Context-Aware Adaptive Remote Access for IoT Applications

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

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

A. Arfaoui is with the Digital Security Unit, SupCom University of Carthage, Ariana, Tunisia, and also with the DRIVE EA1859, University of Bourgogne Franche Comté, 58000 Nevers, France (e-mail: amel.arfaoui@u-bourgogne.fr).

S. Cherkaoui is with the INTERLAB Research Laboratory, University of Sherbrooke, Sherbrooke, QC, Canada (e-mail: soumaya.cherkaoui@usherbrooke.ca).

A. Kribeche and S. M. Senouci are with the DRIVE EA1859, University of Bourgogne Franche Comté, 58000 Nevers, France (e-mail: ali.kribeche01@u-bourgogne.fr; sidi-mohammed.senouci@u-bourgogne.fr).

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, if 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

TABLE I
STATE-OF-THE-ART EVALUATION SUMMARY [(+) CONSIDERED PARAMETER/(−) NOT CONSIDERED PARAMETER]

Scheme	Data access	Confidentiality	Integrity	Authentication	Context-awareness	Privacy	Adaptation	Complexity
[8]	-	+	+	+	-	+	-	High
[9]	RBAC	+	+	+	-	+	-	Medium
[10]	-	+	+	+	-	+	-	Low
[11]	-	+	+	+	-	+	-	Low
[13]	capBAC	+	+	+	-	+	-	Medium
[14]	capBAC	+	+	+	-	+	+	Medium
[17]	RBAC	+	+	+	-	+	-	High
[20]	RBAC	+	+	+	-	+	-	High

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 $\mathbb{G}_1$ be a cyclic additive group of prime order q and $\mathbb{G}_2$ be a cyclic multiplicative group of the same order q. A bilinear pairing is a map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ and satisfies the following properties.

1) *Bilinear:* A map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is bilinear if and only if $\forall\, P, Q \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_q$, we have $e(aP, bQ) = e(P, Q)^{ab}$.

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 $(g, g^a, g^b, g^c), \forall a, b, c \in \mathbb{Z}_q$.

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(1^λ):* Given a security parameter λ, 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, $\mathcal{T}$):* It takes the public parameters PK, a plaintext M, and an access structure $\mathcal{T}$ 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 $\perp$.

C. Access Policy Structures

An access structure $\mathcal{T}$ 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 $(\text{att}_0, \ldots, \text{att}_3)$, 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 $T_x^{c_j}$ can be integrated in an access structure to restrict user's access permission by the contextual parameter c_j 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.

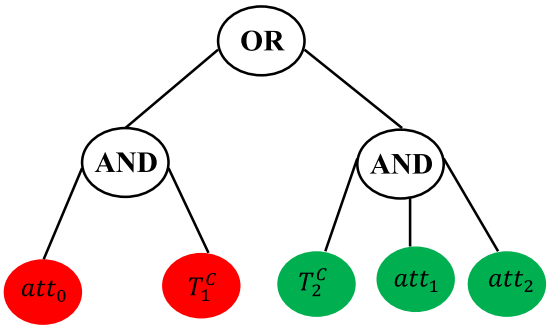

Fig. 1. Example of access structure.

For example, in Fig. 1, the contextual token T_2^C is related to the contextual parameter C and users who satisfy "$\text{att}_1 \wedge \text{att}_2$" 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 key 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).

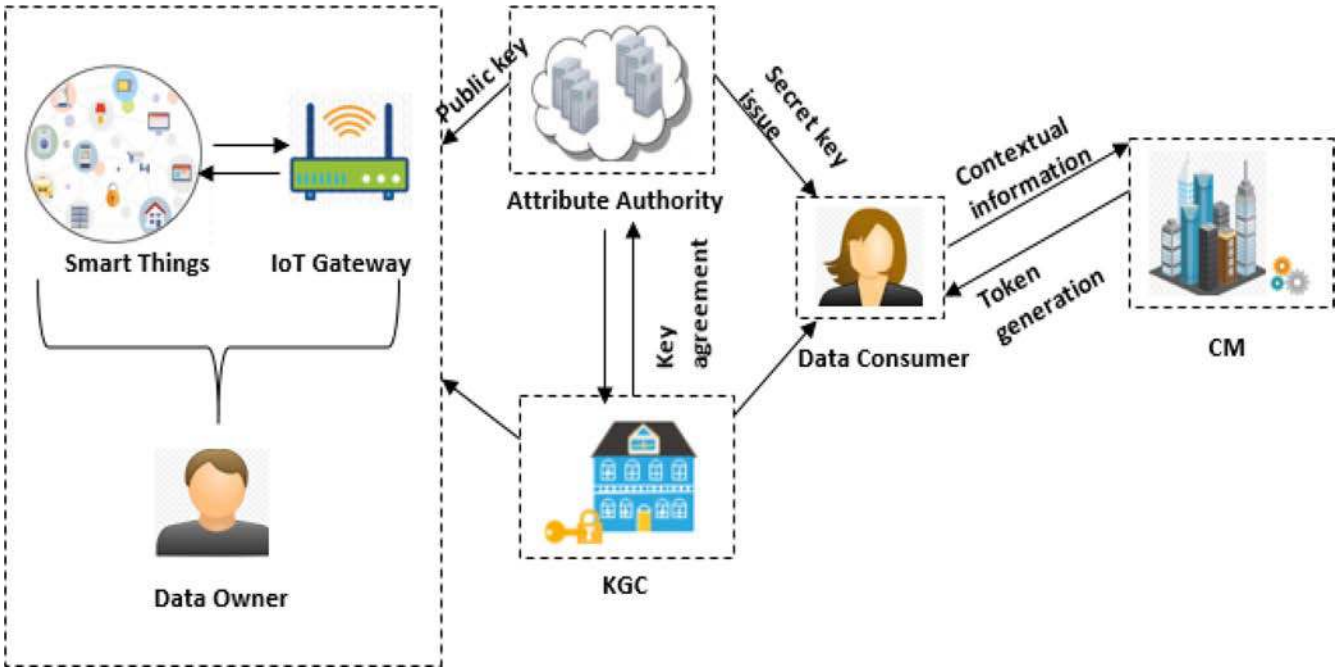

Fig. 2. System model.

TABLE II
VARIABLES AND THEIR DESCRIPTIONS

Notation	Description
q	A large prime number
$\mathbb{G}_1$	An additive group with order q
$\mathbb{G}_2$	A multiplicative group with order q
e	A bilinear pairing
g	A generator of the group $\mathbb{G}_1$
H_1, H_2	One-way hash functions
PK, MK	System public key and master key
PK_i	The public key of entity i
SK_i	The secret key of entity i
γ_{GW}	The signature key of the gateway
K_{ver}	The verification key
S	The attributes set of user
$\mathcal{T}$	An access structure
$T_x^{c_j}$	A contextual token for a parameter $c_j \in$ {time, location, data type, situation sensitivity}
AT	Access token for a given context
TK	Authentication token
$\mathbb{F}_{c_j}$	Unified format of the contextual parameter c_j

B. Security Model

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 an 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,

Algorithm 1 System Initialization

1. Let $\mathbb{G}_1$ be a bilinear group of prime order q, g a generator of $\mathbb{G}_1$, $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ a bilinear map
2. Let $H_1 : \{0, 1\}^* \rightarrow \mathbb{G}_1, H_2 : \mathbb{G}_2 \rightarrow \mathbb{Z}_q^*$ be one-way hash functions
3. For $i \in$ Zq and a set S= $\{s_1, s_2 \ldots, s_m \in \mathbb{Z}_q\}$, the Lagrange coefficient $\Delta_{i,s} = \prod_{l \in S,\ l \neq i} \frac{(x-l)}{(i-l)}$
4. The AA chooses random exponents α_1, $\beta \in \mathbb{Z}_q$, sets $h = g^{\beta}$ and generates the public/private pair key: $PK_A = \{\mathbb{G}_1, h, g, e(g, g)^{\alpha_1}\}$ /MK_A= $\{\alpha_1, \beta\}$
5. The KGC selects a random parameter α_2 and computes the public key $PK_{\text{KGC}} = e(g, g)^{\alpha_2}$ and the secret key $MK_{\text{KGC}} = \{\alpha_2\}$
6. The KGC selects a signature key $\gamma_{GW} \in \mathbb{Z}_q$, and calculates the verification key $K_{ver} = g^{\gamma_{GW}}$
7. KGC and AA publish the public parameters of the system $PK = \{\mathbb{G}_1, h, g, e(g, g)^{\alpha}, K_{ver}\}$ where $\alpha = \alpha_1 + \alpha_2$ and kept secret the master key $MK = \{\{\alpha_1, \beta\}, \{\alpha_2\}\}$
8. The CM defines a secret key δ_{c_j} for each contextual parameter $c_j \in$ {location, time, data type, situation sensitivity}, $\forall\ j \in [1, N]$ where N is the number of considered contextual parameters. The public key $PK_{c_j} = \{\mathbb{F}_{c_j},\ \gamma_{c_j} = g^{\delta_{c_j}}\}$

Algorithm 2 Key Commitment

Input: The public parameters PK, the master key MK, the set of attributes S
Output: SK_U

1. KGC picks a random $\tau \in \mathbb{Z}_q$, and computes $V = g^{X/\tau} = g^{(\alpha_1+\alpha_2)\beta/\tau}$, and sends $\{V, \text{PoK}(\tau, X)\}$ to AA.
2. AA chooses a random $\tau_1 \in \mathbb{Z}_q$ and computes $V_1 = V^{\tau_1/\beta}$, $X_1 = h^{r\tau_1}$, then, it sends $\{V_1, X_1, \text{PoK}(\tau_1, \beta, r)\}$ to KGC
3. KGC picks a random number $\tau_2 \in \mathbb{Z}_q$ and computes $V_2 = (V_1^{\tau}.X_1)^{\tau_2}$, then, it sends $\{V_2, \text{PoK}(\tau_2)\}$ to AA
4. AA computes $V_3 = {V_2}^{1/\tau_1} = (g^{\alpha_1+\alpha_1}.h^r)^{\tau_2}$ and sends $\{V_3, \text{PoK}(\tau_1)\}$ to KGC
5. KGC computes $SK_{\text{KGC}} = D = {V_3}^{1/\tau_2} = g^{\alpha}h^r$ and sends the partial secret key to the user u
6. AA generates the secret keys of the attribute set S of user u as follows: $SK_{AA,u} = \{D_i = H_1(\text{att}_i)^r,\ \forall\ \text{att}_i \in S,\ L = g^r\}$
7. The user determines his/her personalized secret key as $SK_u = \{D = g^{\alpha}h^r,\ L = g^r,\ D_i = H_1(\text{att}_i)^r,\ \forall\ \text{att}_i \in S\}$

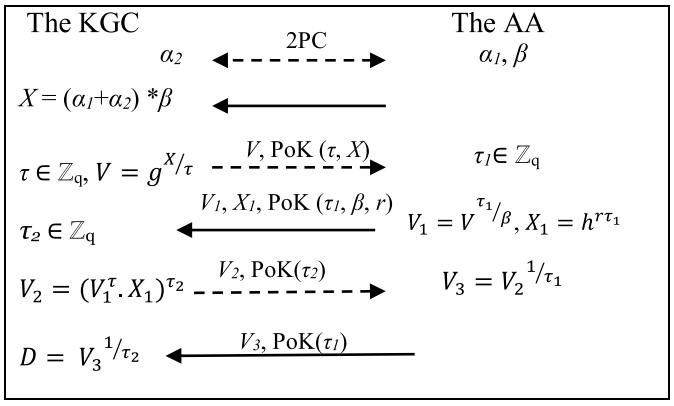

Fig. 3. Enhanced key issuing protocol. "PoK" is a proof of knowledge of the secret values exploited in the computation.

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 normal 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 $c_j \in \mathbb{F}_{c_j}$. $\mathbb{F}_{c_j}$ 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 first a random unique 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 $MK_A = \{\alpha_1, \beta\}$ and KGC inputs $MK_{\text{KGC}} = \{\alpha_2\}$. Finally, KGC gets $X = (\alpha_1+\alpha_2).\ \beta \bmod q$ [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 $MK_A = \{\alpha_1, \beta\}$ from AA and $MK_{\text{KGC}} = \{\alpha_2\}$ from KGC, and returns $X = (\alpha_1 + \alpha_2).\ \beta \bmod q$ 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_s where K_s is a preshared secret key with the IoT gateway. After that, the IoT gateway encrypts K_s based on an access tree $\mathcal{T}$ 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, σ

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 $\in Z_q$ and set $q_R(0) = s$
3. Select d_R 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 $q_x(0)= q_{parent(x)}(index(x))$
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 $r_{c_j} \in \mathbb{Z}_q$
10. Generate a contextual token $T_x^{c_j} = \{A_x^{c_j} = g^{r_{c_j}},\ B_x^{c_j} = q_x(0) + H_2(e\left(H_1\left(\mathbb{F}_{c_j}\right), \gamma_{c_j}\right)^{r_{c_j}})\}$
11. **End if**
12. The ciphertext CT is constructed based on the access tree $\mathcal{T}$ as follows:

$$CT = \begin{pmatrix} \mathcal{T}, \tilde{C} = K_s e(g,g)^{\alpha s}, C = g^s \\ \forall \mathrm{x}, \mathrm{i} \in [1,\mathrm{n}], \mathrm{j} \in [1,\mathrm{N}], C_x = h^{q_x(0)}, C'_x = H_1(att_i)^{-s} \\ T_x^{c_j} = (A_x^{c_j}, B_x^{c_j}) \end{pmatrix}$$

13. Compute $\sigma = H_1(K_s)^{\gamma_{GW}}$ where γ_{GW} is the signature key of the gateway

data type, and situation sensitivity (emergency/normal). 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 $<\mathrm{CT}\sigma, r, AES(K_s, M)>$. 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 $M_2 = <H'_1,\ SK>_{PK_{GW}}$ where $H'_1 = H_1(M'||r)$ 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 $H'_1 = H_1$. If succeeds, the IoT gateway generates an authentication token TK for a period T_e and sends $M_3 = <TK,\ T_e,\ \mathrm{ID}_i>$ 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 $H'_1 = H_1(TK||T_e||\mathrm{ID}_i)$ to the IoT device, this last verifies if $H'_1 = H_1$. 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.

Algorithm 4 Decryption

Input: The ciphertext CT, the signature σ, the public parameters PK, the set of attributes S, the contextual token $T_x^{c_j}$
Output: The plaintext K'_s

1. The CM generates an access token $AT_x^{c_j} = H_1\left(\mathbb{F}_{c_j}\right)^{\delta_{c_j}}$
2. Upon receiving the access token $AT_x^{c_j}$, the user performs the following steps:
3. Compute $T_x^{c_j\prime} = B_x^{c_j} - H_2(e(AT_x^{c_j}, A_x^{c_j})$
4. **function** (DecryptNode (CT, σ, SK_u, x))
5. **If** x is a leaf node related to a contextual token $T_x^{c_j}$ **then**
6. $F_{x,T_x^{c_j}} = \left(e\left(h.C'_x, L\right).e(C, D_i)\right)^{T_x^{c_j\prime}}$
$= \left(e(g,g)^{r\beta}.e\left(H_1(att_i)^{-s}, g^r\right).e(H_1(att_i)^r, g^s\right)^{T_x^{c_j\prime}}$
$= e(g,g)^{r\beta T_x^{c_j\prime}}$
7. **Else if** x is an attribute leaf node **then**
8. **if** $att_i \in S$ **then**
9. $F_x = e\left(C'_x.C_x, L\right).e(C, D_i)$
$= e\left(H_1(att_i)^{-s}.h^{q_x(0)}, g^r\right).e\left(H_1(att_i)^r, g^s\right)$
$= e(g,g)^{r\beta q_x(0)}$
10. **Else** return $\perp$
11. **End if**
12. **Else**
13. **For** each child z of x **do**
14. $F_z = \mathrm{DecryptNode}(\mathrm{CT}, \sigma, SK_u, \mathrm{z}))$
15. **End for**
16. Let S_x be an arbitrary k_x-sized set of child nodes of x such that $F_z \neq \perp$
17. **If** S_x exists **then**
18. $F_x = \prod_{z \in S_x} F_z^{\Delta_{i,S'_x(0)}}$
$= \prod_{z \in S_x} (e(g,g)^{r\beta q_z(0)})^{\Delta_{i,S'_x(0)}}$
$= \prod_{z \in S_x} e(g,g)^{r\beta q_x(i)\Delta_{i,S'_x(0)}}$
$= e(g,g)^{r\beta q_x(0)}$
where $i = \mathrm{index}(z)$ and $S'_x = \{\mathrm{index}(z) : z \in S_x\}$
19. Return F_x
20. **Else**
21.
22. Return $F_x = \perp$
23. **End if**
24. **End if**
25. **End function**
26. **If** x is a root node **then**
27. $A = \mathrm{DecryptNode}(\mathrm{CT}, \sigma, SK_u, \mathrm{R}))$
$= e(g,g)^{r\beta s}$
28. **End if**
29. The decryption is performed as follows:
30. $K'_s = \frac{\tilde{C}.A}{e(g^s, g^\alpha.h^r)}$
31. **If** $e(\sigma, g) = e(H_1\left(K'_s\right), g^{\gamma_{GW}})$ **then**
32. K'_s is valid
33. **End if**

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:

$$K'_s = \frac{\tilde{C}.A}{e(g^s, g^\alpha.h^r)} = \frac{K_s e(g,g)^{\alpha s}.e(g,g)^{r\beta s}}{e\left(g^s, g^{\alpha+r\beta}\right)}$$

$$= \frac{K_s e(g,g)^{(\alpha+r\beta)s}}{e(g,g)^{(\alpha+r\beta)s}}$$
$$= K_s.$$

Thus, if $e(\sigma, g) = e(H_1(K'_s), g^{\gamma_{GW}}), K'_s$ is valid. Therefore, the data consumer can decrypt the ciphertext using K'_s 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 $r \in \mathbb{Z}_q$. 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 $\mathcal{B}$ and an adversary $\mathcal{A}$. 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 $\{\delta_{c_j}, PK_{c_j}, \mathbb{F}_{c_j}\}$. Then it gives public parameters to the adversary $\mathcal{A}$.
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, $\mathcal{B}$ flips a random bit $\nu \in \{0, 1\}$, and encrypts M_ν under $\mathcal{T}$. The ciphertext is sent to $\mathcal{A}$.
4) *Phase 2:* In this phase, $\mathcal{A}$ 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:* $\mathcal{A}$ outputs a guess ν' of ν. The advantage of the adversary in this game is defined as

$$Adv_A = \left| \Pr(\nu' = \nu) - \frac{1}{2} \right|.$$

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.

Our further analysis 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 $\mathcal{A}$, let p be a bound on the total number of group elements that $\mathcal{A}$ receives from the performed queries to the oracles for the hash function, groups $\mathbb{G}_1, \mathbb{G}_2$, the bilinear map e, and from its interaction with the security game, in which $\mathbb{G}_1$ is bilinear group of prime order q. We have that the advantage of the adversary in the security game is $O(p^2/q)$.

Proof: In the basic CP-ABE security game, the challenger constructs the ciphertext component $\tilde{C}$ that is either $M_0 e(g,g)^{\alpha s}$ or $M_1 e(g,g)^{\alpha s}$. We consider as [21], a modified game where $\tilde{C}$ may be $e(g,g)^{\alpha s}$ or $e(g,g)^{\theta}$ and $\theta \in \mathbb{Z}_q$ 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 $M_0 e(g,g)^{\alpha s}$ and in another case between $e(g,g)^{\theta}$ and $M_1 e(g,g)^{\alpha s}$. Consequently,

any adversary that has advantage ε 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 $\alpha_1, \beta, \alpha_2 \in \mathbb{Z}_q$ and sets $h = g^{\beta}, u_1 = e(g,g)^{\alpha_1}, \nu_1 = g^{\alpha_1}, u_2 = e(g,g)^{\alpha_2}, \nu_2 = g^{\alpha_2}$, and $\alpha = \alpha_1 + \alpha_2$. For each query on an attribute att(x), the challenger returns g^{d_i} and stores (d_i, att(x)) into *H-list* where $d_i \in \mathbb{Z}_q$. B defines the format $\mathbb{F}_{c_j}$ 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 $PK_{c_j} = \{\mathbb{F}_{c_j}, \ \gamma_{c_j} = \ g^{\delta_{c_j}}\}$. Finally, B sends the public parameters $PK = \{g, h, u = u_1.u_2, \ \bigcup \mathbb{F}_{c_j}, \ \gamma_{c_j} \forall\, j \in [1, N]\}$ to the adversary.
2) *Phase 1:* In this phase, $\mathcal{B}$ answers a user's u secret key queries for an attribute set S_u and a given context C. The challenger chooses first $r_u \in \mathbb{Z}_q$ and computes $D = g^{\alpha} h^{r_u}, \ L = g^{r_u}, \ \ D_i = H(\text{att}(x))^{r_u} = g^{\alpha}.g^{\beta r_u}$. Then, $\mathcal{B}$ sends the secret key to $\mathcal{A}$ and stores (SK, u, S_u) into *SK-list.*
3) *Challenge:* $\mathcal{A}$ submits two equal-length challenge messages M_0, $M_1 \in \mathbb{G}_2$ and an access tree $\mathcal{T}$ to the challenger where any secret key issued to $\mathcal{A}$ cannot satisfy $\mathcal{T}$. $\mathcal{B}$ chooses $s \in \mathbb{Z}_q$ and uses linear secret sharing technique to construct shares λ_x of s for all attributes att(x) in $\mathcal{T}$ where λ_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

$$\check{C} = e(g,g)^{\theta}, \ C = g^{s}, \ C_x = g^{\beta\lambda_i} \cdot H(\text{att}(x))^{-s}$$
$$C'_x = H(\text{att}(x))^{-s}.$$

For each contextual token $T_x^{c_j}$, $\mathcal{B}$ sets $T_x^{c_j} = q_x(0) = s_x^0$.

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 $\nu = \eta/\psi$ in the variables $\theta, \alpha, \beta, r_u, \lambda_i, d_i$ and s. Suppose that there are distinct rational functions $\nu = \eta/\psi$, $\nu' = \eta'/\psi'$. 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, $\nu = \nu' \Longleftrightarrow \eta\psi' - \eta'\psi = 0$. 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 $\mathcal{A}$ can distinguish between θ and αs, if there are two distinct queries ν and ν' giving the same output. Assume that $\nu = \gamma\alpha s$ and $\nu' = \gamma'\theta$, thus, $\nu' - \nu = \gamma'\theta - \gamma\alpha s$ such that $\gamma\alpha s = \nu - \nu' + \gamma'\theta$. We will prove that $\mathcal{A}$ 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 $\mathcal{A}$. The adversary can obtain the transcript $\{g, g^{\beta}, g^{s}, g^{\beta\lambda_i}, g^{-sd_i}, g^{\alpha}.g^{\beta r_u}, g^{r_u}, g^{r_u d_i}, g^{\delta_{c_j}}, A_x^{c_j}, B_x^{c_j}\}$. Another transcript can be set as $\{g, g^{\beta}, g^{s}, g^{\beta\lambda_{i'}}, g^{-sd_{i'}}, g^{\alpha}.g^{\beta r_{u'}}, g^{r_{u'}}, g^{r_{u'}d_i{}'}, g^{\delta_{c_j}}, A_x^{c_j}, B_x^{c_j}\}$. Since β is not relevant to construct a query involving αs, we ignore all factors of β. We find that there are two types of outputs with αs from two transcripts in $\mathbb{G}_2$. One is $(\alpha + \beta r_u)s$ (resp.$((\alpha + \beta r_{u'})s)$ by pairing s with $(\alpha + \beta r_u)$ (resp.$(\alpha + \beta r_{u'})$). To generate $\gamma\alpha s$, we need to eliminate $\beta r_u s$, however, no combination can satisfy $\beta r_u s$. Another combination to form $\gamma\alpha s$ is $-(\alpha + \beta r_u)sd_i$ (resp.$-(\alpha + \beta r_{u'})sd_{i'}$). Thus, we need to eliminate $\beta r_u sd_i$. For the element sd_i, we need an element with βr_u and for $r_u d_i$, we should find an element with βs. Nevertheless, none of the combinations can satisfy our requirement. From the above analysis, we can conclude that $\mathcal{A}$ 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 $\mathcal{A}$ exists with a non-negligible advantage ε in the selectively security game against CAABAC. 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 $\mathcal{B}$ that can break the DBDH assumption with a non-negligible advantage.

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

1) *System Initialization:* $\mathcal{A}$ selects a challenge access structure $\mathcal{T}$ 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 $H_1(\mathbb{F}_{c_j})$. For any contextual parameter $c_{j'} \neq c_j$, $\mathcal{B}$ sets $PK_{c_{j'}} = \{\mathbb{F}_{c_{j'}} \ , \ \gamma_{c_{j'}} = \ g^{\delta_{c_{j'}}}\}$ where $\delta_{c_{j'}}$ is randomly chosen for each contextual parameter $c_{j'}$. For the contextual parameter c_j, the challenger sets $PK_{c_j} = \{\mathbb{F}_{c_j} \ , \ \gamma_{c_j} = \ B\}$ and sends public parameters to $\mathcal{A}$.
2) *Phase 1:* It is the same as proof.
3) *Challenge:* $\mathcal{A}$ submits two equal-length random messages M_0, $M_1 \in \mathbb{G}_2$ to $\mathcal{B}$. Then, the challenger flips a random coin $\nu \in \{0, 1\}$, and encrypts M_ν 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, $\mathcal{B}$ sets $H_1(\mathbb{F}_{c_j}) = g^{d_{c_j}}$ where d_{c_j} is a random number of $\mathbb{Z}_q$. Then, the challenger picks a random number r_{c_j} and calculates $A_x^{c_j} = g^{r_{c_j}}$ and $B_x^{c_j} = q_x(0) + H_2(e(g^{d_{c_j}}, \ B\ )^{r_{c_j}})$;
   b) otherwise, the random oracle sets $A = A_x^{c_j} = g^{r_{c_j}}$ and $H_1(\mathbb{F}_{c_j}) = C.g^{d_{c_j}}$. Thus, $B_x^{c_j}$ is computes as follows:

$$B_x^{c_j} = q_x(0) + H_2(e\big(H_1\big(\mathbb{F}_{c_j}\big), \ B\ \big)^{r_{c_j}}$$

$$= q_x(0) + H_2\Big(e\Big(C.g^{d_{c_j}},\ B\ \Big)^{r_{c_j}}\Big).$$

For the set of attributes S_u, 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

$$\begin{aligned} k &= H_2\Big(e\Big(C.g^{d_{c_j}},\ B\ \Big)^{r_{c_j}}\Big) \\ &= H_2\Big(e\Big(g^{c+d_{c_j}},\ g^b\ \Big)^{r_{c_j}}\Big) \\ &= H_2\Big(e(g,g)^{br_{c_j}\left(c+d_{c_j}\right)}\Big). \end{aligned}$$

For $a = r_{c_j}$, we can compute $e(g,g)^{abc}$ from the argument of H_2. Thus, $T_x^{c_j} = (A_x^{c_j},\ B_x^{c_j})$ is a valid contextual token. Furthermore, the interpolation can reconstruct the secret s, and the decryption will recover the plaintext.

Otherwise, if $b = 1, Z = e(g,g)^z$ 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_ν.

1) *Phase 2:* $\mathcal{B}$ 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 $AT_x^{c_j'}$ to $\mathcal{A}$ for the contextual parameter c_j, it is no useful for $\mathcal{A}$ to get the secret related to c_j.
2) *Guess:* The adversary outputs a guess ν' of ν. If $\nu' = \nu$, the challenger will output $b = 0$ to indicate it was given a valid DBDH tuple. Otherwise, it will output $b' = 1$ to indicate that it was given a random 4-tuple.

In the case where $b = 1, \mathcal{A}$ gains no information about ν. Therefore, we have the probability $\Pr(\nu \neq \nu' | b = 1) = 1/2$. As $\mathcal{B}$ guesses $b' = 1$ when $\nu \neq \nu'$, we have $\Pr(b = b' | b = 1) = \Pr(b' = 1 | b = 1) \ = 1/2$.

In the case where $b = 0, \mathcal{A}$ is given a valid contextual token with non-negligible advantage ε and thus, we have $\Pr(\nu = \nu' | b = 0) = \varepsilon + (1/2)$. As $\mathcal{B}$ guesses $b = b'$ when $\nu = \nu'$, we have $\Pr(b = b' | b = 0) = \Pr(b' = 0 | b = 0) \ = \varepsilon + (1/2)$. Hence, we can conclude that the advantage of $\mathcal{B}$ in the DBDH game is

$$\begin{aligned} \Pr\big(b = b'\big) \ - \frac{1}{2} &= \Pr\big(b = b' | b = 1\big)\ .\Pr(b = 1) \\ &\quad + \Pr\big(b = b' | b = 0\big)\ .\Pr(b = 0)\ - \frac{1}{2} \\ &= \frac{1}{2}.\frac{1}{2} + \left(\varepsilon + \frac{1}{2}\right).\frac{1}{2} - \frac{1}{2} \\ &= \frac{\varepsilon}{2}. \end{aligned}$$

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

TABLE III
SECURITY AND FUNCTIONALITY FEATURES COMPARISON [(+) PROVIDED FEATURE/(−) NOT PROVIDED FEATURE]

Scheme	CP-ABE [17]	H-CLSC [7]	PPDAS [20]	CAABAC
Confidentiality	+	+	+	+
Integrity	+	+	+	+
Mutual authentic. User-Gateway	+	+	+	+
Mutual authentic. User-Smart thing	+	-	-	+
Flexible combination	-	+	-	+
Context-privacy	-	+	-	+
No. Burdensome revocation	-	-	-	+
No. Key escrow	-	+	+	+

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 $D = g^\alpha h^r$ 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 $\mathbb{G}_1$ and $\mathbb{G}_2$ is set to 20-byte prime. For an 80-bit security level, p should be a 64-byte prime if $\mathbb{G}_2$ is a q-order subgroup of the multiplicative group of the finite field F_{p2}^*. According to [17], we can set the length of p to 42.5 bytes in the finite field F_{p3}^*. The length of an element in group $\mathbb{G}_1$ 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 $\mathbb{G}_1$ 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

TABLE IV
STORAGE OVERHEAD COMPARISON

Scheme	storage overhead
CP-ABE [17]	$(2*\|\text{att}\|+1)\ \|\mathbb{G}_1\|$
H-CLSC [7]	$\|\mathbb{G}_1\|$=65 bytes
PPDAS[20]	$(3*\|\text{att}\|+13)\ \|\mathbb{G}_1\|+2\|Z_q^*\|$
CABAAC	$(\|\text{att}\|+2)\ \|\mathbb{G}_1\|$

TABLE V
COMMUNICATION OVERHEAD COMPARISON

Scheme	IoT gateway (bytes)	Smart thing (bytes)
CP-ABE [17]	$5\|p\|+24$=236.5	$10\|p\|+76$=501
H-CLSC[7]	$2\|\mathbb{G}_1\|+n\ \|Z_q^*\|+\|M\|+w$=180 (1 user)	-
PPDAS[20]	$27\|p\|+31$=1178.5	$\|p\|+1$=43.5
CABAAC	$6\|p\|+4\|Z_q^*\|+24$=359	$\|AES(K_s, M)\|$=16

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 $\{SK_u = \{D = g^{\alpha}h^r,\ L = g^r,\ D_i = H_1(\text{att}_i)^r,\ \forall\ \text{att}_i \in S\}\}$, whose size is $(|\text{att}| + 2)|\mathbb{G}_1|$, 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 message $<CT, \sigma, r, \text{ID}_i, TK, T_e,\ AES(K_s, M)>$ whose size is $|T| + |\check{C}| + |C| + |C'_x| + |T_x^{\mathcal{C}_j}| + |C_x| + |\sigma| + |r| + |\text{ID}_i| + |TK| + |T_e| + |AES(K_s, M)|$. We assume as [17] that $|\text{ID}_i|$, $|T_e|$, and $|T|$ have 1-byte, 1-byte, and 4-bytes, respectively. In addition, the IoT gateway has to send to the smart thing the message $<\text{ID}_i, TK, T_e>$ whose size is $|\text{ID}_i| + |TK| + |T_e|$. The smart thing has only to send to the IoT gateway the encrypted challenge M with a symmetric key, so, the message size is $|AES(K_s, M)|$. As illustrated in Fig. 4, the communication cost of H-CLSC [7] linearly increases with the number of users but in our scheme, it is independent of the number of users.

4) Computation Cost: For the computation cost comparison, we consider the operations affecting the communication overhead in both the IoT device and the IoT gateway. Particularly, we focus on the pairing, exponentiation, multiplication, and symmetric encryption operations. We denote T_E, T_M, T_P, and T_{Enc}, the computation time required for one exponentiation operation, one scalar multiplication in $\mathbb{G}_1$, one pairing operation, and one symmetric encryption (AES-128), respectively.

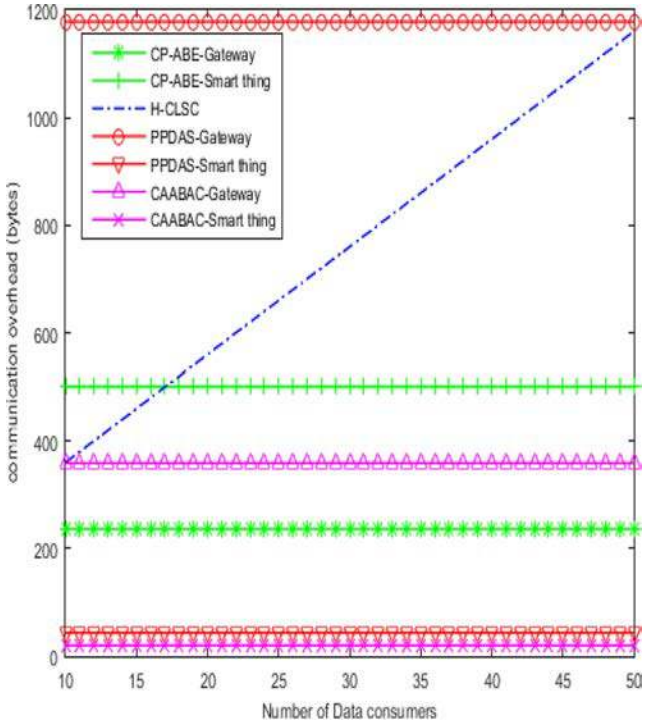

Fig. 4. Communication overhead versus number of data consumers.

TABLE VI
COMPUTATION COST COMPARISON

Scheme	IoT gateway (ms)	Smart thing (ms)
CP-ABE [17]	$5T_P$=481	$10T_P+1T_{Enc}$=962.1
H-CLSC[7]	$3T_P+6T_M+T_{Enc}$=472.72	-
PPDAS[20]	$11T_P+25T_E+4T_M$ =2520.38	$1T_E$=53.85
CAABAC	$7T_p$=673.4	$1\ T_{Enc}$=0.1

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 $\boldsymbol{T_x^{\mathcal{C}_j}}$ is generated for all users and only the data consumer whose context satisfies the context

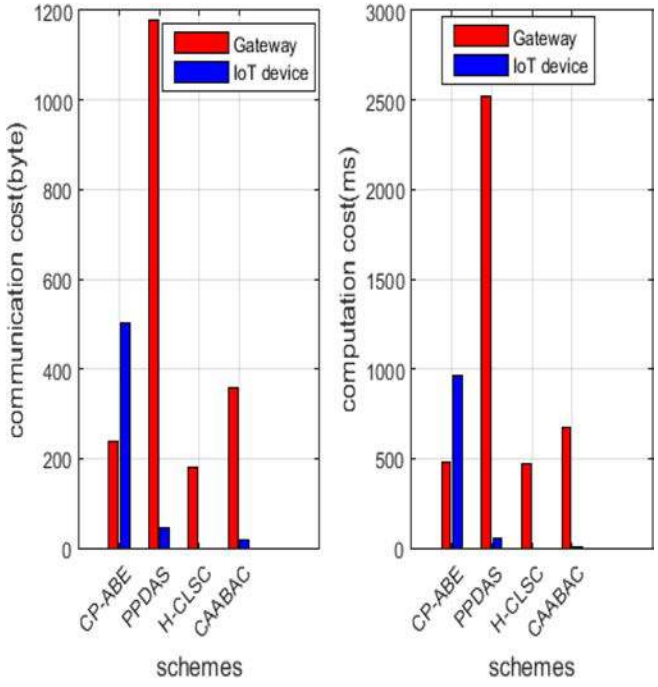

Fig. 5. Communication and computational cost comparisons.

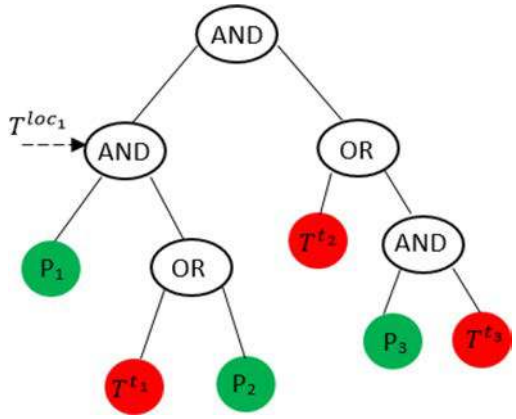

Fig. 6. Access structure of a general context.

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 contextual attributes are handled as normal attributes, the KGC has to generate decryption key for each user at every context 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 contextual parameters are leaf nodes but it lacks a general method

Algorithm 5 E-CAABAC Encryption

Input: An access tree $\mathcal{T}$, public parameters PK, a plaintext M, contextual parameter c_j, symmetric key K_s
Output: The ciphertext CT, σ

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_R random points from Z_q to completely define the polynomial q_R
4. **For** each node x in the tree $\mathcal{T}$, it is associated with two values q_x^0 and q_x^1
5. **If** x is related to a contextual parameter c_j, **then**, choose random $s_x^{c_j} \in \mathbb{Z}_q$ for the contextual token.
6. **End if**
7. The value q_x^1 is computed as:
$$\begin{cases} q_x^1 = q_x^0 - \sum_{j=1}^{N} s_x^{c_j} \text{ if } x \text{ is related to } \{c_j,\ j \in [1,\ N]\} \\ q_x^1 = q_x^0 \text{ otherwise} \end{cases}$$
8. **For** any nonleaf node x, the polynomial q_x is randomly chosen with $q_x(0) = q_x^1$ and its degree $d_x = k_x - 1$.
9. **End for**
10. **For** any node x except the root node, set $q_x^0 = q_{parent(x)}\ (index(x))$. **End For**
11. **End for**
12. **If** x is a node related to the contextual parameter c_j **then**
13. Choose a random number $r_{c_j} \in \mathbb{Z}_q$
14. Generate a contextual token $T_x^{c_j} = \{A_x^{c_j} = g^{r_{c_j}},\ B_x^{c_j} = s_x^{c_j} + H_2(e\left(H_1\left(\mathbb{F}_{c_j}\right), \gamma_{c_j}\right)^{r_{c_j}})\}$
15. **End if**
16. The ciphertext CT is constructed based on the access tree $\mathcal{T}$ as follows:
$$CT = \begin{pmatrix} \mathcal{T}, \tilde{C} = K_s e(g,g)^{\alpha s},\ C = g^s \\ \forall \text{x, i} \in [1, \text{n}], \text{j} \in [1, \text{N}], C_x = h^{q_x^1}, C_x' = H_1(att_i)^{-s} \\ T_x^{c_j} = (A_x^{c_j}, B_x^{c_j}) \end{pmatrix}$$
17. Compute $\sigma = H_1(K_s)^{\gamma_{GW}}$ where γ_{GW} is the signature key of the gateway

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 propose 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^{loc}, T^t the contextual tokens related to location and time, respectively. For instance, 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

Algorithm 6 E-CAABAC Decryption

Input: the ciphertext CT, the signature σ, the public parameters PK, the set of attributes S, the contextual token T_x^{cj}
Output: the plaintext K_s'

1. The CM generates an access token $AT_x^{cj} = H_1\left(\mathbb{F}_{cj}\right)^{\delta_{cj}}$
2. Upon receiving the access token AT_x^{cj}, the user performs the following steps:
3. Compute $T_x^{cj\prime} = B_x^{cj} - H_2(e(AT_x^{cj}, A_x^{cj})$
4. **function** (DecryptNode (CT, σ, SK_u, x))
5. **if** x is an attribute leaf node **then**
6. **if** $att_i \in S$ **then**
7. $F_x{}^{att} = e\left(C_x'.C_x, L\right).e(C,\ D_i)$
$= e\left(H_1(att_i)^{-s}.h^{q_x^1}, g^r\right).e\left(H_1(att_i)^r, g^s\right)$
$= e(g,\ g)^{r\beta q_x^1}$
8. **Else** return $\perp$
9. **End if**
10. **Else**
11. **For** all nodes x (leaf and nonleaf node) related to a contextual token T_x^{cj} **do**
12. $F_x^{cj} = \left(e\left(h.C_x', L\right).e(C, D_i)\right)^{T_x^{cj\prime}}$
$= \left(e(g,g)^{r\beta}.e\left(H_1(att_i)^{-s}, g^r\right).e(H_1(att_i)^r, g^s\right)^{s_x^{cj}}$
$= e(g,\ g)^{r\beta s_x^{cj}}$
13. **End for**
14. **End if**
15. **For** any node x **do**
16. **If** x is related to contextual tokens, **then**
17. $F_x = F_x^{att}.\prod_{j=1}^{N} F_x^{cj}$
$= e(g,\ g)^{r\beta q_x^1}.\prod_{j=1}^{N} e(g,\ g)^{r\beta s_x^{cj}}$
$= e(g,\ g)^{r\beta(q_x^1+\sum_{j=1}^{N} s_x^{cj})}$
$= e(g,\ g)^{r\beta q_x^0}$
18. **Else**
19. $F_x = F_x^{att} = e(g,\ g)^{r\beta q_x^1} = e(g,\ g)^{r\beta q_x^0}$
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, σ , SK_u, z))
25. **End for**
26. Let S_x be an arbitrary k_x-sized set of child nodes of x such that $F_z \neq \perp$
27. **If** S_x exists **then**
28. $F_x{}^{att} = \prod_{z\in S_x} F_z^{\Delta_{i,S_x'(0)}}$
$= \prod_{z\in S_x} (e(g,g)^{r\beta q_z(0)})^{\Delta_{i,S_x'(0)}}$
$= \prod_{z\in S_x} e(g,g)^{r\beta q_x(i)\Delta_{i,S_x'(0)}}$
$= e(g,g)^{r\beta q_x{}^1}$
where i = index(z) and S_x' = {index(z) : $z \in S_x$ }
29. Return $F_x{}^{att}$
30. **Else**
31. Return $F_x{}^{att} = \perp$
32. **End if**
33. **End if**
34. **End function**
35. **If** x is a root node **then**
36. A = DecryptNode (CT, σ , SK_u, R))
$= e(g,g)^{r\beta s}$
37. **End if**
38. The decryption is performed as follows:
39. $K_s' = \frac{C.A}{e(g^s, g^{\alpha}.h^r)}$
40. **If** $e(\sigma, g) = e(H_1\left(K_s'\right), g^{\gamma_{GW}})$ **then**
41. K_s' is valid. **End if**

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.

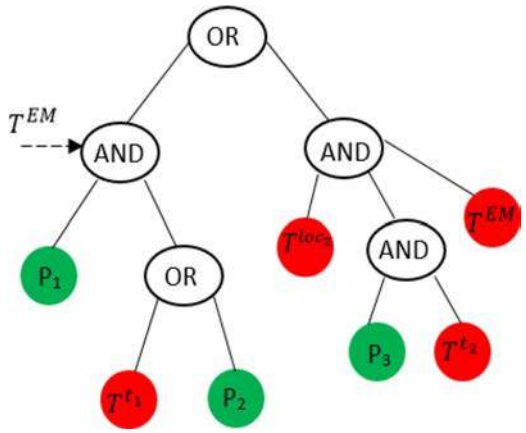

Fig. 7. Access structure in emergency situations.

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^t 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

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

[1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[2] P. Bellavista, G. Cardone, A. Corradi, and L. Foschini, "Convergence of MANET and WSN in IoT urban scenarios," *IEEE Sensors J.*, vol. 13, no. 10, pp. 3558–3567, Oct. 2013.

[3] Y. Xue, J. Hong, W. Li, K. Xue, and P. Hong, "LABAC: A location-aware attribute-based access control scheme for cloud storage," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Washington, DC, USA, 2016, pp. 1–6.

[4] J. Hong *et al.*, "TAFC: Time and attribute factors combined access control for time-sensitive data in public cloud," *IEEE Trans. Services Comput.*, to be published.

[5] H. Ghafghazi, A. Elmougy, H. T. Mouftah, and C. Adams, "Location-aware authorization scheme for emergency response," *IEEE Access*, vol. 4, pp. 4590–4608, 2016.

[6] Y. Yang, X. Liu, and R. H. Deng, "Lightweight break-glass access control system for healthcare Internet-of-Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3610–3617, Aug. 2018.

[7] A. Arfaoui, A. Kribeche, O. R. M. Boudia, A. Ben Letaifa, S. M. Senouci, and M. Hamdi, "Context-aware authorization and anonymous authentication in wireless body area networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kansas City, MO, USA, 2018, pp. 1–7.

[8] Z. Wang, H. Ding, J. Han, and J. Zhao, "Secure and efficient control transfer for IoT devices," *Int. J. Distrib. Sensor Netw.*, vol. 9, no. 11, Jan. 2013.

[9] D. E. Kouicem, B. Abdelmadjid, and L. Hicham, "Distributed fine-grained secure control of smart actuators in Internet of Things," in *Proc. IEEE Int. Symp. Parallel Distrib. Process. Appl. IEEE Int. Conf. Ubiquitous Comput. Commun. (ISPA/IUCC)*, Guangzhou, China, 2017, pp. 653–660.

[10] M. Wazid, A. K. Das, N. Kumar, M. Conti, and A. V. Vasilakos, "A novel authentication and key agreement scheme for implantable medical devices deployment," *IEEE J. Biomed. Health Inform.*, vol. 22, no. 4, pp. 1299–1309, Jul. 2018.

[11] M. Wazid, A. K. Das, V. Odelu, N. Kumar, M. Conti, and M. Jo, "Design of secure user authenticated key management protocol for generic IoT networks," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 269–282, Feb. 2018.

[12] Y. Ashibani, D. Kauling, and Q. H. Mahmoud, "A context-aware authentication framework for smart homes," in *Proc. IEEE 30th Can. Conf. Elect. Comput. Eng. (CCECE)*, Windsor, ON, Canada, 2017, pp. 1–5.

[13] S. Gusmeroli, S. Piccione, and D. Rotondi, "A capability-based security approach to manage access control in the Internet of Things," *Math. Comput. Modell.*, vol. 58, nos. 5–6, pp. 1189–1205, Sep. 2013.

[14] P. N. Mahalle, B. Anggorojati, N. R. Prasad, and R. Prasad, "Identity establishment and capability based access control (IECAC) scheme for Internet of Things," in *Proc. 15th Int. Symp. Wireless Pers. Multimedia Commun.*, Taipei, Taiwan, 2012, pp. 187–191.

[15] J. L. Hernández-Ramos, M. P. Pawlowski, A. J. Jara, A. F. Skarmeta, and L. Ladid, "Towards a lightweight authentication and authorization framework for smart objects," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 4, pp. 690–702, Apr. 2015.

[16] D. Hussein, E. Bertin, and V. Frey, "A community-driven access control approach in distributed IoT environments," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 146–153, Mar. 2017.

[17] C. Hu, H. Li, Y. Huo, T. Xiang, and X. Liao, "Secure and efficient data communication protocol for wireless body area networks," *IEEE Trans. Multi-Scale Comput. Syst.*, vol. 2, no. 2, pp. 94–107, Apr.–Jun. 2016.

[18] N. Oualha and K. T. Nguyen, "Lightweight attribute-based encryption for the Internet of Things," in *Proc. 25th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Waikoloa, HI, USA, 2016, pp. 1–6.

[19] L. Touati, Y. Challal, and A. Bouabdallah, "C-CP-ABE: Cooperative ciphertext policy attribute-based encryption for the Internet of Things," in *Proc. Int. Conf. Adv. Netw. Distrib. Syst. Appl.*, Béjaïa, Algeria, 2014, pp. 64–69.

[20] M. Jahan, S. Seneviratne, B. Chu, A. Seneviratne, and S. Jha, "Privacy preserving data access scheme for IoT devices," in *Proc. IEEE 16th Int. Symp. Netw. Comput. Appl. (NCA)*, Cambridge, MA, USA, 2017, pp. 1–10.

[21] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Security Privacy (SP)*, Berkeley, CA, USA, 2007, pp. 321–334.

[22] M. Wazid, A. K. Das, V. Odelu, N. Kumar, and W. Susilo, "Secure remote user authenticated key establishment protocol for smart home environment," *IEEE Trans. Dependable Secure Comput.*, to be published.

[23] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. Adv. Cryptol. (CRYPTO)*, 2001, pp. 213–229.

[24] S. Wang, K. Liang, J. K. Liu, J. Chen, J. Yu, and W. Xie, "Attribute-based data sharing scheme revisited in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 8, pp. 1661–1673, Aug. 2016.

[25] J. T. Schwartz, "Fast probability algorithms for verification of polynomial identities," *J. ACM*, vol. 27, no. 4, pp. 701–717, 1980.

[26] F. Li, B. Liu, and J. Hong, "An efficient signcryption for data access control in cloud computing," *Computing*, vol. 99, no. 5, pp. 465–479, 2017

[27] A. Zhang, L. Wang, X. Ye, and X. Lin, "Light-weight and robust security-aware D2D-assist data transmission protocol for mobile-health systems," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 3, pp. 662–675, Mar. 2017.

[28] *ITEA3-PARFAIT*. [Online]. Avalable: https://itea3.org/project/parfait.html

[29] A. Almehmadi and K. El-Khatib, "On the possibility of insider threat prevention using intent-based access control (IBAC)," *IEEE Syst. J.*, vol. 11, no. 2, pp. 373–384, Jun. 2017.

[30] J. Wu, S. Guo, H. Huang, W. Liu, and Y. Xiang, "Information and communications technologies for sustainable development goals: State-of-the-art, needs and perspectives," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2389–2406, 3rd Quart., 2018.

[31] S. Chen *et al.*, "Internet of Things based smart grids supported by intelligent edge computing," *IEEE Access*, vol. 7, pp. 74089–74102, 2019.

[32] J. Wu, S. Guo, J. Li, and D. Zeng, "Big data meet green challenges: Big data toward green applications," *IEEE Syst. J.*, vol. 10, no. 3, pp. 888–900, Sep. 2016.

[33] J. Wu, I. Bisio, C. Gniady, E. Hossain, M. Valla, and H. Li, "Context-aware networking and communications: Part 1 [guest editorial]," *IEEE Commun. Mag.*, vol. 52, no. 6, pp. 14–15, Jun. 2014.

[34] R. Atat, L. Liu, J. Wu, G. Li, C. Ye, and Y. Yang, "Big data meet cyber-physical systems: A panoramic survey," *IEEE Access*, vol. 6, pp. 73603–73636, 2018.

[35] J. Liu, Q. Li, R. Yan, and R. Sun, "Efficient authenticated key exchange protocols for wireless body area networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2015, p. 188, Dec. 2015.