2020. He is also serving as an advisory board member of 3rd International Congress on Blockchain and Applications (BLOCKCHAIN '21), Salamanca, Spain, 6-8 October, 2021. His Google Scholar h-index is 67 and i10-index is 193 with over 12,850 citations. He is a senior member of the IEEE.

Joel J. P. C. Rodrigues is a professor at the College of Computer Science and Technology, China University of Petroleum (East China), Qingdao, China; and senior researcher at the Instituto de Telecomunicações, Portugal. Prof. Rodrigues is the leader of the Next Generation Networks and Applications research group (CNPq), Director for Conference Development - IEEE ComSoc Board of Governors, IEEE Distinguished Lecturer, Technical Activities Committee Chair of the IEEE ComSoc Latin America Region Board, the President of the scientific council at ParkUrbis - Covilhã Science and Technology Park, a Past-Chair of the IEEE ComSoc Technical Committee on eHealth, a Past-chair of the IEEE ComSoc Technical Committee on Communications Software, Steering Committee member of the IEEE Life Sciences Technical Community and Publications co-Chair, and Member Representative of the IEEE Communications Society on the IEEE Biometrics Council. He is the editor-in-chief of the International Journal on E-Health and Medical Communications and editorial board member of several high-reputed journals. He has been general chair and TPC Chair of many international conferences, including IEEE ICC, IEEE GLOBECOM, IEEE HEALTHCOM, and IEEE LatinCom. He has authored or coauthored over 1000 papers in refereed international journals and conferences, 3 books, 2 patents, and 1 ITU-T Recommendation. He had been awarded several Outstanding Leadership and Outstanding Service Awards by IEEE Communications Society and several best papers awards. Prof. Rodrigues is a member of the Internet Society, a senior member ACM and a fellow of IEEE.

Authors and Affiliations

N. Sivaselvan[1,2] · K. Vivekananda Bhat[3] · Muttukrishnan Rajarajan[1] · Ashok Kumar Das[4] · Joel J. P. C. Rodrigues[5,6]

✉ K. Vivekananda Bhat
kv.bhat@manipal.edu

✉ Ashok Kumar Das
iitkgp.akdas@gmail.com; ashok.das@iiit.ac.in

N. Sivaselvan
siva.selvan@manipal.edu; sivaselvan.natarajan@city.ac.uk

Muttukrishnan Rajarajan
r.muttukrishnan@city.ac.uk

Joel J. P. C. Rodrigues
joeljr@ieee.org

[1] Department of Electrical and Electronic Engineering, City University of London, London, UK

[2] Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, India

[3] Department of Computer Science and Engineering and Centre for Cryptography, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, India

[4] Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500032, India

[5] College of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266555, China

[6] Instituto de Telecomunicações, 6201-001 Covilhã, Portugal