Context-Aware Adaptive Remote Access for IoT Applications
Amel Arfaoui, Soumaya Cherkaoui, Ali Kribeche, Sidi Mohammed Senouci

To cite this version:

HAL Id: hal-02556455
https://hal.science/hal-02556455
Submitted on 11 Feb 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License
Context-Aware Adaptive Remote Access for IoT Applications

Amel Arfaoui, Soumaya Cherkaoui, Ali Kribeche, and Sidi Mohammed Senouci

Abstract—The rapid growth of communication networking, ubiquitous sensing, and signal processing has spurred the emergence of the Internet of Things (IoT) era. As a novel cutting-edge technology, the IoT enables a plethora of smart-devices equipped with diverse computing, sensing, and actuation capabilities to be connected to the Internet. Thus, it promises to provide a revolutionary and fully connected “smart” world while greatly developing economies and enhancing the quality of life. IoT is indeed an emergent global phenomenon, where real-time remote access to data and applications opens new unprecedented opportunities for ubiquitous monitoring and managing. In such dynamic, interconnected, and heterogeneous environment where the context conditions (location, time, situation sensitivity, etc.) are continuously and frequently changing, context-aware and adaptive solutions for data access are required to respond to the applications’ needs. Nevertheless, until now, no schemes provide concrete context-aware access control mechanisms in IoT. In this article, we design a novel context-aware attribute-based access control (CAABAC) that considers the dynamic context changes. The proposed approach incorporates the contextual information with the ciphertext-policy attribute-based encryption (CP-ABE) to guarantee adaptive contextual access to data. The extensive analysis and simulations prove both the effectiveness and efficiency of the proposed scheme. Specifically, context-aware and adaptive remote access is enabled while outperforming other benchmarked schemes in terms of storage, communication, and computational cost.

Index Terms—Adaptive authorization, attribute-based encryption, context-aware remote access, Internet of Things (IoT).

I. INTRODUCTION

The Internet of Things (IoT) has been introduced as a ubiquitous and pervasive paradigm that connects transparently and seamlessly a multitude of digital devices to the Internet [1]. Recently, IoT has gained a potential momentum as the universal technology that will achieve a sustainable development [30] while creating a smart and comfortable future. Particularly, integrating smart devices as well as information and communication technologies into traditional systems has enabled significant solutions for a wide range of applications, such as smart home, industrial automation, healthcare, energy management, and smart grid [2], [31]. Hence, IoT promises a transformative value for manifold IoT applications by enabling real-time remote control and monitoring of smart things. For instance, this arises in industrial applications to manage assets and in smart homes to monitor home appliances, such as the control of home alarm systems, the monitoring of the temperature and humidity levels, turning on/off lights, etc.

The rapid development of IoT was accompanied with the generation of a large amount of data, which leads to a new big data era [32]. However, the produced data are exposed to many security vulnerabilities and threats, such as eavesdropping, data modification, man-in-the-middle-attack, DoS attack, etc. This challenge is amplified by the dynamic and heterogeneous structure of IoT where applying static and traditional security solutions is inefficient. In this context, adaptive security mechanisms become mandatory to meet application requirements in terms of security and privacy preservation. For example, authentication and authorization policies should be adaptively selected according to the context changes. The context invokes information characterizing the situation of an entity or a set of entities to define their current status [33]. It provides relevant and meaningful information, also known as “contextual information,” which is easily interpretable and understandable [34]. The contextual information can be defined by many parameters characterizing the user, its activities, its physical environment, the location (physical or virtual) from which the access is performed, time, situations of social networks, and network states. The user can be characterized by several attributes, such as its identity, role, or adversarial state. For example, the intention and motivation levels to get access may define the adversarial state of the user [29] who may be in a normal state, in a panic state, or impaired state. The location may be defined using different parameters where GPS coordinates can be used for physical location and IP address for virtual localization. In addition, data type is an expressive contextual parameter that determines the sensitivity of the requested data. Moreover, adaptive access in emergency or normal situation should be considered in order to make the right decision at the right time by the right party. Many other contextual parameters can be integrated according to the application scenario.

To emphasize the importance of the contextual information, some research works [3], [4] proposed fine-grained access...
control schemes using the ciphertext-policy attribute-based encryption (CP-ABE) embedded with time/location factors. In CP-ABE, each user is associated with a set of attributes and the data is encrypted according to an access structure. Only users whose attributes satisfy the access policy can decrypt the ciphertext. The aforementioned mechanisms are limited to the use of time or location as contextual information. In addition, the existing authorization solutions adopt only a break-the-glass strategy to deal with emergency situations. In this approach, a master key is given to authorized data consumers to decrypt the ciphertext [5], [6]. In such critical condition, location and time are relevant parameters where access is provided to data consumers who are within a predefined location area from data owners at the time of an emergency. Consequently, it seems crucial to conceive an effective access control scheme that grants adaptive data access to authorized users for a multivalued context. A trivial solution to handle the above challenge is to integrate a user’s role and the contextual parameters into access policies as a set of normal attributes [7]. However, the main difference between a dynamic context and a set of attribute is that attributes are assigned to users according to their identities which will be statically maintained for a long time while the contextual information involves a set of dynamic conditions, which are frequently changing over time. For instance, in a contextual parameter such as location/time is considered as a normal attribute of the user, the attribute set will change permanently anywhere at any time. For instance, in [7], whenever the context changes, the key generator center (KGC) issues a new contextual decryption key to guarantee data confidentiality. This solution is obviously burdensome in real scenarios and introduces heavy computation and communication cost [3].

To tackle all the aforementioned challenges, we introduce a novel context-aware authentication and authorization scheme to adaptively provide access privileges to authorized users while considering the dynamic context changes. The main purpose of the proposed approach is to dynamically and efficiently assign access rights under a given situation. Specifically, we aim to ensure context-aware and secure remote control of smart things. Thus, we design a context-aware attribute-based access control (CAABAC) scheme that integrates the contextual information with attributes into the access policy. Specifically, we exploit the worthiness of the CP-ABE scheme to grant fine-grained authorization. In addition, we use a contextual token concept to define the different contextual parameters characterizing a given context. Particularly, the contextual information is involved in the access policy as contextual tokens whose corresponding secret can be revealed only under a predefined context by a context manager (CM). The CM is deployed to verify the validity of data consumer’s requests and generate access tokens for them. To decrypt a ciphertext, the user needs not only to possess the proper attribute set but also to have a valid access token. The contextual tokens can be set arbitrarily in the access policy along with attributes, and one ciphertext can be linked to many contextual tokens as the number of the considered contextual parameters (location, time, emergency or normal situation, data type, etc.).

Thus, the contextual information can be adaptively embedded with the users’ attributes. The major contributions of this article can be summarized as follows.

1) A novel context-aware access control approach for adaptive and secure control of smart things based on the contextual information is proposed. Specifically, we handle the fine-granularity of the CP-ABE scheme and embed the contextual information into access structures using contextual tokens. This alleviates the burdensome revocation of secret keys when a user’s context changes.

2) The key escrow problem of CP-ABE is addressed. Specifically, both KGC and attribute authority (AA) cooperate to generate users’ private keys, so that no authority can reveal the users’ secret keys. Therefore, data confidentiality and privacy are ensured.

3) A rigorous security proof and an informal security analysis are carried out on the proposed scheme to demonstrate its capacity and its security strength against well-known attacks.

4) A thorough comparative analysis of the performance in terms of security and functionality properties, storage, communication, and computation overhead is performed. The comparative study proves the effectiveness and efficiency of the proposed scheme compared to the benchmark schemes.

5) A design of an extended CAABAC scheme, where multiple contextual parameters can be considered and appended to arbitrary nodes, is proposed. To the best of our knowledge, we are the first to design an effective scheme that supports general context-aware access control.

The rest of the article proceeds as follows. In Section II, we review some existing works related to remote access control in IoT. We briefly discuss the relevant mathematical preliminaries in Section III. The system model is presented in Section IV. A novel CAABAC scheme for access control is presented in Section V, followed by the performance analysis in Section VI. Section VII provides an effective method to define access policies for general context. Finally, Section VIII concludes this article.

II. RELATED WORK

Security and privacy issues are becoming the main barriers facing the global deployment of IoT. One of the most critical concerns in IoT lies in the design of secure and privacy-preserving mechanisms. However, the open, dynamic, and heterogeneous structure of IoT, as well as the resource-constrained devices, impose more challenges to develop effective and efficient security solutions. In this perspective, many research works have addressed particular attention to different security facets, such as authentication and access control. Wang et al. [8] proposed a remote access control scheme to secure the communication between IoT devices and controllers (smartphone or tablet). A trust center is used to assign the control process to legitimate and authenticated controller nodes. It has been demonstrated that their scheme
is secure against various attacks, such as DoS, desynchronization, and replay attacks. However, the high number of exchanged messages introduce heavy communication costs. Kouicem et al. [9] addressed the problem of remote secure control of smart actuators. For this purpose, they proposed a distributed lightweight fine-grained access control based on the attribute-based encryption scheme and used a one-way hash chain technique for authentication. But, they performed only a role-based access control which makes their scheme vulnerable to several attacks. Wazid et al. [10], [11] proposed lightweight and secure remote user authentication schemes in IoT. They proved that their schemes are efficient and meet the security and privacy requirements. Nevertheless, they did not consider fine-grained access control which is a paramount concern for remote control applications. In [12], a context-aware authentication framework is designed for smart home applications. The proposed model provides real-time and continuous authentication between the users and smart things while considering the contextual information (such as the user’s location, profile, calendar, request time, etc.). Yet, the model does not provide a structured formulation to integrate the contextual information in the authentication process.

Recently, capability-based access control (CapBAC) was introduced as a suitable authorization tool to be implemented in IoT [13], [14]. In this approach, a central entity is deployed to define access privileges and deliver authorization tokens to users. However, the use of a central entity that verifies and validates users’ access rights leads to a single point of failure and impedes end-to-end security. To tackle these issues, distributed CapBAC was proposed [15], [16] to delegate the authorization decisions to the IoT devices themselves. However, IoT objects are often resource-impoverished and may be easily compromised. Therefore, distributed CapBAC is ill-equipped to address access control in untrustworthy IoT environment.

Attribute-based cryptography (ABC) is considered as a promising technique to ensure fine-grained access control. In such a scheme, data is encrypted according to an access policy, and only the user whose attribute set fits the access structure can decrypt the ciphertext. In this context, several studies looked into the feasibility of applying CP-ABE to resource-constrained devices. For instance, Hu et al. [17] proposed to implement the CP-ABE scheme in sensor nodes to secure the communications between them and the data sink/data consumers. Specifically, they focused on a role-based access control. However, the proposed scheme suffers from the key escrow problem and high computational cost. Hence, it is impractical for resource-constrained devices that cannot support the heavy overhead of CP-ABE. To deal with this, some contributions [18], [19] proposed to implement a multiauthority CP-ABE scheme where heavy cryptographic operations are outsourced to more powerful nodes in the network. Jahan et al. [20] developed a robust and fine-grained access control scheme with user revocation in IoT. In their scheme, some operations of CP-ABE are delegated to IoT devices. Moreover, they used multiple authorities for decryption keys generation in order to solve the key escrow problem.

To the best of our knowledge, none of the above works (presented in Table I) addresses the design of effective context-aware authentication and authorization scheme. Besides, even if the contextual information is involved in some approaches, it is considered as a set of normal attributes or no concrete construction is provided. Furthermore, in such approaches, only time or location is used to define the context. Thus, the following crucial challenges arise to conceive secure remote control of smart things: 1) How to properly and efficiently define access policies of different users while considering the dynamic context changes? 2) How to guarantee both context-aware and fine-grained access control in IoT? 3) How to resolve the key escrow problem? 4) How to minimize computational, storage, and communication overhead and provide a lightweight security solution for resource-constrained devices? The context-aware and fine-grained access control approach proposed in this article and detailed in the next sections carries out all the aforementioned challenges.

### III. Mathematical Preliminaries

In this section, we briefly discuss the bilinear pairings characteristics and the cryptographic primitives required to design and analyze the proposed scheme.

#### A. Bilinear Pairings

Let \( G_1 \) be a cyclic additive group of prime order \( q \) and \( G_2 \) be a cyclic multiplicative group of the same order \( q \). A bilinear pairing is a map \( e : G_1 \times G_1 \rightarrow G_2 \) and satisfies the following properties.

1) **Bilinear**: A map \( e : G_1 \times G_1 \rightarrow G_2 \) is bilinear if and only if \( \forall P, Q \in G_1 \) and \( a, b \in \mathbb{Z}_q \), we have \( e(aP, bQ) = e(P, Q)^{ab} \).

#### Table I

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Data access</th>
<th>Confidentiality</th>
<th>Integrity</th>
<th>Authentication</th>
<th>Context-awareness</th>
<th>Privacy</th>
<th>Adaptation</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>[8]</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>High</td>
</tr>
<tr>
<td>[9]</td>
<td>RBAC</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>Medium</td>
</tr>
<tr>
<td>[10]</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>Low</td>
</tr>
<tr>
<td>[11]</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>Low</td>
</tr>
<tr>
<td>[13]</td>
<td>capBAC</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>Medium</td>
</tr>
<tr>
<td>[17]</td>
<td>RBAC</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>High</td>
</tr>
<tr>
<td>[20]</td>
<td>RBAC</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>High</td>
</tr>
</tbody>
</table>
2) Nondegeneracy: \( \exists P, Q \in \mathbb{G}_1 \) where \( e(P, Q) \neq 1_{\mathbb{G}_2} \).
3) Computability: \( \forall P, Q \in \mathbb{G}_1 \), there is an efficient algorithm to compute \( e(P, Q) \) in polynomial time.

The security of the proposed scheme depends on the following intractable problem.

1) Decision Bilinear Diffie–Hellman (DBDH) Problem: Given two groups \( \mathbb{G}_1 \) and \( \mathbb{G}_2 \) with the same prime order \( q \), a bilinear map \( e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2 \) and a generator \( g \) of \( \mathbb{G}_1 \), the objective of DBDH is to compute \( e(g, g)^{abc} \) in \( (\mathbb{G}_1, \mathbb{G}_2, e) \) from the given \( (g^a, g^b, g^c) \), \( \forall a, b, c \in \mathbb{Z}_q \).

B. Ciphertext-Policy Attribute-Based Encryption

CP-ABE [21] is a cryptography prototype for one-to-many secure communication. The system model of a CP-ABE-based approach consists of the following entities: KGC, the data owner, and the user. The KGC is used to publish system parameters and generate secret keys for data consumers. The data owner is responsible for access policy definition and data encryption under the predefined policy. In the CP-ABE scheme, the access policy is expressed as a tree over an attribute set and logic gates. Each user gets his/her secret key from the KGC based on his/her attribute set. CP-ABE scheme consists of the following four algorithms.

1) Setup(\( \lambda \)): Given a security parameter \( \lambda \), the KGC generates a master secret key \( MK \) that is kept private and a public key \( PK \) shared by users.
2) KeyGen (\( PK, MK, S \)): The KGC takes the master secret key \( MK \), the attribute set of the user \( S \), and the public key of the system \( PK \) as inputs. It generates the private key \( SK_U \).
3) Encryption (\( PK, M, T \)): It takes the public parameters \( PK \), a plaintext \( M \), and an access structure \( T \) as inputs. The algorithm will encrypt \( M \) and generate a ciphertext \( CT \).
4) Decryption (\( CT, SK_U \)): The receiver takes as input the ciphertext \( CT \) and his/her decryption key \( SK_U \). The algorithm outputs a message \( M \) or a reject symbol \( \bot \).

C. Access Policy Structures

An access structure \( T \) consists of several nodes of a policy tree, and several contextual tokens as illustrated in Fig. 1. A leaf node represents a set of attributes (\( att_0, \ldots, att_i \)), and each nonleaf node defines a threshold gate (“AND,” “OR,” or other threshold gates). Each nonleaf node \( x \) takes two logic value \( nx \) and \( kx \), where \( nx \) is the number of its child node, and \( kx \) is the threshold. Specifically, \( kx = 1 \) if \( x \) is an OR gate, or \( kx = nx \) if \( x \) is an AND gate [21]. A contextual token \( T^C \) can be integrated in an access structure to restrict user’s access permission by the contextual parameter \( c_i \) that may be time, location, situation sensitivity, etc. In the proposed CAABAC, contextual tokens are presented as leaf nodes, which can be considered as a set of special attributes. They are generated by a data owner when encrypting his/her data, and an access token is delivered to authorized data consumers to decrypt each contextual token.

For example, in Fig. 1, the contextual token \( T^C_2 \) is related to the contextual parameter \( C \) and users who satisfy “\( att_1 \wedge att_2 \)” cannot access data until the contextual information is verified.

IV. SYSTEM AND SECURITY MODELS

In this section, we present the network and security models to be used in the design of the CAABAC scheme for generic IoT network.

A. System Model

We consider an IoT remote control system (presented in Fig. 2) that consists of the following parties: KGC, AA, CM, IoT gateway, smart things, and data consumers.

The different functions of each entity are presented as follows.

1) The KGC and the AA are semi-trusted entities. They are responsible for system initialization where public parameters and users’ secret keys are generated. They cooperate in delivering users’ attributes secret keys to solve the key escrow problem.
2) CM is responsible for the control of the dynamic context changes. It verifies the validity of a user’s request to access data under a given context and generates access tokens to enable authorized data consumers decrypting the ciphertext.
3) IoT gateway is deployed as a powerful node that cooperates with the IoT device to define the appropriate access rules and encrypt data based on the CP-ABE scheme. In addition, it is used as a relay node between the smart thing and the data consumer. It also conducts the management of remote access control to smart things.
4) Smart things are resource-constrained devices that constitute the control system network. These devices are deployed in an area of interest and remotely controlled by data consumers.
5) Data consumers are the users who aim to communicate with IoT devices and execute remote actions on them. To decrypt a message, data consumers have not only to possess the set of attributes that satisfy the access structure but also to meet the contextual requirements related to time, location, situation sensitivity (normal/emergency), and data type (sensitive, nonsensitive).
To ensure the security of a CP-ABE scheme, the system should be indistinguishable against chosen plaintext attacks (IND-CPA). The details of the security model of CP-ABE can be found in [22].

The proposed CAABAC can guarantee context-aware fine-grained access control only when a user who satisfies the attribute set can access the data under a specific context. The proposed scheme is considered as compromised if data consumers satisfying the following conditions can decrypt the ciphertext: 1) A user whose attribute set does not match the access policy and 2) A user who does not meet the context requirements can gain access privileges, even if he/she has the proper attribute set.

In CAABAC, we assume that AA and KGC are semi-trusted entities. Namely, they are honest-but-curious; meaning they can honestly execute the assigned tasks, but they will try to disclose as much sensitive content as possible. The CM is considered fully trusted, and CM is managed by local administrators who aim to secure the communication between the communicated parties. The owner data is encrypted by both the IoT gateway and smart things. On the one hand, the IoT gateway is assumed to be fully trusted, and it cooperates with IoT devices to encrypt data and define access policies according to the attribute set and the contextual information. On the other hand, smart things are resource-constrained devices that may be easily compromised. In this article, we assume that smart things are available and not compromised. In fact, they can effectively perform symmetric encryption of the data. Data consumers are dishonest. We assume they try to decrypt data beyond the data for which they have access rights, so a malicious or unauthorized user may collude with other users to access data beyond his/her privileges.

V. CONTEXT-AWARE ATTRIBUTE-BASED ACCESS CONTROL SCHEME

In this section, we first list the basic notations used to detail the proposed scheme. Then, we present the main features and a detailed description of the proposed CAABAC scheme that ensures an adaptive access control according to the dynamic context changes.

A. Notation

See Table II.

B. CAABAC Scheme Description

To ensure context-aware remote access control in IoT, we introduce the contextual information (location, time, data type,
emergency or normal situation, etc.) into the basic CP-ABE. One of the biggest issues is how to incorporate dynamic contextual parameters with CP-ABE. As mentioned previously, there is a lack of effective approaches considering a general and dynamic context, except considering time/location information or handling the context as a set of attributes.

The main idea of our novel scheme is to acquire the fine granularity of CP-ABE and avoid the burdensome overhead imposed by the user’s revocation due to frequent context changes. For this purpose, we introduce a contextual token mechanism to guarantee dynamic access control while considering the contextual information. Specifically, we integrate contextual tokens into the access structure to restrict access privileges by the contextual information. The contextual information combined with the user’s attribute set, determine whether the user satisfies the access policy. The motivation for distinguishing contextual parameters from other attributes is that they are intrinsic dynamic attributes. Thus, this concept enables efficient updates.

A contextual token is set with the contextual parameter $c_j \in [c_1, \ldots, c_n]$ is a unified format of the contextual parameter, such as “dd/mm/yyyy” for time, “GPS coordinates” for location, and “event: emergency” for situation sensitivity. It is placed upon leaf nodes in the access structure, arbitrarily defining context-aware access privilege. CAABAC implements an identity-based encryption (IBE) algorithm [23] to generate contextual tokens, in which a contextual parameter is treated as an identity. A conspicuous property of tokens is that they are unique for different ciphertexts. That means a token related to ciphertext $A$ cannot be used to decrypt a ciphertext $B$ ($B \neq A$).

Successful decryption implies not only a proper attribute set but also a valid access token. In fact, a data consumer interacts with the CM which validates the context requirements and delivers an access token. Without the suitable access token, the data consumer cannot decrypt the ciphertext properly even if he/she satisfies the attribute set. In the following, we describe the detailed discussion of various phases related to the proposed scheme: system initialization, key generation, encryption, and decryption and communication.

1) System Initialization: In this phase, both KGC and AA generate their secret keys and distribute the public parameters to all the entities in the system. In addition, the CM defines the secret keys of the contextual parameters [24].

2) Key Generation: This phase is executed by both AA and KGC to generate users’ secret keys based on their attributes. A private key has to be generated for each attribute a user possesses. To this end, AA selects a random number $r \in \mathbb{Z}_q$ for the user $u$. Then, AA and KGC perform a secure two-party computation (2PC) protocol, where AA inputs $MK_A = \{a_1, \beta\}$ and KGC inputs $MK_{KGC} = \{a_2\}$. Finally, KGC gets $X = (a_1 + a_2) \mod q$ [24]. After the 2PC protocol, AA and KGC execute the key commitment algorithm (Algorithm 2).

Fig. 3 presents a direct description for the above key issuing protocol. Here, the first step denotes a 2PC protocol which inputs $MK_A = \{a_1, \beta\}$ from AA and $MK_{KGC} = \{a_2\}$ from KGC, and returns $X = (a_1 + a_2) \mod q$ to KGC. Based on the enhanced key issuing protocol, AA and KGC generate the user’s private key.

3) Encryption: In this phase, a smart thing assigns a random challenge $M$ for each instruction $I$ that it can execute. Then, it cooperates with the IoT gateway to encrypt the challenge $M$. For this purpose, the IoT device performs first symmetric key encryption to encrypt $M$ with $K_I$ where $K_I$ is a pre-shared secret key with the IoT gateway. After that, the IoT gateway encrypts $K_I$ based on an access tree $T$ for a given contextual information that includes location, time,
Algorithm 3 Encryption

Input: An access tree $\mathcal{T}$, public parameters $PK$, contextual parameter $c_j$, symmetric key $K_s$

Output: The ciphertext $CT$, $\sigma$

1. For each node $x$ in the tree $\mathcal{T}$, choose a polynomial $q_x$ whose degree is $d_x = k_x - 1$. End for
2. Pick a random $s \in \mathbb{Z}_q$ and set $q_R(0) = s$
3. Select $d_g$ random points from $\mathbb{Z}_q$ to completely define the polynomial $q_R$
4. For any other node $x$ in $\mathcal{T}$ do
   5. Set $q_1(0) = q_{\text{parent}(x)}(\text{index}(x))$
   6. Select $d_x$ random points from $\mathbb{Z}_q$ to completely define $q_x$
   7. End for
8. If $x$ is a leaf node related to the contextual parameter $c_j$ then
   9. Choose a random number $r_x \in \mathbb{Z}_q$
   10. Generate a contextual token $T_x^{\sigma} = (A_x^{\sigma} = g^{s_i}, B_x^{\sigma} = q_x(0) + H_2(\{H_1(x_i), y_0\}^{(s_i)}))$
   11. End if
12. The ciphertext $CT$ is constructed based on the access tree $\mathcal{T}$ as follows:

\[
CT = \begin{cases} 
\mathcal{T}, \tilde{C} = K_e(g, g)^{s_i}, C = g^s & \text{if } x \text{ is a root node} \\
\forall s, i \in [1, n], j \in [1, N], C_j = h^{\text{index}(x)}(0), C_j' = H_1(\text{att}_i) \cdot s_j & \text{if } x \text{ is an attribute leaf node} \\
T_x^{\sigma} = (A_x^{\sigma}, B_x^{\sigma}) & \text{if } x \text{ is a leaf node related to } c_j 
\end{cases}
\]
13. Compute $\sigma = H_1(K_s)^{\text{index}(x)}$ where $y_{\text{GW}}$ is the signature key of the gateway

Data type, and situation sensitivity (emergency/gateway). The encryption procedure executed by the IoT gateway is detailed in Algorithm 3.

4) Decryption and Communication: When a data consumer needs to perform instructions or commands in a smart thing, direct communication between them is required. Considering the resource limitations of IoT devices, the user sends a request message to the IoT gateway that generates a random nonce $r \in \{0, 1\}^*$ and sends a reply message $<CT, \sigma, r, AES(K_s, M) >$. The data consumer decrypts the ciphertext based on his/her attribute set and the contextual information according to Algorithm 4. The user that has the privilege to decrypt $K_s$ and cover the plaintext $M'$ can prove to the IoT gateway that he/she verifies the attribute set and the context requirements. Then, he/she sends back to the gateway $M_2 = <H_2', SK >_{PK_{GW}}$ where $H_2' = H_1(M' || r)$ and $SK$ is a symmetric key, which will be used to communicate the authentication token between him/her and the smart thing. Upon receiving the proof $M_2$, the IoT gateway decrypts it with its private key and verifies if $H_2' = H_1$. If succeeds, the IoT gateway generates an authentication token $TK$ for a period $T_e$ and sends $M_3 = <TK, T_e, ID_I >$ to both the data consumer and the IoT device, where $ID_I$ is the identity of the instruction $I$ that will be performed by the smart thing. We notice that $M_3$ is encrypted by $SK$ to be sent to the data consumer and $SK_1$ (symmetric preshared key between the gateway and the IoT device) to be sent to the IoT device. When the data consumer sends a communication request $H_3' = H_1(TK || T_e || ID_I)$ to the IoT device, this last verifies if $H_3' = H_1$. If the condition holds, a secure communication channel between the user and the smart thing is set up.

VI. CAABAC SCHEME PERFORMANCE AND SECURITY ANALYSIS

In this section, we first prove the correctness of the proposed scheme. Then, we perform a detailed security analysis using both formal and informal security analysis. Finally, we evaluate the performance of the CAABAC scheme in terms of security features, storage, communication, and computation overhead compared to benchmarking approaches.

A. Correctness of the Proposed Scheme

In this section, we illustrate that the CAABAC scheme is indeed feasible and correct. From Algorithm 4, we can demonstrate whether the received key $K_s$ has been forged as follows:

\[
K_s' = \frac{\tilde{C}_A}{e(g^s, g^{alpha}h^t)} = \frac{K_x e(g, g)^{alpha} e(g, g)^{beta}}{e(g^s, g^{alpha + beta})}
\]
Thus, if \( e(\sigma, g) = e(H_1(K_i'), g^{r_{\text{cw}}}) \), \( K_i' \) is valid. Therefore, the data consumer can decrypt the ciphertext using \( K_i' \) to cover the message \( M \).

B. Security Analysis

In this section, we assess the security effectiveness of the proposed scheme by examining its security properties and its capacity to resist against several attacks.

1) Mutual Authentication: The proposed CAABAC scheme ensures mutual authentication between the user and the IoT gateway as well as between the user and the smart thing. On the one hand, the authentication between the IoT gateway and the data consumer is performed based on a challenge-response technique. On the other hand, a smart thing authenticates a data consumer using an authentication token.

2) Context-Aware Privacy: In our construction, we involve the contextual information to define the access structure and encrypt data. Indeed, based on the embedded contextual tokens into access structures, we can determine who can access what and under which context. A data consumer can decrypt the ciphertext only if he/she meets the context requirements and he/she possesses a valid access token.

3) Collusion Attack Resistance: In the proposed scheme, the set of user’s attributes are associated with a secure \( SK_u \), that is blinded using a personalized secure random number \( r \in Z_q \). In fact, users may collude by combining their attribute sets. However, they cannot combine their secret keys \( SK_u \) to forge a new private key for the combined set of attributes. In addition, data consumers have different attribute sets because they have different roles and context, thus different access rights. Therefore, the collusion will not bring more privileges to adversaries.

4) Resilience Against Escrow Problem: In the proposed scheme, both KGC and AA cooperate to create users’ secret keys. According to Algorithm 2, KGC and AA cannot know each other’s master secret key. Thus, any authority cannot generate the whole secret keys of users individually.

5) Resilience Against Replay Attack: To protect the proposed scheme from a replay attack, the IoT gateway uses a random nonce \( r \). In fact, the user response message \( H_1 = H_1(M||r) \) cannot be used by another user to obtain an authentication token from the gateway. Furthermore, we add an expiration time \( T_e \) to the authentication token \( TK \) to guarantee the validity and freshness of the communicated messages.

6) Data Confidentiality: Data confidentiality of the proposed CAABAC can be analyzed based on the security model that is described in the next section.

C. Security Proof

In this section, we perform formal security analysis. At first, we present the security model through a game between a challenger \( B \) and an adversary \( A \). Then, we prove the data confidentiality of the proposed CAABAC scheme.

1) Security Model:

1) Setup: The challenger runs the initialization algorithm of CAABAC to generate public parameters \( PK \), the master key \( MK \) and the context-related keys \( \{\delta_{c_j}, PK_{c_j}, F_{c_j}\} \). Then it gives public parameters to the adversary \( A \).

2) Phase 1: The adversary can issue queries for a private key of a set of attributes \( S_u \), a set of contextual parameters \( c_j, j \in [1, N] \), and a challenge access policy \( \mathcal{T} \), where \( S_u \) does not satisfy \( \mathcal{T} \) for the given context. The challenger generates the private key associated with \( S_u \) and a series of access tokens that represent the context, and then gives the secret key to the adversary.

3) Challenge: The adversary submits two messages \( M_0 \) and \( M_1 \) with equal lengths. Then, \( B \) flips a random bit \( v \in \{0, 1\} \), and encrypts \( M_v \) under \( \mathcal{T} \). The ciphertext is sent to \( A \).

4) Phase 2: In this phase, \( A \) continues to execute queries as in phase 1 but with the restriction that the attribute set and the access tokens cannot satisfy \( \mathcal{T} \).

5) Guess: \( A \) outputs a guess \( v' \) of \( v \). The advantage of the adversary in this game is defined as

\[
\text{Adv}_A = \left| \Pr(v' = v) - \frac{1}{2} \right|
\]

Definition 1: The proposed CAABAC scheme is selectively secure against chosen-plaintext attacks (CPAs-secure) if all probabilistic polynomial time (PPT) adversaries have at most a negligible advantage in the above security game.

We further analyze classifies adversaries into two categories: 1) an adversary without satisfied attribute set for \( \mathcal{T} \) (Type-I adversary) and 2) an adversary without satisfied context requirements for \( \mathcal{T} \) (Type-II adversary).

We use the generic bilinear group model and the random oracle model to demonstrate that no adversary can break the security of the proposed scheme with any reasonable probability. We note that the security proof technique follows that of [21].

2) Security Proof Against Type-I Adversary:

Theorem 1: For any adversary \( A \), let \( p \) be a bound on the total number of group elements that \( A \) receives from the performed queries to the oracles for the hash function, groups \( G_1, G_2 \), the bilinear map \( e \), and from its interaction with the security game, in which \( G_1 \) is bilinear group of prime order \( q \).

We have that the advantage of the adversary in the security game is \( O(p^2\lambda^2) \).

Proof: In the basic CP-ABE security game, the challenger constructs the ciphertext component \( \hat{C} \) that is either \( M_{\theta}e(g, g)^{\alpha} \) or \( M_1e(g, g)^{\alpha} \). We consider as [21], a modified game where \( \hat{C} \) may be \( e(g, g)^{\alpha} \) or \( e(g, g)^{\theta} \) and \( \theta \in Z_q \) and the adversary has to decide which is the case. The modified game can be considered as hybrid argument in which the adversary has to distinguish between \( e(g, g)^{\theta} \) and \( M_0e(g, g)^{\alpha} \) and in another case between \( e(g, g)^{\theta} \) and \( M_1e(g, g)^{\alpha} \). Consequently,
any adversary that has advantage $\varepsilon$ in the CP-ABE game can be transformed into an adversary that has an advantage at least $\varepsilon/2$ in the modified game.

1) **System Initialization:** The challenger chooses $\alpha, \beta, \alpha_2 \in \mathbb{Z}_q$ and sets $h = g^{\beta}, u_1 = \varepsilon(g, g)^{\alpha_1}, v_1 = g^{\alpha_1}, u_2 = \varepsilon(g, g)^{\alpha_2}, v_2 = g^{\alpha_2}$, and $\alpha = \alpha_1 + \alpha_2$. For any query on an attribute $att(x)$, the challenger returns $g^i$ and stores $(d_i, att(x))$ into $H$-list where $d_i \in \mathbb{Z}_q$. $B$ defines the format $F_c$, for the considered contextual parameters $c_j$. In addition, the challenger chooses a random number $\delta_{c_j} \in \mathbb{Z}_q$ and sets the corresponding public keys as $PK_{c_j} = (\mathbb{F}_c, \gamma_{c_j} = g^{\delta_{c_j}})$. Finally, $B$ sends the secret parameters $PK = \{g, h, u = u_1, u_2, \bigcup F_c, \gamma_{c_j} \forall j \in [1, N]\}$ to the adversary.

2) **Phase 1:** In this phase, $B$ answers a user's $u$ secret key queries for an attribute set $S_u$ and a given context $C$. The challenger chooses first $r_a \in \mathbb{Z}_q$ and computes $D = g^{\beta} h^{r_a}, L = g^{\beta a}, D_1 = H(att(x)^s) = g^{-\beta} h^{r_a}$. Then, $B$ sends the secret key to $A$ and stores $(SK, u, S_u)$ into $SK$-list.

3) **Challenge:** $A$ submits two equal-length challenge messages $M_0, M_1 \in \mathbb{G}_2$ and an access tree $T$ to the challenger where any secret key issued to $A$ cannot satisfy $T$. $B$ chooses $s \in \mathbb{Z}_q$ and uses linear secret sharing technique to construct shares $\lambda_x$ of $s$ for all attributes $att(x)$ in $T$ where $\lambda_x$ is uniformly and independently random in $\mathbb{Z}_q$, and it is a linear combination of independent random variables. Then, the challenger chooses $\theta \in \mathbb{Z}_q$ and generates the ciphertext components

$$\tilde{C} = e(g, g)^{\theta}, \quad C = g^{\theta}, \quad C_\lambda = g^{\delta_{c_j} \cdot H(att(x))^{-s}}.$$

For each context token $T^{c_j}_\lambda$, $B$ sets $T^{c_j}_\theta = q_1(0) = s_0$.

1) **Phase 2:** The challenger proceeds as in phase 1. As this proof focuses on the adversary without adequate attribute set, the access token query will not increase the adversary’s advantage.

2) **Guess:** The adversary outputs a guess bit.

We consider an unexpected collision. An oracle query can be seen as a rational function $v = \eta/\psi$ in the variables $\theta, \alpha, \beta, r_a, \lambda_i, d_s$ and $s$. Suppose that there are distinct rational functions $v = \eta/\psi, v' = \eta'/\psi'$. An unexpected collision event indicates that for two different queries of the two functions, we have the same output due to random choice of variables. If the event happens, then $v = v' \iff \eta/\psi = \eta'/\psi = 0$. Based on Schwartz Zippel lemma [25], this event occurs with probability $O(1/q)$. Hence, the probability of a collision event is at most $O(p^2/q)$. Consequently, the collision will not arise with probability $1 - O(p^2/q)$.

The adversary $A$ can distinguish between $\theta$ and $\alpha s$, if there are two distinct queries $v$ and $v'$ giving the same output. Assume that $v = \gamma(\alpha s)$ and $v' = \gamma'\theta$, thus, $v' - v = \gamma'\theta - \gamma(\alpha s)$ such that $\gamma(\alpha s) = v - v' + \gamma'\theta$. We will prove that $A$ cannot construct a query for $\gamma(\alpha s)$ in $\mathbb{G}_2$. For this purpose, we will define all possible queries in $\mathbb{G}_2$ by means of bilinear map and the group of elements given to $A$. The adversary can obtain the transcript $\{g, g^\beta, g^\gamma, g^{\alpha s}, g^{\alpha s}, g^{\gamma s}, g^{\beta s}, g^{\beta s}, g^{\gamma s}, g^{\delta_{c_j}}, A^{c_j}, B^{c_j}\}$. Another transcript can be set as $\{g, g^\beta, g^\gamma, g^{\alpha s}, g^{\alpha s}, g^{\gamma s}, g^{\beta s}, g^{\beta s}, g^{\gamma s}, g^{\delta_{c_j}}, A^{c_j}, B^{c_j}\}$. Since $\beta$ is not relevant to construct a query involving $\alpha$, we ignore all factors of $\beta$. We find that there are two types of outputs with $\alpha s$ from two transcripts in $\mathbb{G}_2$. One is $\alpha + \beta r_a x$ (resp.$(\alpha + \beta r_a x)$) by pairing $s$ with $(\alpha + \beta r_a x)$ (resp.$(\alpha + \beta r_a x)$). To generate $\gamma(\alpha s)$, we need to eliminate $\beta r_a x$, however, no combination can satisfy $\beta r_a x$. Another combination to form $\gamma(\alpha s)$ is $-(\alpha + \beta r_a s d_i)$ (resp.$-(\alpha + \beta r_a s d_i)$). Thus, we need to eliminate $\beta r_a s d_i$. For the element $s d_i$, we need an element with $\beta r_a$ and for $r_a d_i$, we should find an element with $\beta s$. Nevertheless, none of the combinations can satisfy our requirement. From the above analysis, we can conclude that $A$ fails to construct the query $\gamma(\alpha s)$.

3) **Security Proof Against Type-II Adversary:**

**Theorem 2:** If the DBDH assumption holds, no polynomial-time adversary belongs to the second category (without satisfying context requirements) can selectively break the CABAAC scheme with nonnegligible advantage.

**Proof:** We assume that an adversary $A$ exists with a non-negligible advantage $\varepsilon$ in the selectively security game against CABAAC. The difference between this game and the last one is that the decryption cannot be executed if the contextual information does not meet the requirements. In the following, we build a challenger $B$ that can break the DBDH assumption with a non-negligible advantage.

The challenger $B$ of the DBDH game sets the groups $\mathbb{G}_1$ and $\mathbb{G}_2$, with the bilinear map $e$ and generator $g \in \mathbb{G}_1$. $B$ flips a secure random coin $b \in \{0, 1\}$. If $b = 0$, $B$ sets a tuple $(\alpha, \beta, \gamma, \delta, Z) = (g^a, g^b, g^c, e(g, g)^{\beta b})$; otherwise, the tuple is set as $(g^{a'}, g^{b'}, g^{c'}, e(g, g)^{\beta b'})$ for random $a, b, c, z$.

1) **System Initialization:** $A$ selects a challenge access structure $T$ and integrates the attribute set and the contextual information into it. The only modification in this phase compared to the last game is the generation of $PK_{c_j}$ and $H_1(\mathbb{F}_c)$. For any contextual parameter $c_j \neq c_j$, $B$ sets $PK_{c_j} = (\mathbb{F}_c, \gamma_{c_j} = g^{\gamma_{c_j}})$ where $\gamma_{c_j}$ is randomly chosen for each contextual parameter $c_j$. For the contextual parameter $c_j$, the challenger sets $PK_{c_j} = (\mathbb{F}_c, \gamma_{c_j} = B)$ and sends public parameters to $A$.

2) **Phase 1:** It is the same as the proof.

3) **Challenge:** $A$ submits two equal-length random messages $M_0, M_1 \in \mathbb{G}_2$ to $B$. Then, the challenger flips a random coin $v \in \{0, 1\}$, and encrypts $M_0$ like proof. The difference is to handle the nodes that are associated with contextual tokens. We have two cases:

a) if the contextual parameter value verifies the requirement, $B$ sets $H_1(\mathbb{F}_c) = g^{d_{c_j}}$ where $d_{c_j}$ is a random number of $\mathbb{Z}_q$. Then, the challenger picks a random number $r_{c_j}$ and calculates $A^{c_j} = g^{\gamma_{c_j}}$ and $B^{c_j} = q_1(0) + H_2(e(H_1(\mathbb{F}_c), B)^{\gamma_{c_j}})$.

b) otherwise, the random oracle sets $A = A^{c_j} = g^{\gamma_{c_j}}$ and $H_1(\mathbb{F}_c) = C g^{d_{c_j}}$. Thus, $B^{c_j}$ computes as follows:

$$B^{c_j} = q_1(0) + H_2(e(H_1(\mathbb{F}_c), B)^{\gamma_{c_j}}).$$
Therefore, the proposed scheme is secure against CPA when the set of attributes \( S_a \), there is a Lagrange interpolation for the secret \( s \)

\[
s = \sum_{A_j \in S_1} \lambda_j q_j(0) + \sum_{A_i \in S_2} \lambda_i q_i(0)
\]

where \( S_1 \) is the set of attributes related to the contextual information and \( S_2 \) is the set of other attributes.

If \( b = 0 \), we set \( Z = e(g, g)^{abc} \). The argument of \( H_2 \) (denoted as \( k \)) can be derived as

\[
k = H_2 \left( e(C, g^{d_j}, B)^{r_j} \right)
= H_2 \left( e(g^{c+d_j}, g^b)^{r_j} \right)
= H_2 \left( e(g, g)^{br_j(c+d_j)} \right)
\]

For \( a = r_c \), we can compute \( e(g, g)^{abc} \) from the argument of \( H_2 \). Thus, \( T^2 = (A^2, B^2) \) is a valid contextual token. Furthermore, the interpolation can reconstruct the secret \( s \), and the decryption will recover the plaintext.

Otherwise, if \( b = 1 \), \( Z = e(g, g)^{r} \) is a random element of \( \mathbb{G}_2 \) and the contextual token as well as the secret \( s \) will be randomly generated. Thus, the ciphertext contains no information on \( M_v \).

1) **Phase 2**: \( B \) proceeds as in phase 1. In addition, the adversary can issue access token queries for the different contextual parameters defining the context. However, even if the challenger sends \( AT^2_x \) to \( A \) for the contextual parameter \( c_i \), it is no useful for \( A \) to get the secret related to \( c_j \).

2) **Guess**: The adversary outputs a guess \( v' \) of \( v \). If \( v' = v \), the challenger will output \( b = 0 \) to indicate it was given a valid DDH tuple. Otherwise, it will output \( b' = 1 \) to indicate that it was a random 4-tuple.

In the case where \( b = 1 \), \( A \) gains no information about \( v \). Therefore, we have the probability \( \Pr(v 
= v'|b = 1) = 1/2 \).

As \( B \) guesses \( b' = 1 \) when \( v 
= v' \), we have \( \Pr(b = b'|b = 1) = \Pr(b' = 1|b = 1) = 1/2 \).

In the case where \( b = 0 \), \( A \) is given a valid contextual token with non-negligible advantage \( \epsilon \) and thus, we have \( \Pr(v = v'|b = 0) = \epsilon + (1/2) \).

As \( B \) guesses \( b = b' \) when \( v 
= v' \), we have \( \Pr(b = b'|b = 0) = \Pr(b' = 0|b = 0) = \epsilon + (1/2) \).

Hence, we can conclude that the advantage of \( B \) in the DDH game is

\[
\Pr(b = b') - \frac{1}{2} = \Pr(b = b'|b = 1) \cdot \Pr(b = 1)
+ \Pr(b = b'|b = 0) \cdot \Pr(b = 0) - \frac{1}{2}
\]

\[
= \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} - \frac{1}{2} \cdot \frac{1}{2}
= \frac{\epsilon}{2}.
\]

Therefore, the proposed scheme is secure against CPA when the adversary does not verify the context requirements.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Confidentiality</strong></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td><strong>Integrity</strong></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td><strong>Mutual authentic.</strong></td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td><strong>User-Gateway</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Mutual authentic. User-Smart thing</strong></td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td><strong>Flexible combination</strong></td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td><strong>Context-privacy</strong></td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td><strong>No. Burdensome revocation</strong></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td><strong>No. Key escrow</strong></td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
</tbody>
</table>

4) **Security Proof of the Key Issuing Protocol (Algorithm 2):**

**Theorem 3**: The key issuing protocol of Algorithm 2 is a secure protocol for computing \( D = g^b h^b \) by KGC and AA. We assume that the underlying arithmetic 2PC and zero-knowledge proofs are secure.

**Proof**: The proof of Theorem 3 is similar to that in [24]. Interested readers are referred to [24] for more details about the proof.

D. **Performance Analysis**

This section deals with the comparisons of security properties, storage, computation and communication costs among the proposed scheme and other existing schemes, such as H-CLSC [7], CP-ABE [17], and PPDAS [20]. We notice that the benchmark schemes use different cryptographic techniques to design the access control algorithm. Since CAABAC is specifically designed for resource-constrained devices, we will evaluate the communication and computation overhead at the IoT devices and IoT gateway. We assume that the bilinear \( e \) employs the Tate pairing. The elliptic curve is defined over \( F_p \). The order \( q \) of \( G_1 \) and \( G_2 \) is set to 20-byte prime. For an 80-bit security level, \( p \) should be a 64-byte prime if \( G_2 \) is a \( q \)-order subgroup of the multiplicative group of the finite field \( F_{p^2}^* \). According to [17], we can set the length of \( p \) to 42.5 bytes in the finite field \( F_{p^2}^* \). The length of an element in group \( G_1 \) is 1024 bits using an elliptic curve with 160 bits \( q \). According to the standard compression method [26], the size of an element in group \( G_1 \) can be compressed to 65 bytes.

1) **Security Properties**: In the following, we evaluate the security features of the proposed scheme as compared to the different benchmark schemes in Table III. The major feature of the CAABAC scheme lies in the dynamic and flexible integration of the contextual information into the access structure. In addition, when the context changes, only contextual tokens will change. However, even if H-CLSC [7] ensures flexibility, it introduces additional revocation to revoke users’ attributes which are related to context changes. Furthermore, compared
to existing schemes [17], [20] implementing CP-ABE, our scheme solves the key escrow problem.

2) Storage Overhead: The storage overhead is related to the size of users’ secret keys. In the CAABAC scheme, the data consumer should store \( \{SK_u = \{D = g^{\alpha h'}, L = g^r, D_i = H_1(\text{att}_i)^\gamma, \forall \text{att}_i \in S}\} \), whose size is \(|\text{att}| + 2|G_1|\), where \(|\text{att}|\) is the cardinality of the attribute set. As shown in Table IV, the data consumer in the CAABAC scheme has less storage overhead compared to existing schemes [17], [20] using the CP-ABE algorithm.

3) Communication Overhead: The communication costs of the proposed scheme and the different benchmark schemes are compared in Table V. The ciphertext is stored in the IoT gateway and transmitted to data consumers when requested. Therefore, the communication overhead is mainly associated with the size of the encrypted data. In this analysis, we consider the transmitted messages between the data consumer, the IoT gateway, and the smart thing. In fact, the gateway has to forward to the data consumer the contextual token \( T \), a decryption key and re-encrypt data for the given context changes a heavy computation cost will be induced to the IoT gateway. The computation of one symmetric encryption executed by the IoT device and seven symmetric encryption executed by the IoT device and seven pairing operations performed by the IoT gateway.

In the proposed scheme, the encryption process needs one symmetric encryption executed by the IoT device and seven Tate pairing operations performed by the IoT gateway. The computation cost of the different comparative schemes is given in Table VI. As [27], the algorithms are implemented on an Intel PXA270 processor at 624 MHz to assess the running time of operations. According to [35], the computation of one symmetric encryption takes approximately 0.919 ms on a 32-bit Cortex-M3 microcontroller running at 72 MHz. Correspondingly, the computation of one symmetric encryption on PXA270 takes about 72/624 \times 0.919 \approx 0.1 ms. The computation time of the other considered operations are given as [27]: \( T_E = 53.85 \) ms, \( T_M = 30.67 \) ms, and \( T_P = 96.20 \) ms.

As shown in Fig. 5, the proposed scheme is more efficient than existing schemes applying the CP-ABE algorithm. However, it has more computational cost compared to the H-CLSC scheme. Nevertheless, in H-CLSC [7], when the user context changes a heavy computation cost will be induced to regenerate a decryption key and re-encrypt data for the given context.

5) Cost of Contextual Tokens Generation: In the proposed scheme, a contextual token \( T_c^x \) is generated for all users and only the data consumer whose context satisfies the context.

---

### Table IV

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Storage overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP-ABE [17]</td>
<td>( {2</td>
</tr>
<tr>
<td>II-CLSC [7]</td>
<td>(</td>
</tr>
<tr>
<td>PPDA [20]</td>
<td>( {3</td>
</tr>
<tr>
<td>CAABAC</td>
<td>( {\text{att} + 2</td>
</tr>
</tbody>
</table>

### Table V

<table>
<thead>
<tr>
<th>Scheme</th>
<th>IoT gateway (bytes)</th>
<th>Smart thing (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP-ABE [17]</td>
<td>5</td>
<td>(</td>
</tr>
<tr>
<td>II-CLSC [7]</td>
<td>2</td>
<td>(</td>
</tr>
<tr>
<td>PPDA [20]</td>
<td>27</td>
<td>(</td>
</tr>
<tr>
<td>CAABAC</td>
<td>6</td>
<td>(</td>
</tr>
</tbody>
</table>

### Table VI

<table>
<thead>
<tr>
<th>Scheme</th>
<th>IoT gateway (ms)</th>
<th>Smart thing (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP-ABE [17]</td>
<td>5</td>
<td>( T_P = 481</td>
</tr>
<tr>
<td>II-CLSC [7]</td>
<td>3</td>
<td>( T_P + T_M + T_{enc} = 472.72</td>
</tr>
<tr>
<td>PPDA [20]</td>
<td>11</td>
<td>( T_P + 25</td>
</tr>
<tr>
<td>CAABAC</td>
<td>7</td>
<td>( T_P = 673.4</td>
</tr>
</tbody>
</table>

---

Fig. 4. Communication overhead versus number of data consumers.
requirements can get an access token from the CM. When
the context changes, only a unique contextual token will be
generated to define the changed contextual parameter and
users’ attributes still static. However, in [7], where the con-
textual attributes are handled as normal attributes, the KGC
has to generate decryption key for each user at every con-
text changing. In addition, the data owner has to re-encrypt
data given that the context changes lead to the revocation
of users’ attributes. Therefore, the computation cost will be linear
to the context changing and the number of users.

VII. EXTENDED CAABAC SCHEME FOR
GENERAL CONTEXT

The main construction in Section V presents the basic
scheme to integrate the contextual information into access
structures of the CP-ABE scheme. It considers that the con-
textual parameters are leaf nodes but it lacks a general method
to efficiently define access structure with multiple contextual
parameters that are appended to arbitrary nodes of the access
structure (leaf, nonleaf, or even root). In this section, we pro-
pose an extended CAABAC (E-CAABAC) scheme to provide
a practical method to construct relevant access structures for
general context. Then, we present a case study to apply our
construction in emergency scenario.

A. E-CAABAC Scheme Description

In this construction, we assume that the contextual tokens
are appended to arbitrary nodes. As shown in Fig. 6, we
denote $T^\text{loc}$, $T^\text{t}$ the contextual tokens related to location and
time, respectively. For instance, if at location loc$_1$, a user whose
attribute set satisfies a subpolicy $P_1$ can access data at time $t_1$. A
subpolicy can be either a single attribute or a set of multiple
nodes. In addition, a user whose attributes satisfy $P_1$ and $P_2$ at
loc$_1$ can get access privilege at time $t_2$, while a data consumer
whose attribute set only verifies $P_3$ cannot satisfy the policy
until he/she reaches time $t_3$.

The modification of the basic CAABAC construction
occurs during the encryption and decryption phases. We give
a detailed description of modified in Algorithms 5 and 6.

1) Encryption Algorithm: In the proposed E-CAABAC
scheme, the contextual information can be placed upon any
arbitrary node in the access structure (leaf, nonleaf, or even

Fig. 5. Communication and computational cost comparisons.

Fig. 6. Access structure of a general context.
Algorithm 6 E-CAABAC Decryption

Input: the ciphertext $CT$, the signature $\sigma$, the public parameters $PK$, the set of attributes $S$, the contextual token $T^{i}_{\ell}$

Output: the plaintext $K'_{\ell}$

1. The CM generates an access token $AT_{x}^{S_{i}} = H_{1}(F_{x})^{k_{i}}$
2. Upon receiving the access token $AT_{x}^{S_{i}}$, the user performs the following steps:
3. Compute $T^{i}_{\ell} = B^{i}_{\ell} - H_{2}(e(AT_{x}^{S_{i}}, K'_{\ell}))$
4. function (DecryptNode ($CT$, $\sigma$, $SK_{\ell}$, $x$))
5. if $x$ is an attribute leaf node then
6. if $att_{l} \in S$ then
7. $F_{x}^{att} = e(C_{x}, C_{x}, L_{x}, e(C, D_{x})$
8. $\cdot e(H_{1}(att_{l})^{-i}, s_{i}). e(H_{1}(att_{l}^{\prime}), g_{i}^{\prime})$
9. $\cdot e(g, g)^{r_{a}^{S_{i}}}$
10. End if
11. Else return $\perp$
12. End if
13. For all nodes $x$ (leaf and nonleaf node) related to a contextual token $T^{i}_{\ell}$ do
14. $F^{x}_{\ell} = (e(h, C_{x}, L_{x}, e(C, D_{x}))^{T^{i}_{\ell}}$
15. $\cdot (e(g, g)^{r_{a}^{S_{i}}}. e(H_{1}(att_{l})^{-i}, s_{i}). e(H_{1}(att_{l}^{\prime}), g_{i}^{\prime})^{x_{i}}$
16. End for
17. $F_{x} = F_{x}^{att} \cdot \prod_{j=1}^{n} F_{j}^{x}$
18. Else
19. $F_{x} = F_{x}^{att} = e(g, g)^{r_{a}^{S_{i}}}$
20. End if
21. End for
22. If $x$ is a nonleaf node then
23. For each child $z$ of $x$ do
24. $F_{z} = $ DecryptNode ($CT$, $\sigma$, $SK_{\ell}$, $z$)
25. End for
26. Let $S_{x}$ be an arbitrary $k_{x}$-sized set of child nodes of $x$ such that $F_{x} \neq \perp$
27. If $S_{x}$ exists then
28. $F_{x}^{att} = \prod_{z \in S_{x}} F_{z}^{\Delta_{x}(S_{x}(0))}$
29. $\cdot \prod_{z \in S_{x}} e(g, g)^{r_{a}^{S_{x}(0)}} \cdot e(H_{1}(att_{l})^{-i}, s_{i})$
30. $\cdot e(H_{1}(att_{l}^{\prime}), g_{i}^{\prime})$
31. where $i =$ index($z$) and $S_{x} = \{\text{index}(z) : z \in S_{x}\}$
32. Return $F_{x}^{att}$
33. Else
34. Return $F_{x}^{att} = \perp$
35. End if
36. End function
37. If $x$ is a root node then
38. $A = $ DecryptNode ($CT$, $\sigma$, $SK_{\ell}$, $R_{i}$)
39. $= e(g, g)^{r_{a}^{S_{i}}}$
40. End if
41. The decryption is performed as follows:
42. $K'_{\ell} = C_{A}^{e(g, g)^{r_{a}^{S_{i}}}}$
43. if $e(x, g) = e(H_{1}(K'_{\ell}), g_{i}^{\prime})$ then
44. $K'_{\ell}$ is valid. End if

2) Decryption Algorithm: In this phase, we present the decryption algorithm for the case where the contextual information may be appended to arbitrary nodes. To decrypt a ciphertext, a data consumer has to verify not only the set of attributes but also the contextual information.

B. Case Study: Emergency Situations

Emergency conditions (such as accident, a building on fire, natural disaster, emergency healthcare, etc.) are highly dynamic situations that require accurate, relevant, and timely response. Therefore, contextual information including, the situation sensitivity, time, and location are fundamental to make the right decision at the right time. As shown in Fig. 7, contextual tokens are arbitrarily appended into the access structure to enable data consumers who are within a predefined distance from the location of incident access data at the time of emergency. We denote $T^{EM}$, $T^{loc}$, $T^{i}$ the contextual tokens related to emergency event, location, and time, respectively. Location tokens present the set of locations areas from within a data consumer can access data. When an emergency occurs, the CM gives access tokens to users verifying the location constraints and having a valid date of access. Indeed, it checks the validity of the query generation time, and if it was within the allocated time. In emergency situation, all users satisfying a subpolicy $tP_{1}$ can access data not later than $t_{1}$. In this case, $t_{1}$ is the valid period of the time-related token. In addition, a user whose attributes satisfy $P_{3}$ at loc$_{2}$ can get access privilege at time that does not exceed $t_{2}$.

VIII. Conclusion

In this article, we proposed a new CAABAC scheme to address adaptive and fine-grained access control issue in the IoT. The proposed approach seamlessly integrates the contextual information with the basic CP-ABE scheme to incorporate dynamic special attributes into an access policy. It not only provides fine-grained access control, but also handles dynamic attributes (time, location, situation sensitivity, etc.) in order to grant adaptive and dynamic access to data. From a security perspective, we proved that the root). The smart thing encrypts the challenge $M$ using a symmetric key $K_{s}$, then the IoT gateway performs the following steps to encrypt $K_{s}$ on the basis of CP-ABE scheme.
proposed scheme satisfies the different security requirements and solves the key escrow problem of the CP-ABE algorithm. The performance analysis demonstrates the effectiveness and efficiency of the proposed CAABAC scheme that outperforms the existing access control schemes. We further design an extended CAABAC (E-CAABAC) scheme to deal with general and multifactor context while defining how contextual tokens are placed in the access structure.

ACKNOWLEDGMENT

This work is achieved as part of the European project ITEA PARFAIT [28], which is partially funded by FEDER (European Regional Development Fund), BPIFRANCE, and the BFC region (Bourgogne-Franche-Comté).

REFERENCES


