Journal of Information Security and Applications 53 (2020) 102503

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

Multi-Authority CP-ABE-Based user access control scheme with constant-size key and ciphertext for IoT deployment

Soumya Banerjee[a], Sandip Roy[b], Vanga Odelu[c], Ashok Kumar Das[d,*], Samiran Chattopadhyay[a,h], Joel J. P. C. Rodrigues[e,f], Youngho Park[g]

[a] Department of Information Technology, Jadavpur University, Salt Lake City, Kolkata 700 098, India
[b] Department of Computer Science and Engineering, Asansol Engineering College, Asansol 713 305, India
[c] Department of Computer Science and Engineering, Indian Institute of Information Technology Sri City, Andhra Pradesh 517 588, India
[d] Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500 032, India
[e] Federal University of Piauí (UFPI), 64049-550 Teresina-Pi, Brazil
[f] Instituto de Telecomunicações, 1049-001 Lisbon, Portugal
[g] School of Electronics Engineering, Kyungpook National University, Daegu 41566, Republic of Korea
[h] Northumbria University, Newcastle-upon-Tyne, NE1 8ST, UK

ARTICLE INFO

Keywords:
Internet of things (IoT)
User access control
Attribute-based encryption
Multi-authority ABE
Security
AVISPA

ABSTRACT

With the ever-increasing rate of adoption of internet-enabled smart devices, the allure of greater integration of technologies, such as smart home, smart city, and smart grid into everyday life is undeniable. However, this trend inevitably leaves a massive amount of information and infrastructure connected to the public Internet, which exposes the data to many security threats and challenges. In this paper, we discuss the need for fine-grained user access control for IoT smart devices. The inherently distributed nature of IoT environment necessitates the support of multi-authority attribute-based encryption (ABE) for the implementation of fine-grained access control. Therefore, we present a secure fine-grained user access control scheme for data usage in the IoT environment. The proposed scheme is a three-factor user access control scheme, which supports multi-authority ABE and it is highly scalable as both the ABE key size stored in the user's smart card and ciphertext size needed for authentication request are constant with respect to the number of attributes. Through the formal and informal security analysis, we show that the proposed scheme is secure and robust against several potential attacks required in an IoT environment. Moreover, we demonstrate that the proposed scheme performs at par or better than existing schemes while providing greater functionality features.

1. Introduction

Internet of Things (IoT) refers to a heterogeneous network which comprises of numerous devices that directly or indirectly enable exchange of information over the public Internet [1]. These devices include a wide range of spectrum, such as Radio-Frequency Identification (RFID) tags, common smartphones, and internet-enabled consumer appliances. By the year 2020, it is estimated that the number of IoT devices will reach fifty billion [2]. We also expect that IoT devices be smart enough so that they can work without any human intervention [3]. Therefore, the objective of IoT is as follows [3]: "to provide a strong interaction between the physical world and computer-based systems that can provide improvements in the economic welfare, and accuracy and efficiency while minimizing human participation". The promise of ubiquitous connectivity has powered an unprecedented growth in the adoption of IoT devices, which in turn has exposed the associated security and privacy challenges. IoT promises great economic prospects held by several threats to security and privacy [4,5].

As a large amount of data is generated by the IoT smart devices, it is extremely valuable to well-analyze data. However, the large-scale deployment of IoT incurs new challenges, and IoT security is an important aspect of the IoT environment. IoT architecture provided in Fig. 1 dictates that data be invariably transmitted through heterogeneous networks. To ensure the integrity of the sensitive and private data transmission over potentially insecure networks (i.e., internet), security solutions should be provided in the IoT environment (for example, encryption and access control

* Corresponding author.
E-mail addresses: soumyaBanerjee@outlook.in (S. Banerjee), sandiproy9500@gmail.com (S. Roy), odelu.vanga@iiits.in (V. Odelu), iitkgp.akdas@gmail.com, ashok.das@iiit.ac.in (A.K. Das), samirancju@gmail.com (S. Chattopadhyay), joeljr@ieee.org (J.J. P. C. Rodrigues), parkyh@knu.ac.kr (Y. Park).

https://doi.org/10.1016/j.jisa.2020.102503
2214-2126/

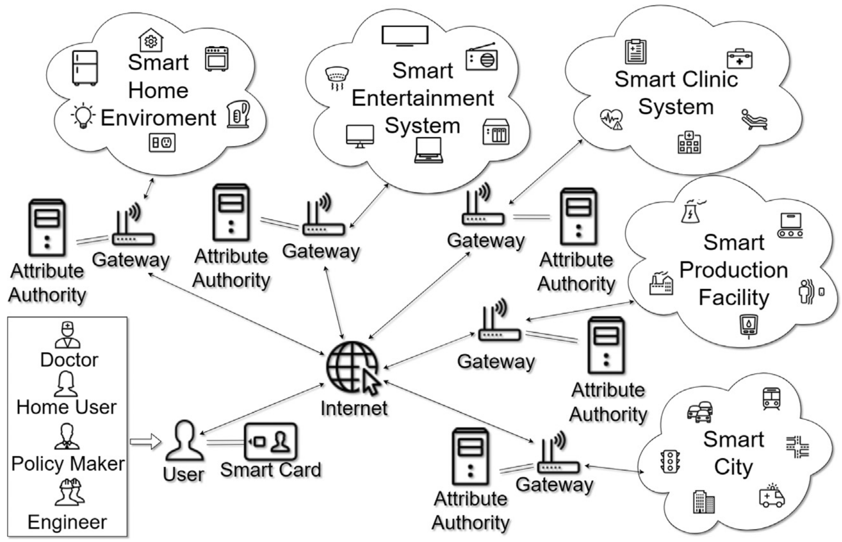

Fig. 1. An IoT architecture suitable for fine grained access control.

mechanisms need to be integrated into IoT deployment) [6]. Kolias et al. discussed how the Mirai botnet and its variants affect the IoT infrastructure [7]. The ubiquitousness of IoT devices allows compromised devices to act as an attack amplifier.

Mirai executes "distributed denial of service (DDoS)" against target servers by constantly compromising a greater number of publicly accessible and weakly configured IoT devices. After Mirai's source code was made publicly available, new variants emerge faster than the deployment of counter measures. This event highlights the need for better security practices to be embedded into the design of the IoT devices. Furthermore, there arise some situations where not all users should have access to all resources. For example, in a smart medicare system, a user (i.e., a doctor) should not be able to access the database of financial information, or an accountant will not be given access to medical histories. Looking up access privileged every time a user requests access is cumbersome and inefficient. Thus, it is often desirable to incorporate fine-grain access control mechanism into the IoT environment. With attribute-based encryption, it is possible to realize a fine grain access control mechanism into IoT environment.

Attribute-based encryption (ABE) was first introduced in a paper titled fuzzy identity-based encryption (FIBE) by Sahai and Waters [8]. Goyal et al. [9] improved upon the concept presented in FIBE and presented ABE as a cryptographic primitive, which realizes an access control mechanism (see Section 4.1). The ABE scheme proposed by Goyal et al. was a Key-Policy Attribute-based encryption (KP-ABE). In KP-ABE, the access policy is encoded into the user's secret key, and the ciphertext is encrypted with respect to the attributes. Bethencourt et al. [10] presented Ciphertext-Policy Attribute-Based Encryption (CP-ABE), where the access policy can be defined during encryption. CP-ABE is more versatile in the sense that who can decrypt an encrypted text can be specified during the encryption step itself. In KP-ABE, who can decrypt a specific message is bound by the user key, consequently must be defined during the key issue. In the context of the IoT environment, the key issue step will likely coincide with user registration. But IoT environment must support dynamic user registration and device enrollment, and the access to the individual smart devices is governed the user's ability to decrypt messages encrypted by the ABE scheme. Thus, it is desirable that access permission to smart devices, instead of being grandfathered in for already existing users, be set up during the device enrollment. This necessitates that fine-grain access is provided through a CP-ABE mechanism.

In ABE protocols, the key and ciphertext sizes are proportional to the number of attributes in the universe. For an IoT environment, the number of attributes in the universe is expected to be huge. The ciphertext size will affect the size of the messages exchanged during authentication and session key establishment. Similarly, the key size affects the space requirement of the user smart card. Since these values must be minimized for any well-designed authentication or access control schemes for IoT, a fine-grained access control scheme for IoT environments cannot be scalable if the key and ciphertext sizes are proportional to the number of attributes in the universe. Attrapadung et al. [11] proposed a KP-ABE scheme with constant size ciphertext. Keita et al. [12] and Yinghui *et a.* [13] independently proposed CP-ABE schemes with constant ciphertext length. Guo et al. [14] also designed an CP-ABE scheme with constant size keys. Odelu et al. [15] presented another CP-ABE scheme for cloud computing that supports both constant size key and ciphertext.

For IoT environment, the users generally need to access different IoT subsystems. From an ABE perspective, it is desirable to account for attributes defined by the external subsystems. For example, consider two subsystems, namely a smart home and a smart medicare system. In the smart home, assume there is a smart first aid kit device untrusted doctor nor a trusted, but un-qualified person should make alterations to the device. The *AUTHORIZED_BY_RESIDENT* attribute is defined in the smart home system, but the attribute *DOCTOR* is an attribute defined in the smart medicare system. The solution to such a problem leads to the multi-authority ABE concept. Chase and Chow [16] proposed an improved multi-authority extension to the Key policy Attribute-Based Encryption (KP-ABE) scheme proposed by Sahai and Waters [8].

Access control is enforced after authentication. A fine-grained access control scheme for IoT environment that incorporates multi-factor authentication relives the need for two independent security layer for the already computationally limited IoT infrastructure. It has been observed that schemes relying on two-factor authentications generally fall vulnerable to password guessing attacks in presences of a privileged insider armed with a stolen smart card [17]. A well designed three-factor authentication scheme can withstand the discussed attacks.

In this paper, we propose a highly scalable fine-grained anonymous three-factor user access control scheme for IoT environment with support for multiple attribute authorities. The proposed scheme is supported by an underlying ABE that has constant-size key and ciphertext along with the policy hidden CP-ABE.

1.1. Research contributions

The main research contributions of this paper are highlighted below.

- An extension to the IoT architecture has been presented in order to make it suitable for fine-grained access control.
- A fine-grained anonymous user access control scheme with three-factor authentication for IoT environment has been proposed.
- The proposed scheme has been designed to satisfy the problem statement defined in Section 3.4.
- A mathematical proof of correctness for the underlying ABE scheme has been provided.
- The formal security analysis under the Real-Or-Random (ROR) model [18] proves the session key security of the proposed scheme.
- The scheme has been simulated under the widely accepted formal security verification tool, AVISPA [19], to verify its resistance against replay and man-in-the-middle attacks. Additionally, informal security analysis has been undertaken to demonstrate the proposed scheme's resistance to various well-known attacks against (passive/active) adversary.
- A comparative study has been performed to summarize the functional features of ABE schemes and another one on communication and computational overheads analysis as well as security and functionality features among the authentication and access control schemes.
- Finally, the practical implementation of the proposed scheme has been carried out under the NS3 simulation environment in order to measure the network impact of the proposed scheme.

1.2. Paper outline

Section 2 summarizes the related research. Section 3 defines the architecture and problem statement. Section 4 briefly overviews the relevant mathematical background that is utilized in this paper. Section 5 presents the proposed scheme with its various phases. The mathematical proof of correctness for the underlying ABE scheme with respect to the proposed scheme is provided in Section 6. This section also includes formal as well as informal security analysis of the presented scheme. Moreover, the results of the AVISPA simulation are presented in this section. The comparative study with related schemes and practical perspective of the presented scheme through NS3 simulation study are also provided in Sections 7 and 8, respectively. Finally, Section 9 concludes the work.

2. Related work

In this section, we provide a brief overview of the security challenges associated with IoT infrastructure as well as a summary of the application of ABE in related domains.

Mineraud et al. [20] observed that malware, as well as the presence of inherent design flaws, opened huge security challenges for the upcoming IoT infrastructure. Alqasen [21] explained that due to the diverse heterogeneous nature of IoT, the security challenges need to be investigated and solved with a specific focus on IoT architecture. An authentication scheme needs to guaranty anonymous mutual authentication while resisting known attacks like denial of service, man-in-the-middle, to name a few [22]. Additionally, such a scheme needs to solve some issues specific to IoT environments like the issues of dynamic smart device addition, the capture of unattended smart devices, among others. Schemes, while using already proposed and proved cryptographic primitives, need to specifically verify that their application together does not expose any unforeseen attack surface.

Jeong et al. [23] investigated the user authentication problem in smart homes, which is an application of IoT. They proposed a solution based on a One-Time Password (OTP). Unfortunately, their scheme not only fails to assure mutual authentication but also lacks user anonymity and untracability properties. For ubiquitous computing devices, Hunumanathappa et al. [24] advocated a three-way user authentication scheme which used pass-phrases to guaranty device attestation. Santoso et al. [25] utilized Elliptic Curve Cryptography (ECC) technique to build a user authentication scheme for smart homes. Unfortunately, their scheme also fails to provide anonymity and untracability properties.

Challa et al. [26] presented an ECC signature-based user authentication scheme for IoT applications that is secure against several well-known attacks. However, their scheme incurs more computational overhead as compared to other schemes. Zhou et al. [27] presented a lightweight anonymous user authentication scheme using only one-way hash and bitwise XOR operations. Banerjee et al. [28] proposed a physically secure lightweight user authentication scheme for IoT that utilized physically unclonable functions (PUF) to provide resistance against stolen device impersonation attacks.

Shahzad and Singh [29] discussed how a continuous authentication can preclude the abuse of IoT devices. However, their solution mainly relies on the devices to maintain permanent physical contact with the user. For devices that do not maintain physical contact with the user, they proposed some surrogates which effectively require additional overhead and can be considered invasive. Chuang et al. [30] also presented an alternative approach for continuous authentication. They advocated a two-phase solution, where a periodic comparatively expensive, static authentication session generates a token, which is then used by subsequent lightweight continuous authentication sessions.

However, in these schemes, the fine-grained access control is not supported. Few fine-grained access control schemes have been proposed for IoT or wireless sensor network (WSN) architecture. The main challenge in designing such a scheme is the mitigation of the issue of limited computation capability of the smart device or the sensor node contrasting with the inherently computationally expensive calculations associated with ABE. As Turkanovic et al. [31] integrated WSN into the IoT environment, we describe some access control schemes in the context of fine-grained access control in the IoT environment.

Authors in [32] proposed an anonymous authentication scheme for global mobility networks (GLOMONET) where they highlighted that under the currently accepted model for anonymous authentication, the users is not anonymous to the registration server. They discussed how this level on anonymity might not be sufficient under all circumstances. The authors presented a user-group based construction that during authentication, for a group with k members, provided $k - anonymity$, while cryptographically guaranteeing the group membership of the user. The proposed scheme was a user authentication scheme and as such did not support any fine grained access control mechanism. But the issue raised in this paper can be solved more efficiently through a fine grained access control.

He et al. [33] presented privacy-preserving access control scheme based on ring signature for multi-user WSNs. But as it utilized ring signature as the cryptographic primitive, it could not provide fine-grained access control. Yu et al. [34] presented the first fine-grained access control scheme for distributed WSNs. The scheme was implemented as a KP-ABE based on bilinear pairing groups in ECC. Under the scheme, the base station gathered the data form sensor node and encrypted with ABE, which a user with matching attributes could decrypt. Ruj et al. [35] observed that under the scheme proposed in [34], any change in the user's access structure (due to change in his/her attributes) required the issue of a new user key. They proposed an improvement, such that a user key can be modified to accommodate the user's changed attributes. The scheme in [35] was designed to accommodate multiple attribute authorities.

Chatterjee and Roy [36] cryptanalyzed the schemes presented in [34] and [35] to show that both schemes are vulnerable to insider attacks where users with lower privilege could access restricted data. Chatterjee and Das [37] proposed a KP-ABE based fine grained user access control scheme for WSNs resistant to insider attacks.

Several authors have investigated fine-grained grained access control for cloud computing architecture. While capability wise IoT smart devices are not at all comparable to cloud computing infrastructures, cloud computing architecture has similarity with IoT architecture in terms of the distributed nature and oftentimes requires multiple attribute authorities.

Lounis et al. [38] utilized CP-ABE to present a cloud-based architecture for secure dissemination of medical sensor data. He et al. [39] proposed, a lightweight fine-grained access control scheme for WSN-integrated cloud computing, that utilizes the proxy services on cloud to execute the computationally heavy ABE computation. This is computation outsourcing, where part of the ABE computation is off-loaded onto the assisting cloud.

Liu et al. [40] designed a "fine-grained two-factor authentication access control system for web-based cloud computing services". In their scheme, an attribute-based access control mechanism has been implemented due to the requirements of both a secret user key and a security device. Moreover, under this scheme, the users who satisfy an access policy is known only to the server. Thus the scheme is considered policy hidden. Li et al. [41] extended the multi-authority ABE scheme proposed by Chase and Chow [16] in order to provide fine-grained access control with accountability for cloud applications.

Belguith et al. [42] presented a mechanism, called PHOABE, which is a policy hidden multi authority ABE for cloud-assisted IoT. One of the primary advantages of PHOABE is that the computational overhead of ABE computation is off-loaded onto the assisting cloud. PHOABE is a very versatile ABE scheme, but it requires outsourcing computation to the assisting cloud to minimize computation overhead on IoT devices. But, the availability of such a cloud is conditional. Moreover, none of the schemes discussed have constant size ciphertexts and secret keys, which, as we discussed previously, is necessary for scalability in IoT architecture. Moreover, the majority of the discussed schemes described do not present a complete session key agreement mechanism, and the authors in [36] demonstrated that an otherwise well designed ABE scheme, can be compromised during session key agreement phase.

Recently, the authors in [43] proposed a CP-ABE scheme for fine-grained access control in IoT architecture that doesn't rely on any cloud infrastructure and describes a complete session key agreement scheme. However, the presented scheme is limited to a single attribute authority, and it is also further limited in the sense that the smart card storage cost and the sizes of the messages are directly proportional to the number of attributes used in the system.

In this paper, we propose a fine-grained access control scheme with constant-size key and ciphertext suitable for IoT architecture without any assisting cloud infrastructure. Additionally, the proposed scheme multiple attribute authority and is policy hidden by its design.

3. Architecture and problem statement

In this section, we describe an IoT architecture, the associated trust, communication, and threat model, suitable for fine-grained access control. We also define the problem statement of this work and outline the contribution and the paper organization in this section.

3.1. System architecture

In this section, we discuss an IoT architecture shown in Fig. 1 that is suitable for fine-grained access control. Under this IoT architecture, multiple smart devices together form a smart environment in which the devices are connected to the internet through the gateway node(s). The registered users can access the services of the designated smart devices through the gateway node(s) after the authentication process is completed. It is worth noting that a user may have attributes defined under multiple smart environments at the same time.

In order to provide fine-grained access control in the described architecture shown in Fig. 1, it is needed to define how a user is eligible to access different smart devices. As discussed previously, a natural solution to deal with this problem is the used of CP-ABE. We envision an attribute authority associated with each gateway node, and the access policy $\mathbb{P}$ of a smart device can be defined during its enrollment process. Similarly, during the registration of a legitimate user, his/her access policy, $\mathbb{A}$ needs to be also defined. If $\mathbb{P} \subset \mathbb{A}$, a user can access the services provided by the smart device.

From Fig. 1, it is clearly apparent that a user can have different roles defined by attribute authorities from different smart networks. This demands that a set of attributes, and consequently, the access policies need to be defined globally. In the proposed architecture, a user is registered with any one of the gateway node(s) (also known as the attribute authorities), but the secret credentials should be composed with the help of all relevant (under which the user has the defined attributes) gateway node(s).

3.2. Trust and communication model

In this section, we define the trust and communication model under the IoT architecture shown in Fig. 1. We have different smart devices *Dev*, the associated gateway nodes *GWN* and attribute authority AA_k, and the users *U* with their respective smart cards SC_U. *GWN* and the associated attribute authority AA_k are considered trusted.

Enrollment of a smart device, *Dev*, with gateway node, *GWN*, establishes mutual trust between them. *U* trusts *GWN*, and *GWN*, in turn, has pre-existing trust in the smart card SC_U that it, in conjuncture with the relevant attribute authorities, has issued during the user's registration. Both *U* and *Dev* require to establish mutual trust among each other during the authentication phase. The mutual trust is established between *U* and *Dev* through their ability to successfully perform the ABE computation described in Section 5.6.

In terms of communication, *U, GWN*, AA_k and *Dev* can communicate with each other through the internet. In the case of *U* and *Dev*, the messages can be routed transparently through *GWN*. Fig. 2(a) and 2(b) visualize the trust and communication models, respectively under the proposed IoT architecture provided in Fig. 1.

3.3. Threat model

In this paper, we adapt the widely-used Dolev-Yao (DY) threat model [44] in which an adversary $\mathcal{A}$ has full control over the communication media. Thus, $\mathcal{A}$'s capability is not only interception but also modification or deletion of the messages. Furthermore, it is assumed that $\mathcal{A}$ can recover sensitive information from stolen/lost smart card of a legitimate user through power analysis attacks [45,46].

In addition, the current *de facto* standard, known as the Canetti and Krawczyk (CK) adversary model [47], is also adapted in the proposed scheme, where $\mathcal{A}$ has all capabilities of a DY adversary, and in addition he/she can subvert the private keys as well as the session states. This requires the assurance that the leakage of ephemeral secrets or session keys should have a minimal consequence on the security of unrelated sessions.

Authors in [48], [49] presented that unlike most individual smart devices, the gateway nodes can be physically secured from $\mathcal{A}$ by putting the gateway nodes in physical locking systems. Thus, the gateway nodes are considered as secure and trusted. However, we assume that some of the smart devices can be physically captured by $\mathcal{A}$. Consequently, $\mathcal{A}$ can recover all sensitive information stored in the captured smart devices. As a result, it is imperative that exposure of information stored in an individual smart device should not affect the security of any other entity in the IoT network.

3.4. Problem statement

In the previous sections of this work, we have highlighted the need for a fine-grained access control scheme for the IoT environment and observed the absence of such a scheme in the existing literature. In this section, we formalize the requirements fo such a scheme.

- The scheme should support fine-grained access control after three-factor authentication.
- The underlying ABE scheme should be a CP-ABE designed to support multiple attribute authorities.
- The scheme should be designed to operate within the resource-constrained IoT devices without the assistance of any cloud infrastructure.
- The scheme should be highly scalable; thus, the underlying ABE scheme should have constant-size key and ciphertext with respect to the number of attributes.
- The scheme should define a complete access control mechanism, including user registration, device enrollment, authentication, and session key establishment.
- The scheme should be anonymous and privacy-preserving and resistant to known attacks.

4. Mathematical preliminaries

In this section, we discuss the relevant mathematical background for the scheme proposed in this paper.

4.1. Attribute and access structures

We use an extended version of the attribute and access structures introduced in [15]. Assuming there are N attribute authorities and n attribute values to be defined among them, the universe of attributes is considered as $\mathbb{U} = \{A_1, A_2, \cdots, A_n\}$. An access structure $\mathbb{A} \subseteq \mathbb{U}$ is represented with n-bit string $a_1 a_2 \cdots a_n$, where $a_i = 1$ if $A_i \in \mathbb{A}$ and 0 if $A_i \notin \mathbb{A}$.

A user access structure $\mathbb{A}$ is comprised by aggregating all the access structures $\mathbb{A}_k = a_{1_k} a_{2_k} \cdots a_{n_k}$ from attribute authority AA_k, where $k \in [1, N]$. $\mathbb{A}_k$ describes the attributes controlled by attribute authority AA_k. A single attribute is controlled by a single attribute authority. An attribute A_i which is not controlled by attribute authorities AA_k, we have $a_{i_k} = 0$. Note that we can define $\mathbb{A} = \sum_{k=1}^{N} \mathbb{A}_k$ as $\mathbb{A}_j \cap \mathbb{A}_k = \phi$, where $j, k \in [1, N]$ and $j \neq k$.

In AND gate access policy, a smart device with access structure $\mathbb{P} \subseteq \mathbb{U}$, where $\mathbb{P} = b_1 b_2 \cdots b_n$, is accessible by a user with the access structure $\mathbb{A}$ if and only if $\mathbb{P} \subseteq \mathbb{A}$. In other words, for $\mathbb{A}$ to satisfy $\mathbb{P}$, the condition $(a_i - b_i) \geq 0$ must be satisfied, $\forall i \in [1, n]$.

4.2. Bilinear pairing

Let G_1 and G_2 be two elliptic groups, and G_T be a multiplicative group of prime order q. A bilinear pairing group $\mathbb{G} = \{G_1, G_2, G_T, g, h, e(\cdot)\}$ is defined with a bilinear pairing e: $G_1 \times G_2 \rightarrow G_T$ of prime order q which satisfies the following conditions [14]:

- $\forall g \in G_1$, $h \in G_2$, and $a, b \in Z_q$, $e(g^a, h^b) = e(g, h)^{ab}$.
- If g and h are the generators of G_1 and G_2, respectively, $e(g, h)$ is a generator of G_t.
- There exists an efficient algorithm to calculate $e(g, h)$, $\forall g \in G_1$, $h \in G_2$.

4.3. Pseudo-Random functions

In this paper, we utilize the pseudo-random functions described in [16] to independently generate user specific shared key (component). The authors in [16] adapted the pseudo-random function (PRF) proposed by Dodis et al. [50].

Two distinct attribute authorities, say AA_j and AA_k can use the complementary PRFs, $PRF_{jk}(\cdot)$ and $PRF_{kj}(\cdot)$, respectively. For a specific user with identity EID_U,

$$PRF_{jk}(EID_U) = g^{(\delta_{jk}.x_j.x_k/(s_{jk}+EID_U))},$$

where s_{jk} is a key pre-shared between AA_j and AA_k, and x_j and x_k are their private keys, respectively, and

$$\delta_{jk} = \begin{cases} 1, & \text{if } k > j \\ -1, & \text{otherwise} \end{cases}$$

Additionally, we define a pseudo-random number generator $RNG_{seed}(\cdot)$ based on the Advanced Encryption Standard (AES) symmetric-key encryption algorithm [51] can be used as a cryptographically secure random number generator. We utilize the counter mode of an 128-bit AES with the *seed* value as the key in order to define *RNG*.

4.4. Attribute-Based encryption

The underlying cipher text policy attribute based encryption (CP-ABE) scheme is an amalgamation of the KP-ABE scheme presented in [16], and the CP-ABE scheme presented in [15] consists of the following four phases.

- *Setup:* A security parameter ρ and the universe of attributes $\mathbb{U} = \{A_1, A_2, \cdots, A_n\}$ are supplied as inputs to produce a master secret and public keys pair (*MPK, MSK*) as output.
- *Encrypt:* It applies encryption function which takes an access policy $\mathbb{P}$, a plaintext message *Msg* and the public key *MPK* as inputs in order to have the output as a ciphertext *C*.

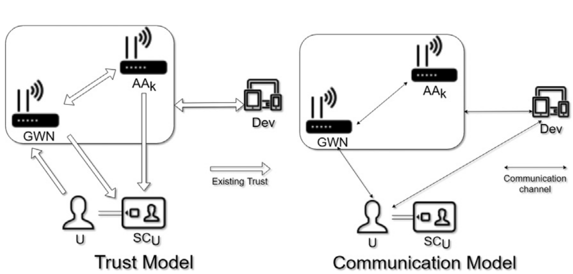

Fig. 2. (a) Trust model and (b) Communication model under IoT architecture.

- *KeyGen:* It is supplied with an attribute set $\mathbb{A}$, the master secret key *MSK* as well as public-key *MPK* to obtain the output as the secret user key (decryption key) k_u of $\mathbb{A}$.
- *Decrypt:* It applies decryption function which is supplied with a ciphertext C generated with $\mathbb{P}$, k_u corresponding to $\mathbb{A}$ and *MPK* as inputs to get the output as the original plaintext message *Msg* or $\perp$ (null).

We now present a concrete construction for the underlying ABE scheme as follows. In our proposed scheme discussed in Section 5, there is a central controller and N attribute authorities. $\mathbb{U} = \{A_1, \cdots, A_n\}$ denotes the universe of attributes that can be represented by the string $\mathbb{U} \to \{0,1\}^n$. Let $\mathbb{A}_k = a_{1_k} \cdots a_{n_k}$ describes the attributes that are controlled by the attribute authorities k such that $\mathbb{A}_k \subseteq \mathbb{U}$, $\mathbb{A}_j \cap \mathbb{A}_k = \phi$, where $j, k \in [1, N]$ and $j \neq k$. If an attribute A_i is not controlled by the attribute authorities k, we then make $a_{i_k} = 0$.

The following algorithms are described below.

4.4.1. Setup

A security parameter along with $\mathbb{U}$ are supplied as inputs to *Setup* algorithm which outputs the master key pair, say {*MSK*, *MPK*}. The detailed steps are given below.

- The controller picks a "bilinear pairing group", say $\mathbb{G} = \{G_1, G_2, G_T, p\}$ with generators $g \in G_1$ and $h \in G_2$ and defines a bilinear pairing $e(\cdot)$ such that $e(g^a, h^b) = e(g, h)^{ab}$ $\forall a, b \in \mathbb{Z}_p$.
- The controller then picks four "collision resistant cryptographic one-way hash functions" $H_1, H_4 : \{0,1\}^* \to \mathbb{Z}_p^*$, $H_2 : \{0,1\}^* \to \{0,1\}^{l_\sigma}$, $H_3 : \{0,1\}^* \to \{0,1\}^{l_m}$, where l_σ and l_m are the lengths defined by the security parameter and plaintext message, say M, respectively.
- Each attribute authority k randomly picks $\alpha_k, x_k \in \mathbb{Z}_p$ and shares $Q_k = e(g,h)^{\alpha_k}$, $y_k = g^{x_k}$ with all other authorities. Each authority also computes the "system public key" as $\alpha = H_1(\Pi_{k=1}^{N} Q_k) = H_1(e(g,h)^{\sum_{k=1}^{N} \alpha_k})$.
- Each pair of authorities, say j and k, agree upon a secret seed value, say s_{jk}. By definition, it follows that $s_{jk} = s_{jk}$.
- Each authority k then defines $PRF_{jk}(\cdot)$ as $PRF_{jk}(u) = g^{\delta_{jk} \cdot x_j \cdot x_k/(s_{jk}+u)}$, where $j, k \in [1, N]$ and $\delta_{jk} = {}^{\ 1 \text{ if } k>j}_{-1 \text{ otherwise}}$. Only the authorities j and k can compute this as it is same to $y_k^{x_j/(s_{jk}+u)}$ $= y_j^{x_k/(s_{jk}+u)}$.
- The controller randomly picks $K_1, K_2 \in \mathbb{Z}_p$ and calculates $h_i = h^{\alpha^i}$, $u_i = h^{K_1 \alpha^i}$ and $v_i = h^{K_2 \alpha^i}$ $\forall i \in [1, n]$.
- Finally, the controller publishes $MPK = \{\mathbb{G}, e(\cdot), g, h, g^\alpha, \{h_i, u_i, v_i\}_{\forall i \in [1, n]}, H_1, H_2, H_3, H_4\}$ and $MSK = \{\alpha, K_1, K_2\}$.

4.4.2. Encrypt

The *Encrypt* algorithm is supplied with an access policy $\mathbb{P} \subseteq \mathbb{U}$, *MPK* and a plaintext message M as inputs, and the ciphertext $\{\mathbb{P}, R_m, K_{1m}, K_{2m}, C_{\sigma_m}, C_m\}$ is produced as output with the following steps:

- An encryptor randomly picks $\sigma_m \in \{0,1\}^{l_\sigma}$ and calculates $r_m = H_1(\mathbb{P}, M, \sigma_m)$ and $e(g,h)^{r_m}$.
- The encryptor also defines the access policy as $\mathbb{P} = b_1, \cdots, b_n$ and calculates atmost n-degree polynomial $f(x, \mathbb{P})$ defined as

$$f(x, \mathbb{P}) = \prod_{i=1}^{n} (x + H_4(i))^{1-b_i}, \tag{1}$$

where f_i is the coefficient of x^i.
- The encryptor also calculates the following:
$R_m = (g^\alpha)^{r_m} = g^{\alpha r_m}$,
$K_{1m} = (\prod_{i=0}^{n} (u_i^{f_i}))^{r_m} = h^{K_1 f(\alpha, \mathbb{P}) r_m}$,
$K_{2m} = (\prod_{i=0}^{n} (v_i^{f_i}))^{r_m} = h^{K_2 f(\alpha, \mathbb{P}) r_m}$,
$C_{\sigma_m} = H_2(e(g,h)^{r_m}) \oplus \sigma_m$,
and $C_m = H_3(\sigma_m) \oplus M$.
- Finally, the encryptor produces the ciphertext as $C = \{\mathbb{P}, R_m, K_{1m}, K_{2m}, C_{\sigma_m}, C_m\}$.

4.4.3. Keygen

The inputs to the *KeyGen* algorithm are the set access policy $\mathbb{A}_k = a_{1_k} \cdots a_{n_k}$ $\forall k \in [1, n]$ and the master key pair {*MSK*, *MPK*}. The output of this algorithm is the k_u, which is the secret key of user u. It is worth noticing that $A = \sum_{k=1}^{N} \mathbb{A}_k = a_1 \cdots a_n$ is the actual (consolidated) access policy of the u. The following steps are involved here:

- The controller first picks a random number $r_u \in \mathbb{Z}_p$.
- Each attribute authority $k \in [1, N]$ defines an access policy $\mathbb{A}_k = a_{1_k} \cdots a_{n_k}$ for the user u and calculates

$$f(\alpha, \mathbb{A}_k) = \prod_{i=1}^{n} (x + H_4(i))^{1-a_{i_k}}, \tag{2}$$

$$s_{u_k} = \left(\frac{1}{K_1}\right)^{(1/N)} \frac{\prod_{j=1, j\neq k}^{N} PRF_{jk}(r_u)}{f(\alpha, \mathbb{A}_k)},$$

and shares $g^{s_{u_k}}$ with the controller.
- The controller also calculates $s_u = \left(\prod_{k=1}^{N} s_{u_k}\right) - \left(\frac{K_2 r_u}{K_1}\right)$
$= \left(\prod_{k=1}^{N} \left(\frac{1}{K_1}\right)^{(1/N)} \frac{\prod_{j=1, j\neq k}^{N} PRF_{jk}(r_u)}{f(\alpha, \mathbb{A}_k)}\right) - \left(\frac{K_2 r_u}{K_1}\right)$
$= \left(\prod_{k=1}^{N} \left(\frac{1}{K_1}\right)^{(1/N)} \frac{\prod_{j=1, j\neq k}^{N} PRF_{jk}(r_u)}{\prod_{i=1}^{n} (x+H_4(i))^{1-a_{i_k}}}\right) - \left(\frac{K_2 r_u}{K_1}\right)$
$= \left(\left(\frac{1}{K_1}\right) \frac{\prod_{k=1}^{N} \prod_{j=1, j\neq k}^{N} PRF_{jk}(r_u)}{\prod_{k=1}^{N} \prod_{i=1}^{n} (x+H_4(i))^{1-a_{i_k}}}\right) - \left(\frac{K_2 r_u}{K_1}\right)$
$= \left(\left(\frac{1}{K_1}\right) \frac{\prod_{j,k=1, j\neq k}^{N} PRF_{jk}(r_u)}{\prod_{i=1}^{n} (x+H_4(i))^{1-\sum_{k=1}^{N} a_{i_k}}}\right) - \left(\frac{K_2 r_u}{K_1}\right).$
Since by the design, $\sum_{j,k=1, j\neq k}^{N} PRF_{jk}(\cdot) = 1$ and $\mathbb{A}_k \mid k \in [1, N]$ are disjoint subsets of $\mathbb{U}$, it follows that
$\sum_{k=1}^{N} a_{i_k} = a_i$,
$s_u = \frac{1}{K_1}\left(\frac{1}{\prod_{i=1}^{n} (x+H_4(i))^{1-a_i}} - K_2 r_u\right)$
$= \frac{1}{K_1}\left(\frac{1}{f(\alpha, \mathbb{A})} - K_2 r_u\right).$
The user secret key is $k_u = \{g^{r_u}, g^{s_u}\}$.

4.4.4. Decrypt

The inputs to the *Decrypt* algorithm is taken as the ciphertext $\{\mathbb{P}, R_m, K_{1m}, K_{2m}, C_{\sigma_m}, C_m\}$ and the secret key K_u with access structure $\mathbb{A}$. The output is then a plaintext message M or null ($\perp$). The following steps are involved in this process:

- The decryptor checks if $\mathbb{A}$ satisfies $\mathbb{P}$, i.e., $\mathbb{A} \subseteq \mathbb{P}$. If it returns $\perp$, the decryptor terminate. Otherwise, the decryptor calculates the at most $n- \mid \mathbb{P} \mid$-degree polynomial $F(x, \mathbb{A}, \mathbb{P})$ defined by

$$F(\alpha, \mathbb{A}, \mathbb{P}) = \prod_{i=1}^{n-|\mathbb{P}|} (x + H_4(i))^{a_i - b_i}, \tag{3}$$

where F_i is the coefficient of x^i and $F_0 \neq 0$.
- The decryptor also calculates the following:
$W = e\left(R_m, \prod_{i=1}^{n-|\mathbb{P}|} (h_{i-1})^{F_i}\right),$
$= e\left(g^{\alpha r_m}, \prod_{i=1}^{n-|\mathbb{P}|} h^{a^{i-1} F_i}\right),$
$= e(g,h)^{\alpha r_m \sum_{i=1}^{n-|\mathbb{P}|} a^{i-1} F_i}$

$$
\begin{aligned}
&= e(g,h)^{r_m \sum_{i=1}^{n-|\mathbb{P}|}(\alpha^i F_i + r_m F_0 - r_m F_0)},\\
&= e(g,h)^{r_m F(\alpha) + r_m F_0},\\
U &= e(g^{s_u}, K_{1m}) = e(g,h)^{K_1 f(\alpha,\mathbb{P}) r_m s_u)},\\
V &= e(g^{r_u}, K_{2m}) = e(g,h)^{K_2 f(\alpha,\mathbb{P}) r_m r_u)},\\
UV &= e(g,h)^{K_1 f(\alpha,\mathbb{P}) r_m s_u) + K_2 f(\alpha,\mathbb{P}) r_m r_u}\\
&= e(g,h)^{r_m f(\alpha,\mathbb{P})(K_1 s_u + K_2 r_u)}\\
&= e(g,h)^{r_m \frac{f(\alpha,\mathbb{P})}{f(\alpha,\mathbb{A})}}\\
&= e(g,h)^{r_m F(\alpha)},\\
\left(\frac{UV}{W}\right)^{\frac{1}{F_0}} &= \left(\frac{e(g,h)^{r_m F(\alpha)}}{e(g,h)^{r_m F(\alpha) + r_m F_0}}\right)^{\frac{1}{F_0}} = e(g,h)^{r_m},\\
\sigma_{m'} &= H_2(e(g,hj)^{r_m}) \oplus C_{\sigma_m},\\
M' &= C_m \oplus H_3(\sigma_{m'}),\\
r_{m'} &= H_1(\mathbb{P}, M', \sigma_{m'}).
\end{aligned}
$$

- Finally, the decryptor verifies whether the condition: $e(g,h)^{r_m} \equiv e(g,h)^{r_{m'}}$ satisfies or not. If it is not valid, the output is $\perp$. Otherwise, the output is the plaintext message M.

4.5. Correctness of the ABE scheme

In the following, we validate the correctness of the underlying ABE scheme in our proposed scheme by considering a scenario having N attribute authorities with a total of n attributes. Note that, $\prod_{i=1,j=1,j,i\neq j}^{N} PRF_{ij}(\cdot) = 1$, $a_i = \sum_{k=1}^{n} a_{ik}$ and $S_{U_i} = (\frac{1}{K_1})^{\frac{1}{3}} \frac{\prod_{j=1,j\neq i}^{N} PRF_{ij}(r_U)}{f(\alpha,\mathbb{A}_i)}$.

Therefore,

$$
\begin{aligned}
\prod_{k=1}^{N} s_{U_k} &= \frac{1}{K_1}.\frac{\prod_{i=1,j=1,j,i\neq j}^{N} PRF_{ij}(r_U)}{\prod_{i=1}^{N} f(\alpha,\mathbb{A}_i)}\\
&= \frac{1}{K_1}.\frac{1}{\prod_{k=1}^{N} f(\alpha,\mathbb{A}_k)}\\
&= \frac{1}{K_1}.\frac{1}{\prod_{k=1}^{N}\prod_{i=1}^{n}(\alpha + H_3(i))^{1-a_{ik}}}\\
&= \frac{1}{K_1}.\frac{1}{\prod_{i=1}^{n}(\alpha + H_3(i))^{1-\sum_{k=1}^{N} a_{ik}}}\\
&= \frac{1}{K_1}.\frac{1}{\prod_{i=1}^{n}(\alpha + H_3(i))^{1-a_i}}\\
&= \frac{1}{K_1}.\frac{1}{f(\alpha,\mathbb{A})}
\end{aligned}
$$

The *GWN* then computes $s_U = \prod_{k=1}^{N} s_{U_k} - \frac{K_2 r_U}{K_1} = \frac{1}{K_1}.(\frac{1}{f(\alpha,\mathbb{A})} - K_2 r_U)$. The user secret key is (g^{r_U}, g^{s_U}). This construction shows that the proposed MA-CP-ABE can be reduced to the CP-ABE proposed in the existing scheme [15]. It is also clear to see that s_U cannot be represented as any combination of s_{U^1} and s_{U^2}, for the users U, U^1 and U^2, even if $\mathbb{A} \subset \mathbb{A}^1 \cup \mathbb{A}^2$. Consequently, U^1 and U^2 cannot collude and calculate (g^{r_U}, g^{s_U}), as the user u's key from their keys. This means that the underlying ABE scheme proposed in this paper is collusion-resistant, and its correctness is also verified.

4.6. Selective game for CP-ABE scheme

We describe the *security game* [14,15] involving an adversary $\mathcal{A}$ and a challenger $\mathcal{B}$ that can capture the indistinguishability of messages and the collusion-resistance of user secret keys as follows.

- **Initialization:** $\mathcal{A}$ outputs an n-bit challenge access policy $\mathbb{P}'$ and transmits it to $\mathcal{B}$.
- **Setup:** $\mathcal{B}$ then runs the *Setup* under the security parameter ρ and generates the key master pair (*MPK, MSK*). After that $\mathcal{B}$ supplies the master public-key *MPK* to $\mathcal{A}$.
- **Query:** $\mathcal{A}$ makes the following queries to $\mathcal{B}$:
  - $\mathcal{A}$ queries the secret key k_{u^i} of any attribute set $\mathbb{A}^i$.
  - $\mathcal{A}$ queries the decryption of ciphertext $Enc(\mathbb{P}^i, M^i)$.
- **Challenge:** $\mathcal{A}$ outputs (M_0, M_1) for the challenge. Note that $\mathcal{A}$ does not query for the secret key of an attribute set $\mathbb{A}$ satisfying the relation: $\mathbb{P}' \subseteq \mathbb{A}$. Now, $\mathcal{B}$ picks a random $c' \in \{0, 1\}$, and then responds by computing the challenge ciphertext $E(\mathbb{P}', M_{c'})$ which is returned to $\mathcal{A}$.
- **Query:** $\mathcal{A}$ continues to both secret key and decryption queries except with the query for secret keys of any attribute set $\mathbb{A}$ satisfying the relation $\mathbb{P}' \subseteq \mathbb{A}$, and also the decryption query on $E(\mathbb{P}', M_{c'})$.
- **Guess:** Finally, $\mathcal{A}$ outputs a random guess bit c'_g of c', and the game is won by $\mathcal{A}$ when $c'_g = c'$.

In this game, the advantage ϵ of $\mathcal{A}$ is defined by

$$\epsilon = Pr[c'_g = c'] - \frac{1}{2}.$$

Definition 1. For all t-polynomial time adversaries, who make at most q_s secret key and q_d decryption queries, if ϵ is a negligible function of the security parameter ρ, the CP-ABE scheme is said to be (t, q_s, q_d, ϵ)-selectively secure under the "chosen-ciphertext attack (CCA)".

4.7. Augmented multi-sequence of exponents decisional diffie-Hellman problem

We present the augmented multi-sequence of exponents decisional Diffie-Hellman (n-aMSE-DDH) problem defined in [15], which was adapted from the similar pre-existing works in [14,52].

Let $\mathbb{G} = \{G_1, G_2, G_t, p, e\}$ be a pairing group. Assume there are two polynomials $\varphi(x)$ and $\vartheta(x)$ that are two co-primes. In addition, let g_1 and g_2 be the respective generators of the pairing groups G_1 and G_2. The, given

$$
\begin{aligned}
&g_1,\ g_1^{\alpha},\ g_1^{\alpha^2},\ \ldots,\ g_1^{\alpha^{n-1}},\quad g_1^{\alpha\varphi(\alpha)},\\
&g_2,\ g_2^{\alpha},\ g_2^{\alpha^2},\ \ldots,\ g_2^{\alpha^n},\\
&g_2^{1/\vartheta(\alpha)},\ g_2^{\alpha/\vartheta(\alpha)},\ g_2^{\alpha^2/\vartheta(\alpha)},\ \ldots,\ g_2^{\alpha^n/\vartheta(\alpha)},\\
&g_1^{\gamma\alpha\varphi(\alpha)},\ g_2^{\gamma},
\end{aligned}
$$

and $T \in G_T$, where $T = e(g_1, g_2)^{\gamma\varphi(\alpha)}$ or T is a random element of G_t, the n-aMSE-DDH problem decides whether the element $T = e(g_1, g_2)^{\gamma f(\alpha)}$ or just a random element of G_t. For all t-polynomial time adversaries, if the maximum advantage of solving n-aMSE-DDH problem is ϵ, it is (t, ϵ)-hard problem.

5. The proposed scheme

The proposed scheme combines the desirable attributes of the CP-ABE scheme with constant-size key and ciphertext presented in [15] and also the multi-authority KP-ABE scheme presented in [16] with a session key establishment mechanism for IoT smart devices. The underlying multi-authority CP-ABE scheme has constant size key and ciphertext with respect to the universe of attributes. The key size of the ABE governs the key size of the overall scheme. Similarly, the ciphertext size governs the communication overhead of the overall scheme. Thus, the proposed scheme is highly scalable, even for an arbitrarily large universe of attributes, which is an important necessity for IoT infrastructure. It is worth noticing that the proposed scheme relies on constant-size key/ciphertext based CP-ABE and a station-to-station variant of the Diffie-Hellman key exchange (DHKE) protocol [53]. Furthermore, the proposed scheme

Table 1
Notations used in this paper.

Symbol	Description
U, GWN, AA_k, Dev	Mobile user, gateway node, k^{th} attribute authority and smart device, respectively
ID_X, EID_X	Identity and encrypted identity of an entity X, respectively
q	A sufficiently large prime
g, h	Generators of elliptic curve groups G_1 and G_2, respectively, over finite field Z_q
(k_X, Q_X)	Private and public key pair of an entity X, where $Q_X = g^{k_X} \pmod q$
LTK_X	Long term private key of an entity X
$e(\cdot)$	A bilinear pairing such that $e(g^a, h^b) = e(g, h)^{ab}$ with $a, b \in Z_q^*$
α	Shared private key between the attribute authorities, $\alpha = e(g, h)^{\sum \alpha_k}$
$g^s_{U_k}$	User key component from AA_k
$\{g^{EID_U}, g^{s_U}\}$	Keys for user with encrypted identity EID_U
$Gen(\cdot)$, $Rep(\cdot)$	Probabilistic generation and reproduction functions for biometric fuzzy extractor, respectively
BIO_U	Personal biometrics of a mobile user U
σ_U, τ_U	Secret biometric key and public reproduction parameter associated with BIO_U, respectively
et	Error tolerance threshold value used in fuzzy extractor $Rep(\cdot)$
H_1, H_2, H_3	Collision-resistant cryptographic one-way hash functions
$\|\|$, $\oplus$	String concatenation & bitwise exclusive (XOR) operations, respectively

enables session establishment and access control between users and devices.

The proposed scheme consists of five main phases, namely, 1) setup, 2) device enrollment, 3) user registration, 4) device key precomputation, and 5) login and access control. Apart from these, we have also other phases, such as password & biometric update and dynamic device addition. The notations used in this paper are tabulated in Table 1. The different entities involved in the proposed scheme are the users (U), the gateway nodes (GWN), the attribute authorities AA_k, $k \in [1, N]$, and the smart devices (Dev). It is worth noting that the attribute authority associated with GWN is logically a separate entity included among attribute authorities AA_k. Fig. 3 gives a high-level overview of the proposed scheme. The GWN executes the system setup phase and defines the system key pair. During smart device enrollment, the GWN defines its access structure, generates secret keys, and saves them into Dev prior to the deployment of smart devices. When a user U registers in the system, AA_k defines the component keys based on the user's authorized attributes that are under its control. The GWN then consolidates component keys to define user keys for the user's smart card SC_U. In order to reduce computational overhead, the smart device periodically pre-computes the computationally heavy ABE dependent values, as shown in Section 5.5.

During the login and access control phase, U logins in the system and sends the authentication request message directly to the smart device Dev. On receiving a valid authentication request, Dev calculates a session key utilizing the pre-computed values. The device's ABE challenge values are then sent to the user U. If U is authorized for all the necessary attributes to access Dev, he/she can combine the ABE challenge values in order to calculate the same session key. As only a legitimate user with necessary attributes can recreate the session key, he/she is implicitly authenticated by the smart devices.

In the following subsections, the detailed mechanism of the proposed scheme is provided.

5.1. Setup phase

During this phase, the gateway nodes and the attribute authorities collaborate to calculate the system key pair $\{MSK, MPK\}$ based on the security parameter l_σ and the universe of attributes $\mathbb{U}$ as follows.

- *Step 1.* The GWN defines the bilinear pairing group $\mathbb{G} = \{G_1, G_2, G_T, g, h, e(\cdot)\}$.

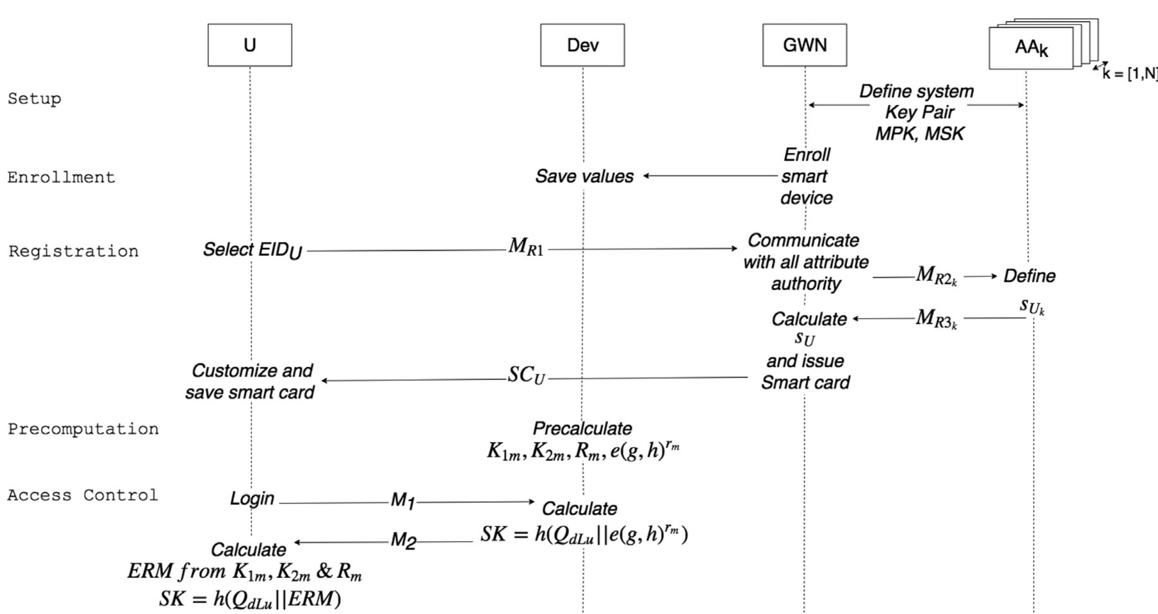

Fig. 3. High level overview of the proposed scheme.

- *Step 2.* The *GWN* chooses three collision resistant one-way hash functions H_1, H_2 and H_3 as $H_1, H_3 : \{0,1\}^* \rightarrow Z_q^*$, $H_2 : \{0,1\}^* \rightarrow \{0,1\}^{l_\sigma}$, where l_σ is a security parameter for the underlying ABE scheme.
- *Step 3.* Each attribute authority AA_k randomly selects α_k, $x_k \in Z_q$ and shares $Q_k = e(g,h)^{\alpha_k}$, $y_k = g^{x_k}$ with all other attribute authorities. Each authority AA_k then calculates the shared private key $\alpha = H_1(\Pi_{k=1}^N Q_k) = H_1(e(g,h)^{\sum_{k=1}^N \alpha_k})$.
- *Step 4.* Each pair of attribute authorities, say AA_j and AA_k agree upon a secret seed $s_{jk}(= s_{jk})$.
- *Step 5.* Each authority AA_k then defines $PRF_{jk}(\cdot)$. Only authorities AA_j and AA_k can calculate it as $g^{x_j x_k}$, which is equal to $y_k^{x_j}$ $(= y_j^{x_k})$.
- *Step 6.* The *GWN* randomly selects K_1, $K_2 \in Z_q$ and computes $h_i = h^{\alpha^i}$, $u_i = h^{K_1 \alpha^i}$ and $v_i = h^{K_2 \alpha^i}$, $\forall i \in [1, n]$, and shares K_1 with all attribute authorities.
- *Step 7.* The *GWN* finally publishes master public key $MPK = \{\mathbb{G}$, $e(\cdot)$, g, h, g^α, $\{h_i, u_i, v_i\}_{\forall i \in [1,n]}$, H_1, H_2, $H_3\}$, and keeps the master secret key $MSK = \{\alpha, K_1, K_2\}$ with it only.

5.2. IoT Device enrollment phase

IoT smart devices can be enrolled into the system dynamically at any time after the setup phase is completed. The steps required to enroll a smart device *Dev* under the proposed scheme are described below.

- The gateway node *GWN* selects a identity ID_{Dev} for the device *Dev*, and generates a random number r_{Dev}, and then calculates the long-term private key $LTK_{Dev} = H_2(ID_{Dev} \| r_{Dev})$ and $Q_{Dev_L} = g^{LTK_{Dev}}$. The *GWN* defines the access policy $\mathbb{P} = \{b_1, b_2, \cdots, b_n\}$ and computes the at most $n-1$ degree polynomial $f(\alpha, \mathbb{P})$ as $f(\alpha, \mathbb{P}) = \prod_{i=1}^n (\alpha + H_3(i))^{1-b_i}$.
- ID_{Dev}, $\mathbb{P}$, $f(\alpha, \mathbb{P})$, and the long term key LTK_{Dev} are then loaded into the smart device before it is deployed in the IoT environment.
- The *GWN* lists ID_{Dev}, $\mathbb{P}$ and Q_{Dev_L} among its list of available devices.

5.3. User registration phase

A user U needs to be registered with the gateway node *GWN* in order to access the services of any smart device *Dev*. This section describes the following steps to register the user U under the proposed scheme:

- *Step 1.* U chooses an identity ID_U, generates a secret random number $r_U \in Z_q$, calculates $EID_U = H_2(ID_U \| r_U)$ and then securely sends it to the *GWN* as registration request message M_{R1}.
- *Step 2.* The *GWN* then securely requests each attribute authority AA_k for partial key component s_{U_k} with the message $M_{R2_k} = \langle EID_U \rangle$.
- *Step 3.* Each attribute authority AA_k, $k \in [1, N]$ defines an access policy $\mathbb{A}_k = a_{1_k} a_{2_k} \cdots a_{n_k}$ for U, calculates $f(\alpha, \mathbb{A}_k) = \prod_{i=1}^n (\alpha + H_3(i))^{1-a_{i_k}}$ and $s_{U_k} = (\frac{1}{K_1})^{1/N} [\prod_{j=1, j\neq k}^N PRF_{jk}(EID_U) / f(\alpha, \mathbb{A}_k)]$, and sends the message $M_{R3_k} = \langle s_{U_k} \rangle$ to the *GWN* over secure channel.
- *Step 4.* On receiving M_{R3_k} from all attribute authorities, the *GWN* calculates

$$s_U = \prod_{k=1}^N s_{U_k} - \frac{K_2 EID_U}{K_1}$$
$$= \prod_{k=1}^N \left(\frac{1}{K_1}\right)^{\frac{1}{N}} \frac{\prod_{j=1, j\neq k}^N PRF_{jk}(EID_U)}{f(\alpha, \mathbb{A}_k)} - \frac{K_2 EID_U}{K_1}$$
$$= \prod_{k=1}^N \left(\frac{1}{K_1}\right)^{\frac{1}{N}} \frac{\prod_{j=1, j\neq k}^N PRF_{jk}(EID_U)}{\prod_{i=1}^n (\alpha + H_3(i))^{1-a_{i_k}}} - \frac{K_2 EID_U}{K_1}$$

User U	Gateway GWN $\langle K_1, K_2 \rangle$	Attribute authority A_k $\langle \alpha, K_1 \rangle$
Select ID_U		
Generate random number r_U		
Compute $EID_U = H_2(ID_U \parallel r_U)$		
$\xrightarrow[\text{secure channel}]{M_{R1}=\langle EID_U \rangle}$		
	$\xrightarrow[\text{secure channel}]{M_{R2_k}=\langle EID_U \rangle}$	
		$f(\alpha, \mathbb{A}_k) = \prod_{i=1}^n (\alpha + H_3(i))^{1-a_{i_k}}$
		$s_{U_k} = \left(\frac{1}{K_1}\right)^{\frac{1}{N}} \frac{\prod_{j=1, j\neq k}^N PRF_{jk}(EID_U)}{f(\alpha, \mathbb{A}_k)}$
		$\xleftarrow[\text{secure channel}]{M_{R3_k}=\langle s_{U_k} \rangle}$
	$s_U = \prod_{k=1}^N s_{U_k} - \frac{K_2 EID_U}{K_1}$	
	$SC_U = \{EID_U, g^{EID_U}, g^{s_U}, DeviceList\}$	
	$\xleftarrow[\text{secure channel}]{SC_U}$	
Select PW_U		
Imprint BIO_U		
Compute $(\sigma_U, \tau_U) = Gen(BIO_U)$,		
$IPB_U = H_2(ID_U \parallel PW_U \parallel \sigma_U)$,		
$r_U^* = r_U \oplus H_2(PW_U \parallel ID_U \parallel \sigma_U)$,		
$EID_U^* = EID_U \oplus H_2(r_U \parallel PW_U \parallel \sigma_U)$,		
$g^{EID_U *} = g^{EID_U} \oplus H_2(r_U \parallel \sigma_U \parallel PW_U)$,		
$g^{s_U *} = g^{s_U} \oplus H_2(PW_U \parallel r_U \parallel \sigma_U)$,		
$DeviceList^* = DeviceList \oplus H_2(PW_U \parallel r_U \parallel \sigma_U)$.		
Insert r_U^*, IPB_U, τ_U into *smart card*		
Replace $EID_U, g^{EID_U}, g^{s_U}$ and $DeviceList$ with $EID_U^*, g^{EID_U *}, g^{s_U *}$ and $DeviceList^*$ in *smart card*		

Fig. 4. Summary of user registration phase.

$$= \frac{1}{K_1} \frac{\prod_{k=1}^N \prod_{j=1, j\neq k}^N PRF_{jk}(EID_U)}{\prod_{k=1}^N \prod_{i=1}^n (\alpha + H_3(i))^{1-a_{i_k}}} - \frac{K_2 EID_U}{K_1}$$
$$= \frac{1}{K_1} \frac{\prod_{j,k=1, j\neq k}^N PRF_{jk}(EID_U)}{\prod_{i=1}^n (\alpha + H_3(i))^{1-\sum_{k=1}^N a_{i_k}}} - \frac{K_2 EID_U}{K_1}.$$

Since by design $\sum_{j,k=1, j\neq k}^N PRF_{jk}(\cdot) = 1$ and $\mathbb{A}_k, k \in [1, N]$ are disjoint subsets of $\mathbb{U}$, we have $\sum_{k=1}^N a_{i_k} = a_i$. Thus, $s_U = \frac{1}{K_1}(\frac{1}{\prod_{i=1}^n (\alpha + H_3(i))^{1-a_i}} - K_2 EID_U) = \frac{1}{K_1}(\frac{1}{f(\alpha, \mathbb{A})} - K_2 EID_U)$. The *GWN* then securely issues the smart card SC_U containing the credentials $\{EID_U$, g^{EID_U}, g^{s_U} *DeviceList*$\}$, where *DeviceList* lists the identities, access policy and public keys of the smart device user U who is authorized to access those information.

- *Step 5.* U on receiving SC_U chooses a password PW_U and imprints his/her biometric BIO_U into the sensor of a specific terminal. SC_U then calculates the secret biometric key σ_U and public reproduction parameter τ_U using the fuzzy generator function $Gen(\cdot)$ as $(\sigma_U, \tau_U) = Gen(BIO_U)$, the identity verification token $IPB_U = H_2(ID_U \| PW_U \| \sigma_U)$ and $r_U^* = r_U \oplus H_2(PW_U \| ID_U \| \sigma_U)$, and saves r_U^*, IPB_U and τ_U its memory.
- *Step 6.* SC_U finally replaces EID_U, g^{EID_U}, g^{s_U} and *DeviceList* with computed $EID_U^* = EID_U \oplus H_2(r_U \| PW_U \| \sigma_U)$, $g^{EID_U *} = g^{EID_U} \oplus H_2(r_U \| \sigma_U \| PW_U)$, $g^{s_U *} = g^{s_U} \oplus H_2(PW_U \| r_U \| \sigma_U)$ and $DeviceList^* = DeviceList \oplus H_2(PW_U \| r_U \| \sigma_U)$.

The gateway node *GWN* also saves EID_U in its *user_information* table. Fig. 4 summarizes the user registration phase.

5.4. User key update phase

As discussed by the authors in [35], it may be sometimes necessary required to alter the access policy of a registered legal user. If the access policy associated with the user U defined by AA_k changes, it should not require a complete re-calculation of the user U's key involving all the attribute authorities. In such a case, as long as the gateway node GWN preserves the information $\{EID_U, s_U, s_{U_{k\in[0,N]}}\}$ for the user U, only that user U, GWN and AA_k can collaborate to update the U's keys. The following steps are then essential to execute this phase:

- *Steps 1 & 2.* These steps are identical as those presented in Section 5.3.
- *Step 3.* Instead of $M_{R3_k} = \langle s_{U_k}\rangle$, AA_k sends $M'_{R3_k} = \langle {s_{U_k}}^{old}, s_{U_k}\rangle$ to the GWN via secure channel, where ${s_{U_k}}^{old}$ and s_{U_k} are computed based on the old and updated access policy, respectively.
- *Step 4.* The GWN computes the updated s'_U by calculating $s'_U = ((s_U + KEID) \times \frac{s_{U_k}}{{s_{U_k}}^{old}}) - KEID$, where $KEID = \frac{K_2 EID_U}{K_1}$. The GWN then sets $s_U = s'_U$, recomputes g^{s_U} and re-issues the smart card SC_U to the user U securely.

5.5. Device key pre-Computation phase

In this phase, the smart device Dev pre-computes the ABE related computation in order to eliminate repeated computational overheads. This step is repeated semi-periodically. For this issue, Dev randomly selects $k_{Dev} \in \{0,1\}^{l_\sigma}$, computes the following

$$r_m = H_1(\mathbb{P}, k_{Dev}),\ R_m = (g^{\alpha})^{r_m} = g^{\alpha r_m},$$
$$K_{1m} = (\textstyle\prod_{i=0}^{n}(u_i^{f_i}))^{r_m} = h^{K_1 f(\alpha,\mathbb{P}) r_m},$$
$$K_{2m} = (\textstyle\prod_{i=0}^{n}(v_i^{f_i}))^{r_m} = h^{K_2 f(\alpha,\mathbb{P}) r_m},$$

and saves R_m, K_{1m}, K_{2m} and $e(g,h)^{r_m}$ into its memory.

5.6. Login and access control phase

A registered user U can login and authenticate him/herself, and then securely negotiate a session key with a smart device provided he/she posses the correct attributes under the proposed scheme through the following steps.

- *Step 1.* U provides his/her identity ID_U and password PW_U, and also imprints biometric BIO_U at the sensor of a particular terminal. The smart card SC_U then calculates $\sigma_U = Rep(BIO_U, \tau_U)$ provided that the Hamming distance between earlier registered biometric and current biometric BIO_U is less than or equal to predefined error tolerance threshold parameter et and $IPB'_U = H_2(ID_U \,\|PW_U\, \|\sigma_U)$. Only if IPB'_U is equal to IPB_U stored in the smart card, the login is successful. SC_U also calculates $r_U = r^*_U \oplus H_2(PW_U \,\|ID_U\, \|\sigma_U)$ and recovers EID_U, $\{D_{jk}\}$, g^{EID_U}, g^{s_U} and $DeviceList$ with $EID_U = EID^*_U \oplus H_2(r_U \,\|PW_U\, \|\sigma_U)$, $g^{EID_U} = g^{EID_U*} \oplus H_2(r_U \,\|\sigma_U\, \|PW_U)$, $g^{s_U} = g^{s_U*} \oplus H_2(PW_U \,\|r_U\, \|\sigma_U)$ and $DeviceList = DeviceList^* \oplus H_2(PW_U \,\|r_U\, \|\sigma_U)$.
- *Step 2.* U then selects the accessed smart device Dev and retrieves identity ID_{Dev}, public key Q_{Dev_L} and access policy $\mathbb{P}$ from $DeviceList$, and checks if $\mathbb{A}$ satisfies $\mathbb{P}$, i.e., if $\mathbb{P} \subseteq \mathbb{A}$. If the condition does not satisfy, the phase is terminated. Otherwise, U selects a short term secret key k_U and calculates $Q_U = g^{k_U}$, $Q_{dLu} = (Q_{Dev_L})^{k_U}$, defines the dynamic identity $DID_U = RNG_{EID_U}(TS_U)$ and $dynamic\_token = (DID_U \,\|IDTS) \oplus H_1(TS_U \,\|Q_{dLu})$, where $IDTS = H_2(ID_{Dev} \,\|TS_U)$ and TS_U is the current time stamp. Finally, the message $M_1 = \langle dynamic\_token, Q_U, TS_U\rangle$ is sent to the smart device Dev via open channel.
- *Step 3.* Dev on receiving M_1 and verifying the freshness of timestamp TS_U by the condition $|TS_U - TS^*_U| < \Delta T$ where ΔT is the maximum allowable transmission delay and TS^*_U is the time when M_1 is received, it calculates $Q_{dLu} = (Q_U)^{LTK_{Dev}}$ and retrieves DID_U and $IDTS$ by $dynamic\_token \oplus H_1(TS_U \,\|Q_{dLu})$. If $H_2(ID_{Dev} \,\|TS_U) \neq IDTS$, the phase will be terminated. Otherwise, Dev continues to calculate the session key $SK = H_2(Q_{dLu} \,\| e(g,h)^{r_m})$ and $cert = H_2(SK \,\|\, TS_{Dev})$. Dev then logs DID_U and TS_U into its $access\_table$, and finally it sends the message $M_2 = \langle R_m, K_{1m}, K_{2m}, cert, TS_{Dev}\rangle$ to U via open channel.
- *Step 4.* U on receiving M_2, SC_U checks the freshness of the message by checking the condition $|TS_{Dev} - TS^*_{Dev}| < \Delta T$, where TS^*_{Dev} is the time when M_2 is received. SC_U calculates at most $n - |\mathbb{P}|$ degree polynomial $F(x,\mathbb{A},\mathbb{P})$ as $F(x,\mathbb{A},\mathbb{P}) = \prod_{i=1}^{n-|\mathbb{P}|}(x + H_3(i))^{a_i-b_i}$. Let F_i represent the coefficient of x^i in the polynomial $F(x,\mathbb{A},\mathbb{P})$. Note that $F_0 \neq 0$. SC_U then computes

$$W = e\left(R_m, \textstyle\prod_{i=1}^{n-|\mathbb{P}|}(h_{i-1})^{F_i}\right) = e(g^{\alpha r_m}, \textstyle\prod_{i=1}^{n-|\mathbb{P}|} h^{a^{i-1}F_i})$$
$$= e(g,h)^{\alpha r_m \sum_{i=1}^{n-|\mathbb{P}|} a^{i-1}F_i}$$
$$= e(g,h)^{r_m \sum_{i=1}^{n-|\mathbb{P}|}(\alpha^i F_i + r_m F_0 - r_m F_0)}$$
$$= e(g,h)^{r_m F(\alpha) + r_m F_0},$$
$$U = e(g^{s_U}, K_{1m}) = e(g,h)^{K_1 f(\alpha,\mathbb{P}) r_m s_U)},$$
$$V = e(g^{EID_U}, K_{2m}) = e(g,h)^{K_2 f(\alpha,\mathbb{P}) r_m r_U)},$$
$$UV = e(g,h)^{K_1 f(\alpha,\mathbb{P}) r_m s_U) + K_2 f(\alpha,\mathbb{P}) r_m r_U}$$
$$= e(g,h)^{r_m f(\alpha,\mathbb{P})(K_1 s_U + K_2 r_U)} = e(g,h)^{r_m \frac{f(\alpha,\mathbb{P})}{f(\alpha,\mathbb{A})}} = e(g,h)^{r_m F(\alpha)},$$
$$ERM = \left(\frac{UV}{W}\right)^{\frac{1}{F_0}} = \left(\frac{e(g,h)^{r_m F(\alpha)}}{e(g,h)^{r_m F(\alpha) + r_m F_0}} y\right)^{\frac{1}{F_0}} = e(g,h)^{r_m}.$$

SC_U also calculates the session key $SK = H_2(Q_{dLu} \,\|ERM)$ and checks if $H_2(SK \,\|TS_{Dev})$ is equal to $cert$. If it is so, U considers the session key as a valid key.

Fig. 5 summarizes the login and access control phase.

User U	Smart device Dev
$\{EID^*_U, g^{EID^*_U}, g^{s^*_U}, DeviceList^*, r^*_U, IPB_U, \tau_U, et\}$	$\{LTK_{Dev}, R_m, K_{1m}, K_{2m}, e(g,h)^{r_m}\}$
Enter ID_U, PW_U Imprint BIO_U Compute $\sigma_U = Rep(BIO_U, \tau_U)$, $IPB'_U = H_2(ID_U \parallel PW_U \parallel \sigma_U)$. If $IPB_U \neq IPB'_U$, terminate Compute $r_U = r^*_U \oplus H_2(PW_U \parallel ID_U \parallel \sigma_U)$, $EID_U = EID^*_U \oplus H_2(r_U \parallel PW_U \parallel \sigma_U)$, $g^{EID_U} = g^{EID_U*} \oplus H_2(r_U \parallel \sigma_U \parallel PW_U)$, $g^{s_U} = g^{s_U*} \oplus H_2(PW_U \parallel r_U \parallel \sigma_U)$, $DeviceList = DeviceList^* \oplus$ $H_2(PW_U \parallel r_U \parallel \sigma_U)$. Retrieve ID_{Dev}, Q_{DevL} and $\mathbb{P}$ from $DeviceList$. Select $k_U \in Z_q$ Compute $Q_U = g^{k_U}$, $Q_{dLu} = (Q_{Dev_L})^{k_U}$, $IDTS = H_2(ID_{Dev} \parallel TS_U)$, $DID_U = RNG_{EID_U}(TS_U)$, $dynamic\_token = (DID_U \parallel IDTS)$ $\oplus H_1(TS_U \parallel Q_{dLu})$. $\xrightarrow{M_1 = \langle dynamic\_token, Q_U, TS_U\rangle}$	
	If $\|TS^*_U - TS_U\| > \Delta T$, terminate Calculate $Q_{dLu} = (Q_U)^{LTK_{Dev}}$, $DID_U \parallel IDTS = dynamic\_token \oplus$ $H_1(TS_U \parallel Q_{dLu})$. If $H_2(ID_{Dev} \parallel TS_U) \neq IDTS$, terminate Using the pre-computed values r_m, K_{1m}, K_{2m} and R_m, compute $SK = H_2(Q_{dLu} \parallel e(g,h)^{r_m})$, $cert = H_2(SK \parallel TS_{Dev})$. Log DID_U, TS_U into $access\_table$ $\xleftarrow{M_2 = \langle R_m, K_{1m}, K_{2m}, cert, TS_{Dev}\rangle}$
If $\|TS^*_{Dev} - TS_{Dev}\| > \Delta T$, terminate $F(x,\mathbb{A},\mathbb{P}) = \prod_{i=1}^{n-\|\mathbb{P}\|}(x + H_3(i))^{a_i-b_i}$ Compute $W = e\left(R_m, \prod_{i=1}^{n-\|\mathbb{P}\|}(h_{i-1})^{F_i}\right)$, $U = e(g^{s_U}, K_{1m})$, $V = e(g^{EID_U}, K_{2m})$, $ERM = \left(\frac{UV}{W}\right)^{\frac{1}{F_0}}$, $SK' = H_2(Q_{dLu} \parallel ERM)$. If $H_2(SK \parallel TS_{Dev}) \neq cert$, terminate	
Both U and Dev agree on common secret session key $SK(= SK')$	

Fig. 5. Summary of login and access control phase.

User U	Smart card
	$\{EID_U^*, g^{EID_U*}, g^{s_U*},$ $DeviceList^*, r_U^*, IPB_U, \tau_U, et\}$
Enter ID_U, PW_U	
Imprint BIO_U	Compute $\sigma_U = Rep(BIO_U, \tau_U)$, $IPB'_U = H_2(ID_U \parallel PW_U \parallel \sigma_U)$. If $IPB_U \neq IPB'_U$, terminate Compute $r_U = r_U^* \oplus H_2(PW_U \parallel ID_U \parallel \sigma_U)$, $EID_U = EID_U^* \oplus H_2(r_U \parallel PW_U \parallel \sigma_U)$, $g^{EID_U} = g^{EID_U*} \oplus H_2(r_U \parallel \sigma_U \parallel PW_U)$, $g^{s_U} = g^{s_U*} \oplus H_2(PW_U \parallel r_U \parallel \sigma_U)$, $DeviceList =$ $DeviceList^* \oplus H_2(PW_U \parallel r_U \parallel \sigma_U)$.
Enter new PW_U^{new}	
Imprint new BIO_U^{new}	Compute $(\sigma_U^{new}, \tau_U^{new}) = Gen(BIO_U^{new})$, $EID_U^{*new} = EID_U \oplus H_2(r_U \parallel PW_U^{new} \parallel \sigma_U^{new})$, $g^{EID_U*new} = g^{EID_U} \oplus H_2(r_U \parallel \sigma_U^{new} \parallel PW_U^{new})$, $g^{s_U*new} = g^{s_U} \oplus H_2(PW_U^{new} \parallel r_U \parallel \sigma_U^{new})$, $DeviceList^{*new} =$ $DeviceList \oplus H_2(PW_U^{new} \parallel r_U \parallel \sigma_U^{new})$, $r_U^{*new} = r_U \oplus H_2(PW_U^{new} \parallel ID_U \parallel \sigma_U^{new})$, $IPB_U^{new} = H_2(ID_U \parallel PW_U^{new} \parallel \sigma_U^{new})$. Replace $EID_U^*, g^{EID_U*}, g^{s_U*}$, $DeviceList^*, r^*, IPB_U$ and τ_U with $EID_U^{*new}, g^{EID_U*new}, g^{s_U*new}$, $DeviceList^{*new}, r^{*new}, IPB_U^{new}$, and τ_U^{new} in smart card, respectively

Fig. 6. Summary of password and biometric update phase.

5.7. Password and biometric update phase

In this section, we detail the process to update the biometric as well as password of a legal registered user U under the proposed scheme with the following steps.

- *Step 1.* U provides his/her identity ID_U and current password PW_U, and also imprints current biometric BIO_U. The smart card SC_U calculates $\sigma_U = Rep(BIO_U, \tau_U)$ and $IPB'_U = H_2(ID_U \parallel PW_U \parallel \sigma_U)$. Only if the calculated IPB'_U is equal to IPB_U available in the smart card, the login is considered as successful. SC_U further calculates $r_U = r_U^* \oplus H_2(PW_U \parallel ID_U \parallel \sigma_U)$ and recovers EID_U, g^{EID_U}, g^{s_U} and $DeviceList$ as in Step 1 (Section 5.6).
- *Step 2.* U now selects a new password PW_U^{new} and imprints a new biometric BIO_U^{new}. It is worth noting that generally biometric residing of a user is unchanged, and therefore, if the user does not want to change old biometric BIO_U, he/she can keep old biometric, that is, in this case $BIO_U^{new} = BIO_U$. SC_U calculates σ_U^{new} and τ_U^{new} using the fuzzy generator function $Gen(\cdot)$ and the identity verification token IPB_U^{new} is recalculated as $H_2(ID_U \parallel PW_U^{new} \parallel \sigma_U^{new})$. SC_U also calculates $r_U^{*new} = r_U \oplus H_2(PW_U^{new} \parallel ID_U \parallel \sigma_U^{new})$, $EID_U^{*new} = EID_U \oplus H_2(r_U \parallel PW_U^{new} \parallel \sigma_U^{new})$, $g^{EID_U*new} = g^{EID_U} \oplus H_2(r_U \parallel \sigma_U^{new} \parallel PW_U^{new})$, $g^{s_U*new} = g^{s_U} \oplus H_2(PW_U^{new} \parallel r_U \parallel \sigma_U^{new})$ and $DeviceList^{*new} = DeviceList \oplus H_2(PW_U^{new} \parallel r_U \parallel \sigma_U^{new})$.
- *Step 3.* SC_U finally replaces EID_U^*, g^{EID_U}, g^{s_U}, $DeviceList$, R_U^*, IBP_U and τ_U with EID_U^{*new}, g^{EID_U*new}, g^{s_U*new}, $DeviceList^{*new}$, r_U^{*new}, IPB_U^{new} and τ_U^{new}, respectively.

Fig. 6 summarizes the password and biometric update phase.

5.8. Dynamic IoT device addition phase

New smart devices can be dynamically enrolled in the system at any time after the setup phase through the steps described in Section 5.2.

Remark 1. Each login request in the proposed scheme includes *dynamic_token*, which contains DID_U and TS_U. Since the *access_table* is consolidated at the *GWN* and the set of registered users is finite, the *GWN* can calculate all $DID'_U = RNG_{EID'_U}(TS_U)$. If $DID'_U = DID_U$, the *GWN* can set $EID_U = EID'_U$. Thus, the *GWN* can pinpoint which login corresponds to which user through a relatively simple search. A malicious user U_{mal} with the correct set of attributes may send a random value instead of DID_U with the intention of eschewing accountability. Therefore, in this case, when the *GWN* tries to reconstruct EID_U from DID_U, it will fail. By tracing the attribute set $\mathbb{P}$ of the smart devices, the *GWN* can find the set of users in which U_{mal} belongs to. However, this drawback can be totally eliminated by a slight modification to the login and access control phase described in Section 5.6, in which instead of directly sending the message M_1 to the smart device, it can be verified by the *GWN*. However, this step will require additional communication overhead of 84 bytes, and the number of messages exchanged between the entities to be increased to 3 from 2.

6. Security analysis

In this section, through a formal security analysis we first comment on the security of the underlying ABE scheme. Next, we present analytical study on the semantic security in the proposed scheme through the widely-accepted “Real-Or-Random (ROR)” model for session key security purpose only. Afterwards, we informally (non-mathematically) demonstrate the proposed scheme’s resistance against various known attacks. Furthermore, we also provide formal security verification using the widely-used AVISPA software verification tool.

6.1. Formal security analysis of the underlying ABE scheme

To prove the security of the underlying ABE scheme (MA-CP-ABE), we adopt the augmented multi-sequence of exponents decisional Diffie-Hellman (n-aMSE-DDH) problem (defined in Section 4.7) in conjuncture with the selective game for CP-ABE scheme (defined in Section 4.6). Our proof is modeled in the similar proofs as provided in [14,15]. We establish the security of MA-CP-ABE in Theorem 1.

Theorem 1. *The proposed MA-CP-ABE scheme is (t, q_s, q_d, ϵ)-selectively secure if the n-aMSE-DDH problem is (t', ϵ')-hard, where $t' = t + O(q_s(n^2 t_{em}) + q_d(n t_{em} + t_{bp}))$, $\epsilon' = \epsilon - \frac{q_{H_2}}{p}$, $n = |\mathbb{U}|$, and t_{em}, and t_{bp} denote the average time required for group exponentiation and pairing operations, respectively, and q_{H_2} is the number of queries made to the H_2 oracle.*

Proof. Suppose an adversary $\mathcal{A}$ exists who can break the security of the proposed MA-CP-ABE with an advantage (t, q_s, q_d, ϵ). We then construct an algorithm $\mathcal{B}$ that should solve the n-aMSE-DDH problem with the advantage $\epsilon' = \epsilon - \frac{q_{H_2}}{p}$ by interacting with $\mathcal{A}$ in the following manner.

Initialization: $\mathcal{A}$ submits the “challenge access policy” $\mathbb{P}^*$, where there are n attributes. Assume that $\mathbb{P}^* = b_1 b_2 \cdots b_n$. $\mathcal{B}$ sets the following:

$$f(x, \mathbb{P}^*) = \prod_{i=1}^{n} \Big(x + H_4(i)\Big)^{1-b_i} = \vartheta(x),$$

$$\prod_{i=1}^{n} \Big(x + H_4(i)\Big)^{b_i} = f(x),$$

where $\vartheta(x)$ and $f(x)$ are the $n - |\mathbb{P}^*|$-degree and $|\mathbb{P}^*|$-degree polynomial functions, respectively.

$\mathcal{B}$ then sends $\{\vartheta(x), \vartheta(x)\}$ to the challenger and receives an instance of the previously defined n-aMSE-DDH problem with pairing group $\mathbb{G} = \{G_1, G_2, G_t, p, e\}$, and a value $T \in G_T$. $\mathcal{B}$ must then identify if $T = e(g_1, g_2)^{\gamma \vartheta(\alpha)}$ or it is just a randomly chosen element of G_t.

Setup: In this phase, $\mathcal{B}$ in collaboration with the attribute authorities, calculates α, selects random numbers rn_1, $rn_2 \in Z_p^*$ and implicitly sets $k_1 = rn_1/\vartheta(\alpha)$ and $k_2 = rn_2/\vartheta(\alpha)$. The master public key parameters h_i, u_i, v_i, g^α and $e(g, h)$ are calculated via the challenge parameters as follows:

$$h = g_2,$$
$$h_i = g_2^{\alpha^i},$$
$$u_i = \left(g_2^{\alpha^i/\vartheta(\alpha)}\right)^{rn_1},$$
$$v_i = \left(g_2^{\alpha^i/\vartheta(\alpha)}\right)^{rn_2},$$
$$g^\alpha = g_1^{\alpha\varphi(\alpha)},$$
$$e(g, h) = e(g_1, g_2)^{\varphi(\alpha)}.$$

$\mathcal{B}$ sends the public key $MPK = \{\mathbb{G},\ e(g, h),\ g^\alpha,\ h_i,\ v_i,\ u_i\}$ to $\mathcal{A}$.

Hash Queries: $\mathcal{A}$ can access the hash oracles (H_1, H_2, H_3 and H_4), and $\mathcal{B}$ also maintains four lists $\mathcal{L}_{H_1}$, $\mathcal{L}_{H_2}$, $\mathcal{L}_{H_3}$ and $\mathcal{L}_{H_4}$ to record the queries and responses. Specifically, if a query has a previous response and the output result was recorded in the list, $\mathcal{B}$ simply responds with the recorded result. Otherwise, $\mathcal{B}$ will perform the following:

- H_4: Let the query to H_4 be $i \in [1, n]$. Note that $\mathbb{P}^* = b_1 b_2 \cdots b_n$. If $b_i = 0$, $\mathcal{B}$ responds to $H_4(i)$ with a new root of the polynomial $\vartheta(x)$; otherwise, if $b_i = 1$, $\mathcal{B}$ responds to $H_4(i)$ with a new root of the polynomial $f(x)$.
- H_2: Let the query to H_2 be $k'_m = KDF(r'_m P)$. $\mathcal{B}$ responds to $H_2(k'_m)$ with a random number $R_i \in \{0, 1\}^{l_{\sigma_m}}$.
- H_3: Let the query to H_3 be t_i. $\mathcal{B}$ responds to $H_3(t_i)$ with a random number $Q_i \in \{0, 1\}^{l_m}$.
- H_1: Let the query to H_1 be $(\mathbb{P}_i,\ M_i,\ t_i)$. $\mathcal{B}$ then responds to $H_1(\mathbb{P}_i,\ M_i,\ t_i)$ with a random number $r_i \in Z_p^*$.

Query: For a user secret key (decryption key) for $\mathbb{A} = a_1 a_2 \cdots a_n$, $\mathcal{B}$ sets

$$f(x, \mathbb{A}) = \prod_{i=1}^{n} \Big(x + H_4(i)\Big)^{1-a_i} = f_\varphi(x, \mathbb{A}) \cdot f_\vartheta(x, \mathbb{A}),$$

where the roots of the polynomials $f_f(x, \mathbb{A})$ and $f_\vartheta(x, \mathbb{A})$ are respectively from $f(x)$ and $\vartheta(x)$. It is worth noticing that the degree of the polynomial $f_f(x, \mathbb{A})$ turns out to be non-zero, if the attribute set $\mathbb{A}$ does not fulfill the challenge access structure $\mathbb{P}^*$.

Choose r randomly from Z_p^* and implicitly set $r_u = k_1 r\alpha/k_2$.

Then, set $g^{r_u} = \left(g_1^{\alpha f(\alpha)}\right)^{w_1 r/w_2}$. Since $\mathbb{A}$ does not satisfy $\mathbb{P}^*$, the following is a polynomial of degree at most $n-1$:

$$\hat{f}(x) = \frac{1}{w_1} \cdot \frac{\vartheta(x) \cdot f(x)}{f(x, \mathbb{A})}.$$

Hence, we can calculate $g_1^{\hat{f}(\alpha)}$ from the parameters $g_1, g_1^\alpha, \ldots, g_1^{\alpha^{n-1}}$ and $\hat{f}(x)$. Furthermore, set $g^{s_u} = g_1^{\hat{f}(\alpha)}(g_1^{\alpha f(\alpha)})^{-r}$, where s_u is implicitly defined as $s_u = \frac{1}{k_1}\left(\frac{1}{f(\alpha,\mathbb{A})} - k_2 r_u\right)$. Finally, $\mathcal{B}$ sends the secret key $k_u = \{g^{r_u},\ g^{s_u}\}$ to the adversary $\mathcal{A}$.

For any decryption query on $E[\mathbb{P}_i, M_i]$, if there exists $(\mathbb{P}_i,\ M_i,\ t_i,\ r_i,\ R_i,\ Q_i)$ in the query list such that the ciphertext is generated using r_i, the decryption query outputs M_i; otherwise, it outputs $\perp$ (null). For all valid encryptions, the responses from the hash oracles are essential and the responses will contain the random number r_i used in encryption. As a result, no query will be aborted.

Challenge: In this phase, $\mathcal{A}$ outputs the two messages $(M_0,\ M_1)$ for the challenge, where no queried user secret keys fulfill the challenge access structure $\mathbb{P}^*$. $\mathcal{B}$ implicitly defines $r_m = \gamma$ and sets $R_m = g_1^{\gamma\alpha f(\alpha)}, K_{1m} = (g_2^\gamma)^{rn_1},\ K_{2m} = (g_2^\gamma)^{rn_2}$. $\mathcal{B}$ picks randomly $R^* \in \{0, 1\}^{l_{\sigma_m}}$ and $Q^* \in \{0, 1\}^{l_m}$, and then sets $C_{\sigma_m} = R^*$ and $C_m = Q^*$. The challenge ciphertext $C^* = \{\mathbb{P}^*,\ R_m,\ K_{1m},\ K_{2m},\ C_{\sigma_m},\ C_m\}$ will be provided to $\mathcal{A}$.

If $T = e(g_1,\ g_2)^{\gamma f(\alpha)}$, $e(g, h)^{r_m} = e(g_1^{f(\alpha)},\ g_2)^\gamma = T$. Using the random oracles, $\mathcal{A}$ must be able to calculate $T = e(g, h)^{r_m}$, and further query it to the H_2 oracle for successful decryption.

Query: In this phase, the response is same as the former phase with the following two restrictions:

(i) No user secret key query fulfils the challenge access structure.
(ii) No decryption query on the challenge ciphertext.

Guess: In this phase, $\mathcal{A}$ outputs a guess bit c'_g, and if there exists a query on T to the H_2 oracle, $\mathcal{B}$ outputs 1; otherwise, it is a random element in G_t.

In the guess phase, if $\mathcal{A}$ breaks the proposed encryption with advantage ϵ, then $e(g, h)^{r_m}$ appears in the query list $\mathcal{L}_{H_2}$ with the probability $\epsilon + 1/2$ at least. The only error event is that T is a random group element, but it is queried to the H_2 oracle. This happens with the probability q_{H_2}/p at most. $\mathcal{B}$ is then able to distinguish $T = e(g_1,\ g_2)^{\gamma f(\alpha)}$ or a random element in G_t with an advantage of at least $\epsilon - q_{H_2}/p$.

Hence, the collision resistance of the proposed scheme follows from the fact that $\mathcal{A}$ can make multiple adaptive secret key queries before and after the challenge phase.

The simulation time is dominated by the decryption key generation and the decryption operation. In addition, each key generation needs $n(n-1)$ point multiplications, and each decryption operation requires n point multiplications and three pairing operations. Summing up all these factors, the total time complexity becomes $O(q_s(n^2 t_{em}) + q_d(n t_{em} + t_{bp}))$. Hence, the theorem is proved. □

6.2. Formal security using ROR model

This section presents formal security of our proposed CP-ABE based access control scheme (CP-ABE-ACS) using the Real-Or-Random (ROR) model [18]. The purpose of this analysis is to prove the semantic security of the proposed CP-ABE-ACS against deriving the session key between a user and an IoT smart device. We assume that an adversary $\mathcal{A}$ interacts with $\mathcal{P}^t$, the t^{th} instance of an executing participant (a user U or an IoT smart device Dev). As mentioned in [54], [55], the ROR model assumes various queries simulating a real attack, such as *Send*($\mathcal{P}^t, m$), *Corrupt(U, a)*, *Test*($\mathcal{P}^t$), *Execute(U, Dev)* and *Reveal*($\mathcal{P}^t$) query, which are discussed in Table 2, and are also provided in detail in [56].

We assume that number of hash *H*, *Send* and *Execute* queries are q_H, q_s and q_e, respectively. Moreover, we assume l_H and l_b are the lengths of hash and biometric key, respectively, whereas L_H, L_A and L_T denote the lists that record outputs of hash H, random oracle, and message transcripts, respectively. In addition, it is assumed that the password space $\mathcal{D}$ follows the frequency distribution according to Zipf's law [57], and C' and s' are the Zipf parameters [57].

The following definitions are necessary for our formal security analysis.

Definition 2 (One-way hash function)**.** A one-way collision-resistant hash function H: $\{0, 1\}^* \to \{0, 1\}^{l_h}$ is a deterministic function which produces a binary string $H(s) \in \{0, 1\}^{l_h}$ of fixed-length l_h as hash output (message digest) on an input with an arbitrary length binary string $s \in \{0, 1\}^*$. An adversary $\mathcal{A}$'s advantage in finding collision is defined as follows:

$$Adv_{\mathcal{A}}^{HASH}(t) = Pr[(s, s') \leftarrow_R \mathcal{A} : s \neq s',\ H(s) = H(s')],$$

where an event E's probability is $Pr[E]$ and $(s, s') \leftarrow_R \mathcal{A}$ means the pair $(s,\ s')$ is randomly chosen by $\mathcal{A}$. An $(\psi,\ t)$-adversary $\mathcal{A}$ attacking H's collision resistance indicates that $\mathcal{A}$'s the runtime is at most t with $Adv_{(A)}^{HASH}(t) \leq \psi$.

Table 2
Different queries and their descriptions.

Query	Description
$Send(\mathcal{P}^t, m)$	It enables $\mathcal{A}$ to send a request message m to $\mathcal{P}^t$, and $\mathcal{P}^t$ replies accordingly
$Corrupt(U, a)$	Depending on a, $\mathcal{A}$ can obtain biometric and password of U
$Test(\mathcal{P}^t)$	$\mathcal{A}$ requests $\mathcal{P}^t$ for the session key SK, $\mathcal{P}^t$ replies probabilistically on outcome of a flipped unbiased coin b
$Execute(U, Dev)$	It enables $\mathcal{A}$ to eavesdrop the messages communicated between U and Dev
$Reveal(\mathcal{P}^t)$	It enables $\mathcal{A}$ to obtain the session key SK generated between $\mathcal{P}^t$ and its partner

Definition 3 (Decisional Diffie-Hellman Problem (DDHP))**.** The DDHP states that given a quadruple $\langle g, g^{k_1} \pmod q, g^{k_2} \pmod q, g^{k_3} \pmod q\rangle$ to decide if $k_3 = k_1k_2$ or a uniform value, where $k_1, k_2, k_3 \in Z_q^*$.

Definition 4 (Semantic security)**.** Let $Adv_{\mathcal{A}}^{CP-ABE-ACS}(t)$ denotes the advantage of an adversary $\mathcal{A}$ running in polynomial time t in breaking the semantic security of the proposed CP-ABE based access control scheme (CP-ABE-ACS) for deriving the session key (SK). Then, $Adv_{\mathcal{A}}^{CP-ABE-ACS}(t) = |2Pr[b' = b] - 1|$, where b and b' are the correct and guessed bits, respectively.

Theorem 2. *Suppose $Adv_{\mathcal{A}}^{CP-ABE-ACS}(t)$ denotes the advantage function of an adversary $\mathcal{A}$ running in polynomial time t in breaking the semantic security of the proposed scheme CP-ABE-ACS as defined in Definition 4. Then, $Adv_{\mathcal{A}}^{CP-ABE-ACS}(t) \le \frac{q_H^2+18q_H}{2^{l_H}} + 2[\max\{C' \cdot q_s^{s'}, q_s(\frac{1}{2^{l_b}}, \varepsilon_{bm})\} + Adv_{\mathcal{A}}^{DDHP}(t)]$, where q_H, q_s, l_H, l_b, C' and s' convey the meanings as mentioned above, ε_{bm} is the probability of false positive in biometrics [58] and $Adv_{\mathcal{A}}^{DDHP}(t)$ is the advantage of $\mathcal{A}$ in solving DDHP (defined in Definition 3). Therefore, assuming $Adv_{\mathcal{A}}^{DDHP}$ is secure, $Adv_{\mathcal{A}}^{CP-ABE-ACS}$ is also secure.*

Proof. It follows the similar proof as in [56]. The five games Gm_i $(i = 0, 1, 2, 3, 4)$ are considered. In a game Gm_i, an adversary $\mathcal{A}$ tries to guess a correct bit b through the *Test* query. This event is defined by S_i and its corresponding probability is denoted by $Pr[S_i]$.

- $\boldsymbol{Gm_0}$**:** The initial game Gm_0 is considered to be identical with the actual protocol executing under the ROR model. Hence, we have,

$$Adv_{\mathcal{A}}^{CP-ABE-ACS}(t) = |2Pr[S_0] - 1|. \tag{4}$$

- $\boldsymbol{Gm_1}$**:** This game models an eavesdropping attack which invokes *Send*, *Test*, *Execute*, *Reveal*, and *Corrupt* queries with respect to the proposed scheme and also considers lists L_H, L_A and L_T for storing results of various queries. Leveraging the indistinguishability of Gm_0 and Gm_1, we obtain,

$$Pr[S_1] = Pr[S_0]. \tag{5}$$

- $\boldsymbol{Gm_2}$**:** This game accounts for the collision probability of various hash (H) queries. U and Dev use current timestamps TS_U along with hash function H_1 and TS_{Dev} along with hash function H_2, respectively, in the messages $M_1 = \langle dynamic\_token, Q_U, TS_U\rangle$ and $M_2 = \langle R_m, K_{1m}, K_{2m}, cert, TS_{Dev}\rangle$. Thus, applying the birthday paradox, we calculate that H query results in collision with probability $\frac{q_H^2}{2^{l_H+1}}$. Hence, we obtain,

$$|Pr[S_2] - Pr[S_1]| \le q_H^2/2^{l_H+1}. \tag{6}$$

- $\boldsymbol{Gm_3}$**:** With collision due to H hash query covered in Gm_2, we compute for the collision probability from the remaining queries in this game. The simulation of *Execute* and *Send* queries is given in Table 3. We have the following two cases:
Case 1. After executing $Send(Dev, M_1)$ query on $M_1 = \langle dynamic\_token, Q_U, TS_U\rangle$, it is noted that $dynamic\_token = (DID_U \parallel IDTS) \oplus H_1(TS_U \parallel Q_{dLu})$ contains $H_1(TS_U \| Q_{dLu}) \in L_{\mathcal{A}}$ which has collision probability at most $\frac{q_H}{2^{l_H}}$. To launch attack successfully, $H_2(ID_U\|PW_U\|\sigma_U)$ of IPB'_U, $H_2(PW_U \|\sigma_U)$ of r_U, $H_2(r_U \|PW_U \|\sigma_U)$ of EID_U and $H_2(ID_{Dev} \|TS_U)$ of $IDTS$ should be revealed to $\mathcal{A}$. The resultant collision probability totals up to $\frac{5q_H}{2^{l_H}}$.
Case 2. Considering $\mathcal{A}$ executes the query $Send(U, M_2)$, and the condition $cert = H_2(SK \|TS_{Dev}) \in L_{\mathcal{A}}$ holds, the calculated probability becomes $\frac{q_H}{2^{l_H}}$. Moreover, Dev computes $dynamic\_token \oplus H_1(TS_U \parallel Q_{dLu})$ of $(DID_U \|IDTS)$, $H_2(ID_{Dev} \|TS_U)$ of $IDTS$ and $H_2(Q_{dLu} \parallel e(g, h)^{r_m})$ of the computed session key SK. The probability for this part is $\frac{4q_H}{2^{l_H}}$. It is worth noting that $\mathcal{A}$ requires the long term key LTK_{Dev} for the derivation of session key $SK = H_2(Q_{dLu} \parallel e(g, h)^{r_m})$ in order to calculate $Q_{dLu} = (Q_U)^{LTK_{Dev}}$ having the intercepted public Q_U in the message M_1 and also to calculate r_m. For this part, the addition probability turns out to be $Adv_{\mathcal{A}}^{DDHP}(t)$ with polynomial time execution t (see Definition 3).
Summing both the cases, we obtain,

$$|Pr[S_3] - Pr[S_2]| \le 9q_H/2^{l_H} + Adv_{\mathcal{A}}^{DDHP}(t). \tag{7}$$

- $\boldsymbol{Gm_4}$**:** In this game, using the *Corrupt* query $\mathcal{A}$ can attempt to guess user's password PW_U and biometric key σ_U. The maximum probability up to correctly guessing the biometric key σ_U is $\max\{q_s(\frac{1}{2^{l_b}}, \varepsilon_{bm})\}$ [56], [59], and that for password PW_U is $C' \cdot q_s^{s'}$ [57]. As without these guessing attacks, the games Gm_3 and Gm_4 are identical, we have,

$$|Pr[S_4] - Pr[S_3]| \le \max\left\{C' \cdot q_s^{s'}, q_s\left(\frac{1}{2^{l_b}}, \varepsilon_{bm}\right)\right\}. \tag{8}$$

Finally, $\mathcal{A}$ must guess the correct bit b. Thus, it is then clear that $Pr[S_4] = 1/2$.

Table 3
Simulation of Execute and Send queries.

Simulation of ***Execute***(U, Dev) query occurs in succession with simulation of *Send* queries as given below.
Compute $dynamic\_token$ and Q_U as given in Fig. 5.
U sends the message $M_1 = \langle dynamic\_token, Q_U, TS_U\rangle$ to Dev.
Compute R_m, K_{1m}, K_{2m}, and $cert$ as given in Fig. 5.
Dev sends the authentication message $M_2 = \langle R_m, K_{1m}, K_{2m}, cert, TS_{Dev}\rangle$ to U.
Note that $\langle dynamic\_token, Q_U, TS_U\rangle \leftarrow Send(U, \textbf{start})$, and $\langle R_m, K_{1m}, K_{2m}, cert, TS_{Dev}\rangle \leftarrow Send(Dev, \langle dynamic\_token, Q_U, TS_U\rangle)$.
Finally, M_1 **and** M_2 **are returned.**

Send query simulation is done as per the proposed scheme.
(a) On $Send(U, \textbf{start})$ query, U responses as follows.
Compute $dynamic\_token$, Q_U, and TS_U as in Fig. 5.
Output $M_1 = \langle dynamic\_token, Q_U, TS_U\rangle$.
(b) Over $Send(Dev, \langle dynamic\_token, Q_U, TS_U\rangle)$ query, Dev responds as follows.
Test if $|TS_U * - TS_U| \le \Delta T$, and if so, generate Q_{dLu} and $DID_U\|IDTS$. Verify if $H_2(ID_{Dev}\|TS_U) = IDTS$ holds. Terminate if verification fails. Moreover, Dev computes R_m, K_{1m}, K_{2m}, and $cert$, and output $M_2 = \langle R_m, K_{1m}, K_{2m}, cert, TS_{Dev}\rangle$.
(c) U answers $Send(U, \langle R_m, K_{1m}, K_{2m}, cert, TS_{Dev}\rangle)$ query as mentioned below.
Check if $|TS_{Dev}^* - TS_{Dev}| \le \Delta T$. If verification passes, compute W, U, V, ERM, and SK' as given in Fig. 5.
Finally, verify if $H_2(SK' \parallel TS_{Dev}) = cert$ holds. If verification fails, terminate the current session. Otherwise, compute and accept SK' as the session key.
On establishment of the shared session key SK $(= SK')$, both U and Dev **terminate the session.**

Applying the triangular inequality, we have,

$$\begin{aligned}|Pr[S_0] - \frac{1}{2}| &= |Pr[S_1] - Pr[S_4]| \\ &\leq |Pr[S_1] - Pr[S_2]| + |Pr[S_2] - Pr[S_4]| \\ &\leq |Pr[S_1] - Pr[S_2]| + |Pr[S_2] - Pr[S_3]| \\ &\quad + |Pr[S_3] - Pr[S_4]|. \end{aligned} \tag{9}$$

□

Eqs. (4) -(9) lead to the following result:

$$\begin{aligned}\frac{1}{2}Adv_{\mathcal{A}}^{CP-ABE-ACS}(t) &= |Pr[S_0] - \frac{1}{2}| \\ &\leq \frac{q_H^2}{2^{l_H+1}} + \frac{9q_H}{2^{l_H}} + Adv_{\mathcal{A}}^{DDHP}(t) \\ &\quad + \max\left\{C' \cdot q_s^{s'}, q_s\left(\frac{1}{2^{l_b}}, \varepsilon_{bm}\right)\right\}. \end{aligned} \tag{10}$$

By multiplying 2 to both sides of Eq. (10) and then rearranging the terms, we obtain $Adv_{\mathcal{A}}^{CP-ABE-ACS}(t) \leq \frac{q_H^2+18q_H}{2^{l_H}} + 2[\max\{C' \cdot q_s^{s'}, q_s(\frac{1}{2^{l_b}}, \varepsilon_{bm})\} + Adv_{\mathcal{A}}^{DDHP}(t)]$.

6.3. Informal security analysis

Through informal security analysis, we demonstrate that the proposed scheme is secure against various known attacks.

6.3.1. Replay attack

Assume that an adversary $\mathcal{A}$ replays the message M_1 to *Dev*. *Dev* will then check the time stamp TS_U attached with M_1 and reject the message as the timestamp validation will fail. In the unlikely scenario, the message M_1 can be replayed within the time window ΔT by $\mathcal{A}$, and in that case $\mathcal{A}$ will receive the response message $M_2 = \langle R_m, K_{1m}, K_{2m}, cert, TS_{Dev}\rangle$. However, without the knowledge of the secret key k_U, in addition, a proper set of attributes, $\mathcal{A}$ cannot proceed further. Thus, the replay attack is protected by the proposed scheme.

6.3.2. Forgery attack

If $\mathcal{A}$ attempts to forge the message M_1 to *Dev*, $\mathcal{A}$ cannot compose *dynamic_token* as he/she lacks access to ID_{Dev}, which is made available only to the valid users through *DeviceList* saved in their smart cards. Furthermore, without proper attributes $\mathcal{A}$ cannot use M_2 to derive *SK*. Similarly, $\mathcal{A}$'s attempts to forge the message M_2 will also fail as it is not possible to construct *cert* without *Dev*'s long term secret key LTK_{Dev}. Thus, the proposed scheme is secure against forgery attacks.

6.3.3. Denial-of-Service attack

Let an adversary $\mathcal{A}$ attempt to execute a denial-of-service attack by repeatedly forging the messages M_1 and M_2 to *Dev* and *U*, respectively. But the condition $H_2(ID_{Dev}\|TS_U) \neq IDTS$ can detect the forgery and terminate execution. Thus, the proposed scheme is secure against denial-of-service attacks.

6.3.4. Impersonation attack

Let $\mathcal{A}$ attempt to impersonate a legally registered user *U* and intercept the message M_1. Since $M_1 = \langle dynamic\_token, Q_U, TS_U\rangle$, and Q_U & TS_U are encapsulated with *dynamic_token*, which is itself secured with secret K_U, $\mathcal{A}$ can neither recover EID_j nor modify M_1 without forging the entire message. This means that $\mathcal{A}$ cannot impersonate *U*. Similarly, if $\mathcal{A}$ attempts to impersonate the smart device *Dev*, he/she will fail in generating Q_{dLu}, which is required to form the session key *SK*. Furthermore, since the *GWN* plays no role in the session key establishment of the login and access control phase, it cannot be impersonated too. Thus, the proposed scheme also is also secure against impersonation attacks.

6.3.5. Man-in-the-Middle attack

If $\mathcal{A}$ attempts to intercept the message M_1 and then send the modified M_1 to *Dev*, the message $M_1 = \{dynamic\_token, Q_U, TS_i\}$ has Q_U and TS_U which are encapsulated with *dynamic_token*, whereas *dynamic_token* is itself secured with secret k_U. $\mathcal{A}$ cannot then modify M_1. Similarly, the attempts by $\mathcal{A}$ to intercept and modify other message M_2 will fail. Thus, the proposed scheme is secure against man-in-the-middle attack.

6.3.6. Stolen smart card and off-line guessing attacks

Let an adversary $\mathcal{A}$ possess the lost or stolen smart card SC_U of a legal registered user *U*, and through power analysis attacks [45,46] extract the credential values EID_U^*, g^{EID_U*}, g^{s_U*}, *DeviceList**, r_U^*, IPB_U and τ_U from SC_U. Out of these, except IPB_U and τ_U, none is stored in plaintext and it requires combination of the secret identity, password and biometric to retrieve r_U, EID_U, g^{EID_U}, g^{s_U*} and *DeviceList*. Note that IBP_U is the one-way hash of the secret identity ID_U, password PW_U and biometric secret key σ_U. $\mathcal{A}$ through off-line guessing attack will have to simultaneously guess ID_U, PW_U and σ_U, which is computationally infeasible. Thus, the proposed scheme is secure against lost or stolen smart card as well as off-line guessing attacks.

6.3.7. Privileged-Insider attack

Suppose an adversary $\mathcal{A}$ being a privileged-insider user of the *GWN* knows the registration information EID_U during a user *U*'s registration phase, where $EID_U = h(ID_U \|r_U)$. Further, assume that $\mathcal{A}$ has subverted the smart card SC_U of *U* and recovered the stored values including EID_U^*, r_U^* and IPB_U, where $EID_U^* = EID_U \oplus h(r_U \|PW_U \|\sigma_U)$, $r_U^* = r_U \oplus h(PW_U \|ID_U \|\sigma_U)$ and $IPB_U^{new} = h(ID_U \|PW_U \|\sigma_U)$. However, it is computationally infeasible to guess any of the secret values from these information. Thus, the proposed scheme is secure against privileged-insider attack.

6.3.8. Perfect forward secrecy

Let $\mathcal{A}$ somehow know the session key $SK = H_2(Q_{dLu} \,\|\, e(g,h)^{r_m})$ in a session. But, none of k_U, k_{Dev}, LTK_{Dev} or $MSK = \{\alpha, K_1, K_2\}$ can be learned from *SK*. Furthermore, since *SK* is composed of both long term as well as short term secrets, all other past and future session keys remain secure due to usage of long term secret keys LTK_{Dev} and *MSK*, and short term random Q_{dLu}. Thus, the proposed scheme provides perfect forward secrecy.

6.3.9. Ephemeral secret leakage (ESL) attack

Let an CK-adversary $\mathcal{A}$ somehow learn one or both of the session specific secrets (k_U, k_{Dev}). Since the session key $SK = H_2(Q_{dLu} \,\|\, e(g,h)^{r_m})$ depends on short term secret k_U, and long term secrets k_{Dev} and LTK_{Dev}, *SK* remains secure if not both short term and long term secrets are not compromised. Even if $\mathcal{A}$ learns the long term key LTK_{Dev} of the *Dev*, $\mathcal{A}$ still cannot subvert the session key *SK* as depends both on short term secrets k_U and k_{Dev} too. Thus, the proposed scheme is secure against ESL attack.

6.3.10. Stolen verifier attack

The *GWN* does not maintain any verification of specific information as it is not involved in the session key establishment procedure during the login and access control phase. A device *Dev* does not store any verifier table as a legal user *U* is granted access based on his/her attributes (ability to reconstruct $e(g,h)^{r_m}$). Thus, the proposed scheme is also secure from the stolen verifier attack.

6.3.11. Anonymity, untractability and collusion resistance

Suppose an adversary $\mathcal{A}$ eavesdrops the messages $M_1 = \{dynamic\_token, Q_U, TS_i\}$ and $M_2 = \langle R_m, K_{1m}, K_{2m}, cert, TS_{Dev}\rangle$.

None of the eavesdropped values contain any identifying information for the user U or the smart device Dev except for $dynamic\_token$. $dynamic\_token$ is itself composed of dynamic identity DID_U encapsulated with random nonce and timestamp. Thus, the proposed scheme provides user anonymity. In addition, all of the eavesdropped values are either composed of some random nonces and/or some timestamps, and consequently, these are always unique across different sessions, which make $\mathcal{A}$'s task difficult for tracing U. As a result, the proposed scheme also provides user untractability property.

In our scheme, the secret credentials of the user and device are generated using the CP-ABE key generation technique. The proposed CP-ABE is secure against collusion attack, that is, even we combine all the keys generated by the CP-ABE technique, the task of generation of a new key is still a computationally hard problem for the adversary (see sections 4.5 and 6.1). Because the proposed scheme is based on the CP-ABE technique, the scheme is naturally secure against the collusion attack.

6.4. Formal security verification through AVISPA

AVISPA [19] is a push-button tool for automated verification of security protocols. AVISPA can discover the presence of man-in-the-middle and replay attacks in a security protocol. In order to evaluate a protocol, it must be modeled in HLPSL (High Level Protocol Specification Language) [60]. The HLSPL is then translated to Intermediate Format (IF), which is further evaluated to produce the Output Format (OF) by one of the available four backends: 1) “On-the-fly Model-Checker (OFMC)”, 2) “Constraint Logic based Attack Searcher (CL-AtSe)”, 3) “SAT-based Model-Checker (SATMC)” and 4) “Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP)”. More detailed descriptions of these back-ends can be further found in [19].

HLPSL is a role-oriented language where each entity in the network is modeled as a basic role. In addition, we have two mandatory composite roles, namely session and goal & environment, which represent different scenarios involving basic roles. The OF has the following sections [60]:

- SUMMARY signifies if the tested protocol is safe, unsafe, or if the analysis is inconclusive.
- DETAILS explain why the tested protocol is concluded as safe, or under what criteria the tested protocol is exposed to an attack, or why the analysis leads to an inconclusive result.
- PROTOCOL defines the HLPSL specification of the tested protocol in intermediate form.
- GOAL is being performed by AVISPA using HLPSL specifications.
- BACKEND is the name of the back-end that is used for the analysis.
- The trace of possible vulnerability to the tested protocol is provided, if any, along with some useful statistics and relevant comments.

We modeled the proposed scheme in HLPSL by specifying roles for the mobile user (defined in Fig. 7), smart device (defined in Fig. 8) and the gateway node (defined in Fig. 9), along with the compulsory roles for session, environment and goals (defined in Fig. 10). Consider Fig. 7 which shows the HLPSL role specification for a mobile user U. In this role, we have implemented the user registration, login and access control phases. The user registration process takes place via secure communication with a pre-shared key between the mobile user U and the GWN. The login and access control phase is implemented via public channel. During the login and access control phase, the user U first sends the message $M_1 = \langle dynamic\_token, Q_U, TS_U \rangle$ to the smart device Dev via open channel. After receiving the message M_1, Dev sends the message $M_2 = \langle R_m, K_{1m}, K_{2m}, cert, TS_{Dev} \rangle$ to U via open channel.

```
role mobileuser(MU, SD, GWN : agent,
         H: hash_func,
           %simulating secure channel
         SecureChannel: symmetric_key,
         SND, RCV: channel(dy))
     % SD: smart device, MU: mobile user, GWN: gateway node
played_by MU
def=
local
State: nat,
Ru, IDu, EIDu, Su, Pw, Sig, IPB, TSu, IDdev, QdevL: text,
Ku, Qu, Qdlu, IDTS, DynamicToken, DIDu, SK, Cert, TSdev: text,
K1FRm, K2FRm, K1h, K2h, Rm, RRm, EIDuG, SuG, Qdevl: text,
ERM, K1m, K2m, GEIDuG, GSuG: text
init
State := 0
transition
% User registration phase
1. State = 0  ∧ RCV(start) =|>
   State' := 3 ∧ Ru' := new()
∧ IDu' := new() ∧ EIDu' := H(IDu'.Ru')
      % Send registration request to the GWN via secure channel
∧ SND({EIDu'}_SecureChannel)
∧ secret(EIDu', sEIDu, {MU,GWN})
         ∧ secret(IDu', sIDu, MU)
         ∧ secret(Ru', sRu, MU)
% Receive smart card from the GWN securely
2. State = 3 ∧ RCV({EIDu. exp(g,EIDu).exp(g,Su).
                    IDdev.QdevL}_SecureChannel) =|>
   State' := 4 ∧ Pw' := new()
     ∧ Sig' := new()
     ∧ IPB' := H(IDu.Pw'.Sig')
     ∧ secret(IDdev, sIDdev, {MU,GWN,SD})
     ∧ secret(Qdevl, sQdevl, {MU,GWN,SD})
     ∧ secret(EIDuG, sEIDuG, {MU,GWN})
     ∧ secret(SuG, sSuG, {MU,GWN})
     ∧ secret(Pw', sPw, MU)
          ∧ secret(Sig', sSig, MU)
          ∧ secret(IPB', sIPB, MU)

% Login and access control phase
3. State = 4 ∧ RCV(H(g)) =|>
   State' := 6∧ TSu' := new() ∧ Ku':= new()
∧ Qu':= exp(g, Ku') ∧ Qdlu' := exp(QdevL, Ku')
∧ IDTS' := H(IDdev. TSu')
      % simulating  DID_u = RNG_EID_u(TSu)
∧ DIDu' := H(EIDu'.TSu' )
∧ DynamicToken' := xor(DIDu'.IDTS',H(TSu'.Qdlu'))
      % Send message M1 to smart device Dev via open channel
∧ SND( DynamicToken'.Qu'.TSu')
% User has freshly generated the value DynamicToken
        ∧ witness (MU, SD, sDynamicToken, DynamicToken')

      % Receive message M2 from smart device Dev via open channel
4. State = 6 ∧ RCV( exp(g,Rm).exp(K1h,Rm).exp(K2h,Rm).
                    H(H(Qdlu.ERM).TSdev).TSdev) =|>
      % simulating ABE
   State':= 8 ∧ ERM' := H(K1m'.K2m'.exp(RRm, EIDuG).exp(RRm, SuG))
    ∧ SK' := H( Qdlu.ERM' )
    ∧ Cert' := H(SK'.TSdev)
     ∧ request(SD, MU, sERM, ERM')
     ∧ request(SD, MU, sCert, Cert')
     ∧ request(SD, MU, sSK, SK')
end role
```

Fig. 7. HLPSL role specification for a mobile user.

The declaration “secrecy_of” defines the parameters which will be kept secret to which agents. The declarations “witness” and “request” respectively define the weak and strong authentication. The HLPSL implements the Dolev-Yao threat model [44] and as a result, AVISPA validates a tested security protocol whether it is safe against “replay attack” and “man-in-the-middle attack”. In AVISPA, an intruder (always denoted by the notation (i)) also takes an active part during the communication. We have applied the popular OFMC and CL-AtSe back-ends for formal security verification under the “SPAN, the Security Protocol ANimator for AVISPA” [61]. We have ignored the results under the SATMC and TA4SP back-ends, because at present the HLPSL implementation does not sup-

```
role smartdevice (MU, SD, GWN: agent,
           H : hash_func,
           SecureChannel: symmetric_key,
           SND, RCV: channel(dy))
      % SD: smart device, MU: mobile user, GWN: gateway node
played_by SD
def=
local
  State: nat,
  LTKdev, K1, K2,  Su, EIDu, TSu, TSdev, Rm, RRm, K1m, K2m, K1h, K2h: text,
  GEIDuG,  GSuG, IDdev, Ku, Qu, Qdevl, ERM,  SK, Cert, DynamicToken: text
init
  State := 2
transition
% Device enrolement and precomputation phase
       % Recieve preloaded information from the GWN securely
1. State =2 /\ RCV({LTKdev.exp(g,exp(g,EIDu)).exp(g,exp(g,Su)).
                  exp(h,K1).exp(h,K2)}_SecureChannel) =|>
  State':=5 /\ Rm' := new() /\ RRm' := exp(g, Rm')
   /\ K1m':= exp(K1h, Rm') /\ K2m':= exp(K2h, Rm')
        % simulating ABE
   /\ ERM' := H(K1m'.K2m'.exp(GEIDuG, Rm').exp(GSuG, Rm') )
   /\ SND(H(g))
% Login and access control phase
       % Receive message M1 from user via open channel
2. State = 5 /\ RCV(xor(H(EIDu.TSu ).H(IDdev. TSu),
                  H(TSu.exp(Qdevl, Ku))). exp(g, Ku).TSu ) =|>
  State':= 8 /\ request(SD, MU, sDynamicToken, DynamicToken)
    /\ TSdev' := new() /\ Qdevl' := exp(Qu, LTKdev)
    /\ SK' := H(Qdevl' . ERM)
    /\ Cert' := H(SK'.TSdev')
    /\ secret(Rm', sRm, SD)
       % Send message M2 to mobile user via open channel
    /\ SND(RRm'. K1m . K2m . Cert'. TSdev')
% SD has freshly calculated the value SK', Cert' for mobile user
    /\ witness (SD, MU, sSK, SK')
    /\ witness (SD, MU, sCert, Cert')
    /\ witness (SD, MU, sERM, ERM)
end role
```

Fig. 8. HLPSL role specification for a smart device.

```
role gatewaynode(MU, SD, GWN: agent,
           H : hash_func,
           SecureChannel : symmetric_key,
           SND, RCV: channel(dy))
      % SD: smart device, MU: mobile user, GWN: gateway node
played_by GWN
def=
local
State : nat,
IDu, Ru, EIDu, Su, EIDuG, SuG, IDdev, LTKdev: text,
Qdevl, K1, K2, GEIDuG,  GSuG,K1h, K2h: text
init
    State := 1
transition
% Mobile user registration phase
      % Received registration request from MU securely
1. State = 1 /\ RCV({ H(IDu.Ru) }_SecureChannel) =|>
  State':= 7 /\ secret(EIDu, sEIDu, {MU,GWN})
          /\ secret(IDu, sIDu, MU)
      % the computations at the attribute authorities abstracted
          /\ Su' := new()
     /\ EIDuG' := exp(g, EIDu)
     /\ SuG' := exp(g, Su)
     /\ IDdev' := new() /\ LTKdev' := new()
     /\ Qdevl' := exp(g, LTKdev')
      % Send smart card to user securely
     /\ SND({EIDu.EIDuG'.SuG'.IDdev.Qdevl}_SecureChannel )
     /\ secret(IDdev, sIDdev, {MU,GWN,SD})
     /\ secret(Qdevl, sQdevl, {MU,GWN,SD})
     /\ secret(EIDu, sEIDu, {MU,GWN})
     /\ secret(EIDuG', sEIDuG, {MU,GWN})
     /\ secret(SuG', sSuG, {MU,GWN})
     /\ secret(Su, sSu, GWN)
% Device enrolement and precomputation phase
     /\ K1' := new() /\ K2' := new()
      % GEIDuG corresponds to e(g,h)^{r_m), simulating ABE
     /\ GEIDuG' := exp(g, EIDuG)
     /\ GSuG' := exp(g, SuG)
     /\ K1h' := exp(h, K1') /\ K2h':= exp(h, K2')
     /\ SND({LTKdev. GEIDuG'.GSuG'.K1h'.K2h' }_SecureChannel)
     /\ secret(K1', sK1, GWN)
     /\ secret(K2', sK2, GWN)
     /\ secret(LTKdev, sLTKdev, {SD,GWN})
end role
```

Fig. 9. HLPSL role specification for the *GWN*.

```
%%% Role specification for the session %%%

role session (MU, SD, GWN: agent,
    H: hash_func,
    SecureChannel: symmetric_key)
def=
  local S1, R1, S2, R2, S3, R3 : channel (dy)
composition
        mobileuser(MU, SD, GWN, H, SecureChannel, S1, R1)
     /\ smartdevice(MU, SD, GWN, H, SecureChannel, S2, R2)
     /\ gatewaynode(MU, SD, GWN, H, SecureChannel, S3, R3)
end role

%%% Role specification for the goal and environment %%%

role environment()
def=
  const mu, sd, gwn: agent,
secureChannel: symmetric_key,
     h1: hash_func,
     g, h: text,
     sSK, sCert, sERM, sDynamicToken, sEIDu, sIDu,
     sPw, sEIDuG, sLTKdev, sRu, sQdevl, sSig, sIPB,
     sSuG, sIDdev, sSu, sRm, sK1, sK2: protocol_id
     intruder_knowledge = {mu, sd, gwn, g, h}
  composition
   session(mu, sd, i,  h1, secureChannel)
/\  session(mu, i, gwn, h1, secureChannel)
   /\  session(i, sd, gwn, h1, secureChannel)
   /\  session(mu, sd, gwn, h1, secureChannel)
end role

goal
  % Confidentiality (privacy)
  secrecy_of sEIDu, sIDu, sPw, sEIDuG, sLTKdev, sRu
  secrecy_of sQdevl, sSig, sIPB, sSuG, sIDdev
  secrecy_of sSu, sRm, sK1, sK2

  % Authentication (replay and man-in-the-middle attacks)
  authentication_on sSK, sERM
  authentication_on sCert, sDynamicToken
end goal

environment()
```

Fig. 10. HLPSL role specification for the session, goal and environment.

SUMMARY **SAFE** DETAILS BOUNDED_NUMBER_OF_SESSIONS PROTOCOL /home/soumya/span/testsuite /results/AVISPA_IoT_ABE.if GOAL as specified BACKEND OFMC STATISTICS TIME 72 ms parseTime 0 ms visitedNodes: 8 nodes depth: 3 plies	SUMMARY **SAFE** DETAILS BOUNDED_NUMBER_OF_SESSIONS TYPED_MODEL PROTOCOL /home/soumya/span/testsuite /results/AVISPA_IoT_ABE.if GOAL As specified BACKEND CL–AtSe STATISTICS Analysed : 0 state Reachable : 0 state Translation: 0.08 seconds Computation: 0.00 seconds

Fig. 11. AVISPA simulation results under OFMC and CL-AtSe back-ends.

port bitwise XOR operations, and AVISPA produces the simulation results under SATMC and TA4SP backends as "inconclusive". The simulation results presented in Fig. 11 clearly show that the proposed scheme is free from man-in-the-middle and replay attacks.

7. Performance and comparative study

In this section, we present a comparative study of ABE schemes in terms of features and another comparative study of authentication and access control scheme in terms of computation and

Table 4
Comparison of ABE schemes.

Scheme	I_1	I_2	$\|Key\|$	$\|C\|$	I_3	I_4	I_5
Sarai et al. [8]	KP	Threshold	$\mathcal{O}(n_u)$	$\mathcal{O}(n)$	×	×	×
Bethencourt et al. [10]	CP	Tree	$\mathcal{O}(n_u)$	$\mathcal{O}(n)$	×	×	×
Chase and Chow [16]	KP	Tree	$\mathcal{O}(n_u)$	$\mathcal{O}(n)$	×	✓	×
Aattrapadung et al. [11]	KP	LSSS	$\mathcal{O}(n_u)$	$\mathcal{O}(1)$	×	×	×
Yu et al. [34] (FDAC)	KP	Tree	$\mathcal{O}(n_u)$	$\mathcal{O}(n)$	×	×	×
Ruj et al. [35]	KP	Tree	$\mathcal{O}(n_u)$	$\mathcal{O}(n)$	×	×	×
Guo et al. [14]	CP	AND	$\mathcal{O}(1)$	$\mathcal{O}(n-n_u)$	×	×	×
Liu et al. [40]	CP	Tree	$\mathcal{O}(n_u)$	$\mathcal{O}(n)$	×	×	×
Odelu et al. [15]	CP	AND	$\mathcal{O}(1)$	$\mathcal{O}(1)$	×	×	×
He [39] (FLAC)	CP	Tree	$\mathcal{O}(n_u)$	$\mathcal{O}(n_u)$	×	×	✓
Li et al. [41]	CP	$AND+W$	$\mathcal{O}(n_u)$	$\mathcal{O}(n)$	✓	✓	×
Belguith et al. [42] (PHOABE)	CP	LSSS	$\mathcal{O}(n_u)$	$\mathcal{O}(n)$	✓	✓	✓
Proposed	CP	AND	$\mathcal{O}(1)$	$\mathcal{O}(1)$	✓	✓	×

Note: ✓: the scheme supports a feature; × : the scheme does not support a feature; I_1: whether KP-ABE or CP-ABE; I_2: type of access structure; *LSSS*: linear secret sharing scheme, $AND+W$ AND with wildcards, $|Key|$: key size; $|C|$: ciphertext size for unit length planetext; I_3: whether policy hidden scheme; I_4: whether supports multi attribute authority; I_5: designed to support computation outsourcing; n: is the number of attributes in the universe, n_u: is the number of attributes defined for the user.

communication overheads, and security and functionality features among the proposed scheme and other related authentication and access control schemes,

7.1. Comparative study on ABE schemes

Table 4 summarizes some of the more significant ABE schemes with respect to functionality features. We observe that the underlying ABE scheme proposed in this paper is rather versatile. In terms of its limitations, the proposed ABE scheme is not designed to support the outsourcing of computation load, like PHOABE [42] and FLAC [39]. But this is intentional, as we outlined in the problem statement, we designed the scheme to have low enough computation overhead, that it should not require any supporting cloud infrastructure. A more pressing limitation of the scheme is that it supports only *AND* access structure instead of the much more versatile non-monotonic access structures in [11]. In this paper, we focused on scalability over expressiveness, but we aim to address this limitation in future works.

7.2. Comparative study on authentication and access control schemes

In this section, we benchmark the proposed access control scheme with related access control and authentication schemes for IoT or WSN deployments. We consider the fine grained access control schemes for WSNs proposed by Yu et al. [34], Ruj et al. [35], Chatterjee and Das [37] and Banerjee et al. [43]. The schemes [34] and [35] are selected for their historical importance, whereas the scheme [37] and [43] are selected as they are close to the proposed scheme in terms of scope. We also include the schemes by Zhou et al. [27], Challa et al. [26] Chang et al. [62] and Banerjee et al. [28] as they are recent authentication schemes for IoT environment. In the scheme [62], we only consider the ECC-based version of their schemes. Banerjee et al. [32] is included in this study for its unique design goal of providing anonymity even from the registration server

We compute the communication overhead of the schemes by assuming identity and hash output are 160 bits each, an ECC point $P=(P_x,P_y)$ requires $(160+160)=320$ bits assuming the prime q to be 160 bits to provide reasonable security while it is compared with an 1024-bit RSA public key cryptosystem, random nonce is 128 bits and timestamp is 32 bits. For fine grained access control schemes, where necessary, we assumed the number of attribute for each user to be 9. For the proposed scheme, total communication overhead due to exchange of two messages M_1 and M_2 is $|M_1|+|M_2|=(320+160+32)+(160+160+160+160+32)=(512+672)=1,184$ bits. The communication costs comparison among the schemes summarized in Table 5 shows that the proposed scheme has the lowest communication overhead among the compared schemes.

Table 5
Communication costs comparison.

Scheme	No. of bits	No. of messages
Proposed	1184	2
Yu et al.[34]	3019	3
Ruj et al.[35]	3019	3
Chatterjee and Das [37]	2144	5
Banerjee et al. [43]	3584	2
Zhou et al. [27]	5856	3
Challa et al. [26]	2528	3
Chang et al. [62]	2272	2
Banerjee et al. [32]	3648	4
Banerjee et al. [28]	2048	3

The computation costs comparison among the schemes are summarized in Table 6. In Table 6, the notations T_h, T_{sym}, T_{em}, T_{bp} and T_f signify the time needed to compute one-way hashing, symmetric encryption, elliptic curve point multiplication, bilinear pairing and fuzzy extractor operations, respectively. Based on the experimental results reported for the user's device in [64], we have $T_{em}\approx 13.405$ ms, $T_{ea}\approx 3.297$ ms, $T_{sym}\approx 1.657$ ms, $T_h\approx 0.056$ ms, $T_{bp}\approx 32.713$ ms. We assume that $T_f\approx T_{em}$ [49]. Here, n represents the number of attributes in the universe and n_u represents the number of attributes defined for a user, which is assumed to be 9. For the schemes [34] and [35], we assume $m=1$. Though the proposed scheme needs more overall computation cost as compared to the schemes [27], [62] and [28], the computation cost for the resource limited smart device is lowest, which is only $4T_h\approx 0.224$ ms. As a result, the proposed scheme is efficient for the smart devices as compared to all existing compared schemes. Additionally, from Table 7, we can see that the proposed scheme provides much greater security and functionality features compared to other existing schemes [27] and [62].

The schemes proposed in [28] and [43] provide comparable security & functionality features with the proposed scheme. However, the scheme proposed in[43] is limited in term of scalability as seen from tables 5 and 6. It does not support multiple attribute authorities, which is a necessity for an IoT environment. The scheme proposed in [28] is computationally very light and is unique in providing resistance to stolen device impersonation at-

Table 6
Computation costs comparison.

Scheme	U	GWN	Dev	Total cost
Proposed	$10T_h + 2T_{em} + 3T_{bp} + T_f$ ≈ 100.297 ms	– –	$4T_h$ ≈ 0.224 ms	$14T_h + 2T_{em} + 3T_{bp} + T_f$ ≈ 100.521 ms
Yu et al. [34]	$4T_{bp}$ ≈ 53.62 ms	– –	$mT_h + (n_u + 1)T_{em} + mT_{sym}$ ≈ 135.706 ms	$4T_{bp} + mT_h + (n_u + 1)T_{em} + T_{sym}$ ≈ 189.326 ms
Ruj et al. [35]	$4T_{bp}$ ≈ 53.62 ms	– –	$mT_h + (n_u + 1)T_{em} + T_{sym}$ ≈ 135.706 ms	$4T_{bp} + mT_h + (n_u + 1)T_{em} + T_{sym}$ ≈ 189.326 ms
Chatterjee and Das [37]	$7T_h + T_{sym} + 3T_{ea} + 4T_{em}$ ≈ 65.503 ms	$2T_h + 2T_{sym} + T_{em}$ ≈ 16.717 ms	$5T_h + 2T_{sym} + (n_u + 1)T_{ea}$ ≈ 36.45 ms	$14T_h + 5T_{sym} + (n_u + 4)T_{ea} + 5T_{em}$ ≈ 118.67 ms
Banerjee et al. [43]	$9T_h + T_{sym} + 2T_{em} + 10T_{bp} + T_f$ ≈ 195.676 ms	– –	$4T_h + 10T_{em} + 1T_{bp}$ ≈ 147.679 ms	$13T_h + T_{sym} + 12T_{em} + 11T_{bp} + T_f$ ≈ 343.355 ms
Zhou et al. [27]	$10T_h$ ≈ 0.56 ms	$7T_h$ ≈ 0.392 ms	$19T_h$ ≈ 1.064 ms	$36T_h$ ≈ 2.016 ms
Challa et al. [26]	$5T_h + 5T_{em} + T_f$ ≈ 54.796 ms	$4T_h + 5T_{em}$ ≈ 67.249 ms	$3T_h + 4T_{em}$ ≈ 40.383 ms	$12T_h + 14T_{em} + T_f$ ≈ 174.937 ms
Chang et al. [62]	$7T_h + 2T_{em}$ ≈ 27.202 ms	$9T_h$ ≈ 0.504 ms	$5T_h + 2T_{em}$ ≈ 27.09 ms	$21T_h + 4T_{em}$ ≈ 54.796 ms
Banerjee et al.[32]	$7T_h + 3T_{sym} + 4T_{em} + 1T_f$ ≈ 91.527 ms	$2T_h + 2T_{sym} + 3T_{em}$ ≈ 43.527 ms	$4T_h + 3T_{sym} + 2T_{em}$ ≈ 31.834 ms	$13T_h + 8T_{sym} + 9T_{em} + 1T_f$ ≈ 166.885 ms
Banerjee et al.[28]	$17T_h + T_f$ ≈ 33.664 ms	$8T_h$ ≈ 17.053 ms	$6T_h + T_f$ ≈ 33.048 ms	$31T_h + 2T_f$ ≈ 83.765 ms

tack. But, this is achieved at cost of communication and storage overheads on the device side. It requires periodic communication between the smart device and the gateway node in order to renew the PUF challenge-response pair essential for authentication purpose. Additionally, the scheme proposed in [28] is an authentication scheme and it does not support fine grained access control.

The proposed scheme has a very low computational overhead due to only four one-way hash operations, which is the lowest among comparable schemes for the resource-constrained IoT devices. This property when considered along with the breadth of features supported by the schemes makes the proposed scheme ideally suitable for IoT deployment. The low computation overhead

Table 7
Security & functionality features comparison.

Features Proposed	[34]	[35]	[37]	[43]	[27]	[26]	[62]	[32]	[28]
☑	NA	NA	☑	☑	☑	☑	☑	☑	☑
☑	NA	NA	☑	☑	☑	☑	☒	☑	☑
☑	NA	NA	☑	☑	☑	☑	☒	☑	☑
☑	☒	☒	☑	☑	☑	☑	☒	☑	☑
☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
☑	☒	☒	☒	☑	☑	☑	☑	☑	☑
☑	NA	NA	☑	☑	☑	☑	☑	☑	☑
☑	NA	NA	☑	☑	NA	☑	☒	☑	☑
☑	NA	NA	☑	☑	NA	☑	☑	☑	☑
☑	☒	☒	☑	☑	☑	☑	☒	☑	☑
☑	☒	☑	☒	☑	☒	☑	☑	☑	☑
☑	☒	☒	☑	☑	☒	☑	☑	☑	☑
☑	☒	☒	☑	☑	☑	☑	☑	☑	☑
☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
☑	NA	NA	☒	☑	☑	☑	☒	☑	☑
3	NA	NA	2	☑	2	3	2	☑	☑
☑	NA	NA	☑	☑	☑	☑	☒	☒	☑
☑	NA	NA	NA	☑	NA	☑	☒	☒	☑
☑	NA	NA	☑	☑	NA	☑	☒	☑	☑
☑	☒	☒	☒	☑	☒	☑	☒	☑	☑
☑	NA	NA	☑	☑	NA	☑	☑	☑	☑
☒	NA	NA	☒	☒	NA	☒	☒	☒	☑
☑	☒	☒	☒	☒	☒	☒	☒	☑	☒
☑	☑	☑	☑	☑	☒	☒	☒	☒	☒
☑	☒	☒	☑	☑	☒	☑	☑	☑	☑
☑	☒	☒	☑	☑	☑	☑	☒	☑	☑

Note: ☑: the scheme is resilient against an attack or it supports a feature; ☒: the scheme is not secure against an attack or it does not support a feature; NA: not applicable to a scheme; DA_1: user anonymity; DA_2: user untraceability; DA_3: off-line password guessing attack; DA_4: fast detection of erroneous input; DA_5: mutual authentication; DA_6: session key agreement; DA_7: user impersonation attack; DA_8: gateway node impersonation attack; DA_9: device impersonation attack; DA_{10}: privileged-insider attack; DA_{11}: forward secrecy; DA_{12}: replay attack; DA_{13}: man-in-the-middle attack; DA_{14}: stolen verifier attack; DA_{15}: stolen smart card attack; DA_{16}: two or three factor authentication; DA_{17}: local password change; DA_{18}: local biometric change; DA_{19}: dynamic sensing node addition; DA_{20}: ESL attack under the CK-adversary model; DA_{21}: physical smart device capture attack; DA_{22}: stolen device impersonation attack; DA_{23}: supports anonymity from registration server. DA_{24}: supports fine-grained access control. DA_{25}: formal security analysis; DA_{26}: formal security verification using AVISPA [19] or ProVerif [63] software tool;

Table 8
Simulation parameters used in simulation.

Parameter	Description
Platform	Ubuntu 16.04 LTS
Tool used	NS3 (3.28)
Wireless protocol	802.11*p*
Simulation time	1200 seconds
Mobility	Random (0–3 m/s)
Routing protocol	Ad-hoc on-demand distance vector routing (AODV) [65]
Medium access control type	IEEE 802.11

Table 9
Various scenarios used in simulation.

Network scenarios	No. of users	No. of smart devices
1	5	20
2	8	20
3	5	35
4	10	20
5	8	35
6	10	35

is also achieved by pre-computing the expensive exponentiation operations. For devices with even more limited computation capability, this step can conceivably be outsourced to an assisting cloud service. However, such a workaround is beyond the scope of this discussion in this paper.

The proposed scheme provides complete anonymity and untracability for the user as discussed in Section 6.3.11. However, due to the usage of pre-computed values, device anonymity is not guaranteed. Furthermore, for the devices with adequate computational capability, when pre-computing is unnecessary, the proposed scheme provides anonymity and untracability for the devices as well. Overall, we conclude that the proposed scheme provides a better trade-off among the computation and communication overheads, and security & functionality features as compared to other related schemes.

8. Practical perspective: NS3 simulation

To study the practical perspective of the proposed scheme, we simulate the scheme under the standard network simulator, NS3 (3.28) [66]. We measure the network impact in terms of throughput and end-to-end delay and show the applicability of the proposed scheme under a real world environment. The simulation parameters are tabulated in Table 8. In our simulation, several users attempt to access multiple smart devices in the IoT environment. The smart devices are assumed to be stationary and accessible to a stationary gateway node. All the smart devices are deployed within 20–100 m from a gateway node. The users free to move randomly with a speed upto 3 m/s (meters per second) across an 150 square meters area centered upon the gateway node. All communications occur over the 2.4 GHz IEEE 802.11 wi-fi standard. We simulate several scenarios varying different number of users and smart devices. The simulation parameters are detailed in Table 9. All other parameters are considered as standard by default as used in NS3.

Fig. 12 (a) and 12(a) plot the network throughput and end-to-end delay, respectively. The network throughput is calculated as $\frac{N_{pkt} * size}{T_{total}}$, whereas the end-to-end delay is calculated as $\sum_{i=0}^{N_{pkt}} (T_{recv_i} - T_{send_i})/N_{pkt}$, where N_{pkt} is the total number of packets received, *size* is the size of the received packet (in bytes), and T_{total}, T_{send_i} and T_{recv_i} are the total time taken, and the time when the i^{th} packet was transmitted and received, respectively, in seconds [49]. From the simulation results, it is apparent that the network throughput increases with the increasing number of transmitted messages. Additionally, the end-to-end delay also increases with more number of transmitted messages. This can be attributed to the increased number of messages contributing to network congestion.

9. Concluding remarks

In this paper, we discussed the importance of a fine-grained access control mechanism for the IoT environment. We then described the non-centralized nature of the IoT environment and presented a model for this purpose. We presented a new, highly scalable, multi authority CP-ABE based anonymous access control scheme suitable for IoT architecture. We provided a detailed security analysis (formal and informal) for the presented scheme along with its formal security verification using the AVISPA tool. The detailed security analysis gives that the proposed scheme can resist various known attacks in the IoT environment. We also presented a thorough comparative study for the proposed access control scheme in IoT architecture and other related existing schemes. It was worth noting that the proposed scheme required less computation cost for resource-limited smart devices and also needed significantly less communication overhead as compared to other schemes too. Finally, we performed a simulation study to measure the network performance parameters for the presented scheme. Overall, the proposed scheme gives a better trade-off among the security and functionality features, and communication and computational overheads as compared to other schemes. Thus, the proposed scheme is suitable for practical application in the IoT environment.

Regarding the future scope of this work, the underlying ABE scheme can be further investigated in the IoT environment.

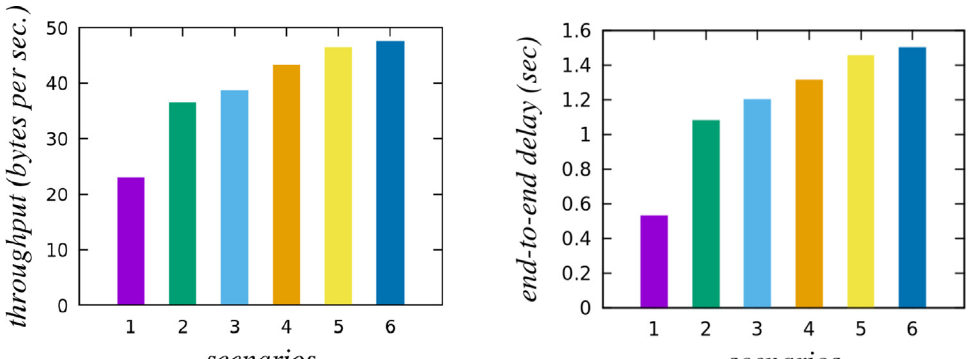

Fig. 12. (a) Throughput (bytes per second), (b) End-to-end delay (seconds).

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We would like to thank the anonymous reviewers and the associate editor for their valuable feedback on the paper, which helped us to improve its quality and presentation. This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Science, ICT & Future Planning (2017R1A2B1002147), in part by the BK21 Plus project funded by the Ministry of Education, Korea (21A20131600011); in part by FCT/MCTES through national funds and when applicable co-funded EU funds under the Project UIDB/EEA/50008/2020; and in part by Brazilian National Council for Scientific and Technological Development (CNPq) via Grant No. 309335/2017 − 5. This work was also supported by the Ripple Centre of Excellence Scheme, CoE in Blockchain (Sanction No. IIIT/R&D Office/Internal Projects/001/2019), IIIT Hyderabad, India.

References

[1] Atzori L, Iera A, Morabito G. The internet of things: a survey. Comput Networks 2010;54(15):2787–805.
[2] Gubbi J, Buyya R, Marusic S, Palaniswami M. Internet of things (iot): a vision, architectural elements, and future directions. Future Generation Computer Systems 2013;29(7):1645–60.
[3] Das AK, Zeadally S, He D. Taxonomy and analysis of security protocols for internet of things. Future Generation Computer Systems 2018;89:110–25.
[4] Carlin A, Hammoudeh M, Aldabbas O. Intrusion detection and countermeasure of virtual cloud systems-state of the art and current challenges. International Journal of Advanced Computer Science and Applications 2015;6(6):1–15.
[5] Ghafir I, Prenosil V, Alhejailan A, Hammoudeh M. Social engineering attack strategies and defence approaches. In: 4th International Conference on Future Internet of Things and Cloud (FiCloud); 2016. p. 145–9. Vienna, Austria
[6] Roman R, Najera P, Lopez J. Securing the internet of things. IEEE Computer 2011;44(9):51–8.
[7] Kolias C, Kambourakis G, Stavrou A, Voas J. DDOs in the iot: mirai and other botnets. Computer (Long Beach Calif) 2017;50(7):80–4.
[8] Sahai A, Waters B. Fuzzy identity-based encryption. In: Cramer R, editor. Annual International Conference on the Theory and Applications of Cryptographic Techniques – Advances in Cryptology Advances in Cryptology (EUROCRYPT'05); 2005. p. 457–73. Aarhus, Denmark
[9] Goyal V, Pandey O, Sahai A, Waters B. Attribute-based encryption for fine-grained access control of encrypted data. In: Proc. of 13th ACM conference on Computer and Communications Security (CCS'06); 2006. p. 89–98. Alexandria, VA, USA
[10] Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy (S&P'07); 2007. p. 321–34. Oakland, California, USA
[11] Attrapadung N, Herranz J, Laguillaumie F, Libert B, De Panafieu E, Ràfols C. Attribute-based encryption schemes with constant-size ciphertexts. Theor Comput Sci 2012;422:15–38.
[12] Emura K, Miyaji A, Nomura A, Omote K, Soshi M. A ciphertext-policy attribute-based encryption scheme with constant ciphertext length. In: International Conference on Information Security Practice and Experience (ISPEC'09), Lecture Notes in Computer Science, vol. 5451. Xi'an, China: Springer Berlin Heidelberg; 2009. p. 13–23.
[13] Zhang Y, Zheng D, Chen X, Li J, Li H. Computationally efficient ciphertext-policy attribute-based encryption with constant-size ciphertexts. In: International Conference on Provable Security (ProvSec'14), Lecture Notes in Computer Science, vol. 8782. Hong Kong: Springer International Publishing; 2014. p. 259–73.
[14] Guo F, Mu Y, Susilo W, Wong DS, Varadharajan V. CP-ABE With constant-size keys for lightweight devices. IEEE Trans Inf Forensics Secur 2014;9(5):763–71.
[15] Odelu V, Das AK, Rao YS, Kumari S, Khan MK, Choo KKR. Pairing-based CP-ABE with constant-size ciphertexts and secret keys for cloud environment. Computer Standards & Interfaces 2017;54:3–9.
[16] Chase M, Chow SS. Improving privacy and security in multi-authority attribute-based encryption. In: Proc. of 16th ACM Conference on Computer and Communications Security (CCS'09); 2009. p. 121–30. Chicago, Illinois, USA
[17] Wang D, Wang P. Two birds with one stone: two-factor authentication with security beyond conventional bound. IEEE Trans Dependable Secure Comput 2016;15(4):708–22.
[18] Abdalla M, Fouque PA, 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; 2005. p. 65–84. Les Diablerets, Switzerland
[19] AVISPA. Automated validation of internet security protocols and applications. 2020a. http://www.avispa-project.org/. Accessed on March 2020.
[20] Mineraud J, Mazhelis O, Su X, Tarkoma S. A gap analysis of internet-of-things platforms. Comput Commun 2016;89:5–16.
[21] Alqassem I. Privacy and security requirements framework for the internet of things (iot). In: 36th International Conference on Software Engineering (ICSE Companion'14); 2014. p. 739–41. Hyderabad, India
[22] Banerjee S., Odelu V., Chattopadhyay S.. A brief overview of User Authentication in Internet of Things architecture. IEEE; Kolkata, India; 2020. 4th International Conference on Computational Intelligence and Networks (CINE).
[23] Jeong J, Chung MY, Choo H. Integrated OTP-based user authentication scheme using smart cards in home networks. In: 41st Annual Hawaii International Conference on System Sciences (HICSS'08) Waikoloa, HI, USA; 2008. 294–294
[24] Hanumanthappa P, Singh S. Privacy preserving and ownership authentication in ubiquitous computing devices using secure three way authentication. In: International Conference on Innovations in Information Technology (IIT'12); 2012. p. 107–12. Abu Dhabi, United Arab Emirates
[25] Santoso FK, Vun NCH. Securing iot for smart home system. In: International Symposium on Consumer Electronics (ISCE'15); 2015. p. 1–2. Madrid, Spain
[26] Challa S, Wazid M, Das AK, Kumar N, Reddy AG, Yoon EJ, et al. Secure signature-based authenticated key establishment scheme for future iot applications. IEEE Access 2017;5:3028–43.
[27] Zhou L, Li X, Yeh KH, Su C, Chiu W. Lightweight iot-based authentication scheme in cloud computing circumstance. Future Generation Computer Systems 2019;91:244–51.
[28] Banerjee S, Odelu V, Das AK, Chattopadhyay S, Rodrigues JJ, Park Y. Physically secure lightweight anonymous user authentication protocol for internet of things using physically unclonable functions. IEEE Access 2019;7:85627–44.
[29] Shahzad M, Singh MP. Continuous authentication and authorization for the internet of things. IEEE Internet Comput 2017;21(2):86–90.
[30] Chuang YH, Lo NW, Yang CY, Tang SW. A lightweight continuous authentication protocol for the internet of things. Sensors 2018;18(4):1104.
[31] Turkanović M, Brumen B, Hölbl M. A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the internet of things notion. Ad Hoc Netw 2014;20:96–112.
[32] Banerjee S, Odelu V, Das AK, Chattopadhyay S, Kumar N, Park Y, et al. Design of an anonymity-preserving group formation based authentication protocol in global mobility networks. IEEE Access 2018;6:20673–93.
[33] He D, Bu J, Zhu S, Chan S, Chen C. Distributed access control with privacy support in wireless sensor networks. IEEE Trans Wireless Commun 2011;10(10):3472–81.
[34] Yu S, Ren K, Lou W. FDAC: Toward fine-grained distributed data access control in wireless sensor networks. IEEE Trans Parallel Distrib Syst 2011;22(4):673–86.
[35] Ruj S, Nayak A, Stojmenovic I. Distributed fine-grained access control in wireless sensor networks. In: IEEE International Parallel Distributed Processing Symposium (IPDPS'11); 2011. p. 352–62. Anchorage, AK, USA
[36] Chatterjee S., Roy S.. Cryptanalysis and enhancement of a distributed fine-grained access control in wireless sensor networks. IEEE; New Delhi, India; 2014. International Conference on Advances in Computing, Communications and Informatics (ICACCI'14), 2074–2083.
[37] Chatterjee S, Das AK. An effective ECC-based user access control scheme with attribute-based encryption for wireless sensor networks. Security and Communication Networks 2015;8(9):1752–71.
[38] Lounis A, Hadjidj A, Bouabdallah A, Challal Y. Healing on the cloud: secure cloud architecture for medical wireless sensor networks. Future Generation Computer Systems 2016;55:266–77.
[39] He H, Zhang J, Gu J, Hu Y, Xu F. A fine-grained and lightweight data access control scheme for WSN-integrated cloud computing. Cluster Comput 2017;20(2):1457–72.
[40] Liu JK, Au MH, Huang X, Lu R, Li J. Fine-grained two-factor access control for web-based cloud computing services. IEEE Trans Inf Forensics Secur 2016;11(3):484–97.
[41] Li J, Chen X, Chow SS, Huang Q, Wong DS, Liu Z. Multi-authority fine-grained access control with accountability and its application in cloud. Journal of Network and Computer Applications 2018;112:89–96.
[42] Belguith S, Kaaniche N, Laurent M, Jemai A, Attia R. PHOABE: Securely outsourcing multi-authority attribute based encryption with policy hidden for cloud assisted iot. Comput Networks 2018;133:141–56.
[43] Banerjee S, Odelu V, Das AK, Chattopadhyay S, Giri D. Anonymous fine-grained user access control scheme for internet of things architecture. In: 5th International Conference on Mathematics and Computing (ICMC'19); 2019. p. 1–18. Bhubaneswar, India
[44] Dolev D, Yao A. On the security of public key protocols. IEEE Trans Inf Theory 1983;29(2):198–208.
[45] Messerges TS, Dabbish EA, Sloan RH. Examining smart-card security under the threat of power analysis attacks. IEEE Trans Comput 2002;51(5):541–52.
[46] Kocher P, Jaffe J, Jun B. Differential power analysis. In: Proceedings of Advances in Cryptology - CRYPTO'99, LNCS, vol. 1666; 1999. p. 388–97. Santa Barbara, CA, USA
[47] Canetti R., Krawczyk H.. Analysis of key-exchange protocols and their use for building secure channels. Springer; 2001. International Conference on the Theory and Applications of Cryptographic Techniques, 453–474.

[48] Bertino E, Shang N, W JSS. An efficient time-bound hierarchical key management scheme for secure broadcasting. IEEE Trans Dependable Secure Comput 2008;5(2):65–70.
[49] Wazid M, Das AK, Odelu V, Kumar N, Susilo W. Secure remote user authenticated key establishment protocol for smart home environment. IEEE Trans Dependable Secure Comput 2020;17(2):391–406.
[50] Dodis Y, Yampolskiy A. A verifiable random function with short proofs and keys. In: 8th International Workshop on Theory and Practice in Public Key Cryptography (PKC'05); 2005. p. 416–31. Les Diablerets, Switzerland
[51] Advanced encryption standard (AES). FIPS PUB 197, National Institute of Standards and Technology (NIST), U.S. Department of Commerce, November 2001. Available at https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.197.pdf. Accessed on January 2020.
[52] Herranz J, Laguillaumie F, Ràfols C. Constant size ciphertexts in threshold attribute-based encryption. In: International Workshop on Public Key Cryptography (PKC'10); 2010. p. 19–34. Paris, France
[53] Diffie W, Van Oorschot PC, Wiener MJ. Authentication and authenticated key exchanges. Designs, Codes and Cryptography 1992;2(2):107–25.
[54] Ballare M, Rogaway P. Random oracles are practical: A paradigm for designing efficient protocols. In: Proceedings of the 1st ACM Conference on Computer and Communications Security (CCS'93); 1993. p. 62–73. Fairfox, VA, USA
[55] Shoup V.. Sequences of games: A tool for taming complexity in security proofs. Cryptology ePrint archieve, Report 2004/332. Available at http://eprint.iacr.org/2004/332, 2004.
[56] Roy S, Chatterjee S, Das AK, Chattopadhyay S, Kumari S, Jo M. Chaotic map-based anonymous user authentication scheme with user biometrics and fuzzy extractor for crowdsourcing internet of things. IEEE Internet Things J 2018;5(4):2884–95.
[57] Wang D, Cheng H, Wang P, Huang X, Jian G. Zipf'S law in passwords. IEEE Trans Inf Forensics Secur 2017;12(11):2776–91.
[58] Pointcheval D, Zimmer S. Multi-factor authenticated key exchange. In: International Conference on Applied Cryptography and Network Security (ACNS'08); 2008. p. 277–95. New York, NY, USA
[59] Chatterjee S, Roy S, Das AK, Chattopadhyay S, Kumar N, Vasilakos AV. Secure biometric-based authentication scheme using chebyshev chaotic map for multi-server environment. IEEE Trans Dependable Secure Comput 2018;15(5):824–39.
[60] von Oheimb D. The high-level protocol specification language hlpsl developed in the eu project avispa. In: Proceedings of 3rd APPSEM II (Applied Semantics II) Workshop (APPSEM'05); 2005. p. 1–17. Frauenchiemsee, Germany
[61] AVISPA. SPAN, the security protocol ANimator for AVISPA. 2020b. http://www.avispa-project.org/. Accessed on March 2020.
[62] Chang CC, Le HD. A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks. IEEE Trans Wireless Commun 2016;15(1):357–66.
[63] Abadi M, Blanchet B, Comon-Lundh H. Models and proofs of protocol security: A progress report. In: Proceedings of 21st International Conference on Computer Aided Verification (CAV'09); 2009. p. 35–49. Grenoble, France
[64] Wu L, Wang J, Choo KR, He D. Secure key agreement and key protection for mobile device user authentication. IEEE Trans Inf Forensics Secur 2019;14(2):319–30.
[65] Perkins CE, Royer EM. Ad-hoc on-demand distance vector routing. In: Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99); 1999. p. 90–100. New Orleans, LA, USA
[66] Henderson TR, Lacage M, Riley GF, Dowell C, Kopena J. Network simulations with the ns-3 simulator. SIGCOMM demonstration 2008;14(14):527.

Soumya Banerjee is pursuing his Ph.D. in Computer Science and Engineering from Jadavpur University, Kolkata, India. He received an M.Tech degree in Software Engineering from Jadavpur University, Kolkata, India. His current research interests include cryptography and network security. He has authored several papers in international journals and conferences in the above areas.

Sandip Roy received his Ph.D. degree in Computer Science and Engineering from Jadavpur University, Kolkata, India. He is currently working as an Assistant Professor in the Department of Computer Science and Engineering of Asansol Engineering College, India. He received an M.Tech degree in Computer Science and Technology from West Bengal University of Technology, India. His current research interests include cryptography and network security. He has published several international journal and conference papers in his research areas.

Vanga Odelu received his Ph.D. degree and M.Tech. degree in Computer Science and Data Processing from IIT Kharagpur, India. He is currently working as an Assistant Professor in the Department of Computer Science and Engineering at Indian Institute of Information Technology Sri City, Chittoor, Andhra Pradesh, India. Prior to this, he is associated with BITS Pilani, Hyderabad as an assistant professor and Korea University, South Korea as a Research Professor. He has been awarded Outstanding Potential for Excellence in Research and Academics (OPERA) by BITS Pilani. He was selected as an Outstanding Young Foreign Scholar "Korean Research Fellowship (KRF-2017)" by the Korean Government. He is an Organizing Chair of International Conference on Mining Intelligence and Knowledge Exploration (MIKE-2019). Track Chair for the Intelligent Security Systems (MIKE-2017 & 2018), and also Member of Advisory Committee MIKE-2018. His current research interests include cryptography, network security, Internet of things, and blockchain technology. He has authored over 50 papers in international journals and conferences.

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. His current research interests include cryptography, network security, blockchain, security in Internet of Things (IoT), Internet of Vehicles (IoV), Internet of Drones (IoD), smart grids, smart city, cloud/fog computing and industrial wireless sensor networks, and intrusion detection. He has authored over 215 papers in international journals and conferences in the above areas, including over 185 reputed journal papers. Some of his research findings are published in top cited journals, such as the IEEE Transactions on Information Forensics and Security, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Smart Grid, IEEE Internet of Things Journal, IEEE Transactions on Industrial Informatics, IEEE Transactions on Vehicular Technology, IEEE Transactions on Consumer Electronics, IEEE Journal of Biomedical and Health Informatics (formerly IEEE Transactions on Information Technology in Biomedicine), IEEE Consumer Electronics Magazine, IEEE Access, IEEE Communications Magazine, Future Generation Computer Systems, Computers & Electrical Engineering, Computer Methods and Programs in Biomedicine, Computer Standards & Interfaces, Computer Networks, Expert Systems with Applications, and Journal of Network and Computer Applications. He was a recipient of the Institute Silver Medal from IIT Kharagpur. He is on the editorial board of KSII Transactions on Internet and Information Systems, International Journal of Internet Technology and Secured Transactions (Inderscience), and IET Communications, is a Guest Editor for Computers & Electrical Engineering (Elsevier) for the special issue on Big data and IoT in e-healthcare and for ICT Express (Elsevier) for the special issue on Blockchain Technologies and Applications for 5G Enabled IoT, and has served as a Program Committee Member in many international conferences. He also severed as one of the Technical Program Committee Chairs of the International Congress on Blockchain and Applications (BLOCKCHAIN'19), Avila, Spain, June 2019.

Samiran Chattopadhyay is currently working as Professor in the Department of Information Technology, Jadavpur University, Kolkata, India. He has received his Ph.D. from Jadavpur University, Kolkata, India, and Masters and Bachelors in computer science and engineering from IIT Kharagpur, India. He is having over 25 years of teaching experience at Jadavpur University, 4 years of industry experience, and 12 years of technical consultancy in the reputed industry houses. He has authored over 110 papers in international journals and conferences.

Joel J. P. C. Rodrigues is a professor at the Federal University of Piauí, Brazil; senior researcher at the Instituto de Telecomunicações, Portugal; and collaborator of the Post-Graduation Program on Teleinformatics Engineering at the Federal University of Ceará (UFC), Brazil. Prof. Rodrigues is the leader of the Next Generation Networks and Applications research group (CNPq), IEEE Distinguished Lecturer, Past-Director for Conference Development - IEEE ComSoc Board of Governors, 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, 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 co-authored over 850 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 Fellow of IEEE.

Youngho Park received his BS, MS,and Ph.D. degrees in electronic engineering, Kyungpook National University, Daegu, Korea in 1989, 1991, and 1995, respectively. He is currently a professor at School of Electronics Engineering, Kyungpook National University. In 1996–2008, he was a professor at School of Electronics and Electrical Engineering, Sangju National University, Korea. In 2003–2004, he was a visiting scholar at School of Electrical Engineering and Computer Science, Oregon State University, USA. His research interests include computer networks, multimedia, and information security.