Context-Aware Adaptive Authentication and Authorization in Internet of Things

Amel Arfaoui, Soumaya Cherkaoui, Ali Kribèche, Sidi-Mohammed Senouci, Mohamed Hamdi

Context-Aware Adaptive Authentication And Authorization in Internet of Things

Amel Arfaoui*ǂ, Soumaya Cherkaoui†, Ali Kribecheǂ, Sidi Mohammed Senouciǂ, Mohamed Hamdi*
*Digital Security Unit, SupCom University of Carthage, Tunisia
ǂDRIVE EA1859, Univ. Bourgogne Franche Comté, France
†INTERLAB Research Laboratory, University of Sherbrooke, Canada
{amel.arfaoui, Sidi-Mohammed.Senouci, Ali.Kribeche01}@u-bourgogne.fr, soumaya.cherkaoui@usherbrooke.ca, mmh@supcom.tn

***Abstract*—The rapid technological advancements in wireless communications, ubiquitous sensing and mobile networking have paved the way for the emergence of the Internet of Things (IoT) era, where "anything" can be connected "anywhere" at "anytime". However, the flourish of IoT still faces various security and privacy preserving challenges that need to be addressed. In such pervasive and heterogeneous environment where the context conditions dynamically and frequently change, efficient and context-aware mechanisms are required to meet the users' changing needs. Therefore, it seems crucial to design an adaptive access control scheme in order to remotely control smart things while considering the dynamic context changes. In this paper, we propose a Context-Aware Attribute-Based Access Control (CAABAC) approach that incorporates the contextual information with the Ciphertext-Policy Attribute-based Encryption (CP-ABE) to ensure data security and provide an adaptive contextual privacy. From a security perspective, the proposed scheme satisfies the security requirements such as confidentiality, context-aware privacy, and resilience against key escrow problem. Performance analysis proves the efficiency and the effectiveness of the proposed scheme compared to benchmark schemes in terms of storage, communication and computational cost.**

Keywords— IoT; Context-aware security; Adaptive, Authorization; Authentication; Attribute Based Encryption.

I. INTRODUCTION

The Internet of Things (IoT) is a revolutionary communication paradigm which consists to connect a multitude of digital devices to the Internet [1]. Today, IoT is beginning to shape the future of many applications where users can remotely control smart things using their smartphones. However, the open nature of wireless communication imposes diverse privacy preserving and security concerns such as eavesdropping, message interception, and data modification. Therefore, the transmitted data between the communicated parties should be handled and analyzed only by authorized users in order to ensure accurate monitoring. Furthermore, the dynamic and heterogeneous structure of IoT induces more challenges for security solutions' design. Indeed, authentication and authorization should be adapted to context changes (such as time, data consumers' roles, location, data type, emergency or normal situation…) in order to make the right decision at the right time by the right party.

Several works and researches are focusing on designing authentication and access control schemes in IoT to deal with security and privacy preserving challenges. In [2], the authors addressed the problem of remote secure control of smart actuators. For this purpose, they proposed a distributed lightweight fine-grained access control based on Attribute-Based Encryption scheme and one-way hash chain for authentication. In [3][4], Capability-Based Access Control (CapBAC) using authorization tokens was introduced as a realistic mechanism to be implemented in IoT. The approach is based on the assignment of authorization decisions to a central entity which delivers privileges to be adopted at the end device. However, the use of a central entity validating users' access rights introduces a single point of failure and prevents end-to-end security. Distributed CapBAC [5][6] tackles these issues by having the authorization executed by the IoT devices themselves. However, IoT objects are often resource-constrained and may be easily compromised. Distributed CapBAC is, therefore, ill-equipped to address access control in untrustworthy IoT environments. In addition, Attribute Based Cryptography (ABC) is considered as a promising tool that can be exploited to provide adaptive access control. In such scheme, each user is associated with a set of attributes and data is encrypted on the basis of an access structure. Only data consumers whose attributes satisfy the access policy can decrypt the ciphertext. In [7], the authors developed a CP-ABE scheme to secure the communications between sensor nodes and the data sink/data consumers. However, the proposed scheme suffers from key escrow problem and high computational cost. Hence, this scheme is inappropriate for resource-constrained devices that cannot support the heavy overhead of the CP-ABE. In [8] [9], the authors proposed fine-grained access control schemes while combining the Ciphertext-Policy Attribute-based Encryption (CP-ABE) with time/location factors.

All the aforementioned works do not involve the contextual information for the authentication and authorization and even if the context is considered, only time or location are used to define the context. Therefore, it is necessary to conceive an effective scheme, which will grant data access only to authorized users under a predefined context. A trivial solution to combine a user's role and contextual information into access policies is to consider the contextual parameters as a set of normal attributes [10]. However, the main difference between a user's dynamic context and her attributes is that attributes are defined on the basis of her identity which will be maintained for a long period while the contextual information is a dynamic condition, which is frequently changing over time. If a

contextual parameter such as location/time is handled as a user attribute, her attribute set will change permanently anywhere at any time. This solution is obviously impractical in real scenarios and introduces heavy computation and communication overhead [8].

In this paper, we propose a context-aware authentication and authorization scheme to adaptively provide secure communication between data consumers and smart things according to the current context. We introduce a novel Context-Aware Attribute-Based Access Control (CAABAC) scheme that combines the contextual information and attributes to ensure an adaptive context-aware privacy. In the proposed scheme, we define fine-grained privileges while exploiting the features of the CP-ABE scheme. In addition, we introduce a contextual token mechanism which is related to the contextual information (location, time, emergency situation, normal situation, data type…), where the corresponding secret should be revealed under a predefined context to generate an access token. To decrypt a ciphertext, the data consumer has not only to possess the appropriate attribute set but also to have an access token under a specific situation.

The major contributions can be summarized as follows:

- A novel context-aware authentication and authorization approach that provides dynamic and secure control of smart things based on the contextual information,
- The contextual information is combined with attributes in access policies using contextual tokens in order to alleviate the burdensome revocation when a user's context changes,
- An enhanced key issuing protocol is presented to resolve the key escrow problem of CP-ABE. In fact, users' private keys are generated based on the cooperation between Key Generator Center (KGC) and Attribute Authority (AA), so that any authority cannot create the whole users secret keys.

The remainder of the paper is organized as follows. Section II presents a mathematical background. The system model is presented in Section III. The new CAABAC scheme for access control is described in Section IV, followed by the performance analysis in Section V. Finally, a conclusion is drawn in Section VI.

II. PRELIMINARIES

In this section, we present some preliminary knowledge regarding a background on Bilinear Pairings and cryptographic primitives exploited in this paper.

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: $\mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ satisfying these properties:

- 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 $e(aP, bQ) = e(P, Q)^{ab}$.
- Non-degeneracy: $\exists P, Q \in \mathbb{G}1$ where $e(P, Q) \neq 1\mathbb{G}_2$.
- 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:

- *DBDH (*Decision *Bilinear Diffie-Hellman) 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. Definitions

1) Ciphertext-Policy Attribute-Based Encryption (CP-ABE): This scheme includes the following four algorithms

- 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.
- KeyGen (PK, MK, S): The KGC takes the master secret key MK, the attribute set S of the user, and the public key of the system PK as inputs. It generates the private key SK_U.
- 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.
- Decryption (CT, SK_U): The receiver takes as input the ciphertext CT, and her decryption key SK_U. The algorithm outputs a message M or a reject symbol $\perp$.

2) Access policy structures

An access structure $\mathcal{T}$ consists of several nodes of a policy tree and several contextual tokens (presented in Fig 1). A leaf node represents a set of attributes (att_0, ⋯, att_3), and each non-leaf node defines a threshold gate ("AND", "OR", or other threshold gates). Each non-leaf 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 [12]. In the structure $\mathcal{T}$, $T_x^{c_j}$ is related to the contextual parameter c_j that may be time, location, situation sensitivity, etc.

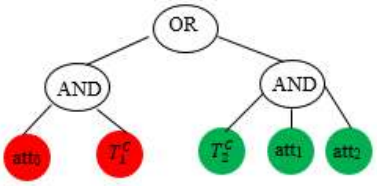

Fig 1. An example of access structure

III. SYSTEM AND SECURITY MODELS

In this section, we first present the different entities of the system model. Then, we describe the security model.

A. System Model

We consider an IoT remote control system presented in Fig 2. It mainly consists of the following entities: Key Generator Center (KGC), Attribute Authority(AA), Context Manager (CM), IoT gateway, smart things, and data consumers. The functions of each entity are as follows:

- The Key Generator Center (KGC) and the Attribute Authority (AA) are semi-trusted entities. They are responsible for

system initialization, public parameters generation and users' secret keys assignment.

- Context Manager (CM) is responsible for the control of the dynamic context changes. It performs operations for the data consumer such as verifying the user's context and generating an access token to enable her decrypting the ciphertext.
- IoT gateway is deployed as a powerful node that cooperates with the IoT device in order to implement the CP-ABE scheme. In addition, it is responsible for the management of remote access control to smart things.
- 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.
- Data Consumers refer to the users who aim to communicate with IoT devices and perform remote actions on them. To decrypt a message, data consumers need not only to have the set of attributes that satisfy the access structure but also to verify the contextual information.

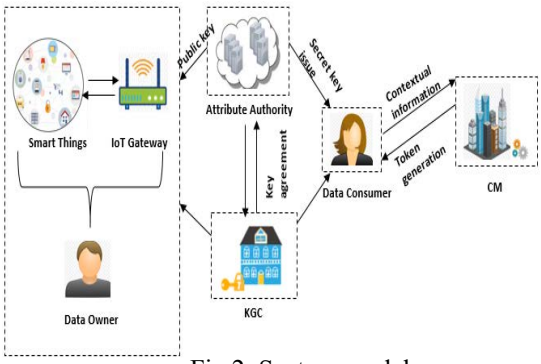

Fig 2. System model

B. Security Model

In the proposed scheme, we consider that ***AA*** and ***KGC*** are semi-trusted: honest-but-curious. That means they will honestly follow the protocol, but they will try to disclose as much secret information as possible. ***The CM*** is assumed fully trusted. ***The IoT gateway*** presents the data owner. It is assumed to be fully trusted given that it cooperates with IoT devices to encrypt data. In this work, we assume that ***Smart things*** are available and are neither compromised nor spoofed. ***Data consumers*** are assumed dishonest. They try to decrypt data even they are unauthorized.

IV. Efficient Context-Aware Attribute-Based Access Control (CAABAC) Scheme

In this section, we first describe the basic notations used in this paper. Then, we present the main features of the proposed scheme that ensures an adaptive access control according to the dynamic context changes.

A. Notations

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 signing 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
AT	Access token for a given context
TK	Authentication token
$\mathbb{F}_{c_j}$	Unified format of the contextual parameter c_j

Table 1. Variables and their descriptions

B. The proposed CAABAC scheme

The main idea of the proposed scheme is to provide secure and adaptive remote control of smart things. For this purpose, we exploit the fine-granularity of CP-ABE and introduce a contextual token concept to ensure dynamic access control while considering the contextual information. Especially, we integrate contextual tokens into the access structure to restrict access privileges by the contextual information. Successful decryption requires not only proper attribute set but also a suitable access token. In fact, a data consumer has to interact with the CM that verifies the context requirements and generates an access token. The proposed scheme is composed of four phases: System initialization, Key Generation, Encryption, as well as Decryption and communication that are presented as follows.

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 context manager defines the secret keys of the contextual parameters.

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 $i \in \mathbb{Z}q$ 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 $\alpha_1, \beta \in \mathbb{Z}_q$, 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 $PK_{KGC} = e(g,g)^{\alpha_2}$ and the secret key $MK_{KGC} = \{\alpha_2\}$
6. The KGC selects a signing 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 $PK_{c_j} = \{\mathbb{F}_{c_j}, \gamma_{c_j} = g^{\delta_{c_j}}\}$

2) Key Generation

This phase is executed by both AA and KGC to generate a secret key for a user u. At first, AA selects a random unique

number $r \in \mathbb{Z}_q$ for the user. Then, AA and KGC execute a secure two-party computation (2PC) protocol, where AA inputs $MK_A = \{\alpha_1, \beta\}$ and KGC inputs $MK_{KGC} = \{\alpha_2\}$. As a result, KGC gets X = $(\alpha_1+\alpha_2)$. β mod q [11]. After the 2PC protocol, AA and KGC perform the following key commitment algorithm:

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 $SK_{KGC} = D = V_3^{1/\tau_2} = g^{\alpha}h^r$ 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: $SK_{AA,u} = \{D_i = H_1(att_i)^r, \forall\ att_i \in S, L = g^r\}$
7. The user determines his personalized secret key as $SK_u = \{D = g^{\alpha}h^r, L = g^r,\ D_i = H_1(att_i)^r, \forall\ att_i \in S\}$

3) Encryption

In this phase, a smart thing defines a challenge *M* to execute an instruction *I* by a user and cooperates with the IoT gateway to encrypt it based on an access tree $\mathcal{T}$ for a given contextual information. At First, the IoT device encrypts the message *M* with K_s by using symmetric encryption method, where K_s is a pre-shared secret key with the IoT gateway. Then, the IoT gateway proceeds as follows to encrypt K_s using CP-ABE.

Algorithm 3 Encryption

Input: An access tree $\mathcal{T}$, public parameters *PK*, contextual parameters 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 $d_x = k_x - 1$
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** 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 Z_q to completely define q_x
7. **End for**
8. **if** x is a leaf node related to a 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. Let $\mathcal{X}$ be the set of leaf nodes in $\mathcal{T}$. The ciphertext *CT* is constructed based on the access tree $\mathcal{T}$ as follows:

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

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

4) Decryption and communication

In this phase, a data consumer who aims to communicate with an IoT device, sends a request message to the IoT gateway. Upon receiving the request, the IoT gateway generates a random nonce $r \in \{0,1\}^*$ and sends $< \text{CT}, \sigma, r,\ AES(K_s, \text{M}) >$ to the data consumer. The data consumer decrypts the ciphertext based on her attribute set and the contextual information according to **Algorithm 4** in order to obtain the symmetric key K_s. Then, she covers the plaintext *M'* and sends 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. Upon receiving M_2, the IoT gateway decrypts it with its private key and verifies if $H_1' = H_1$. If the condition holds, 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 executed by the IoT device. We notice that M_3 is encrypted by *SK* to be sent to the data consumer and SK_1 (a symmetric pre-shared key between the gateway and the IoT device) to be sent to the IoT device. When the data consumer sends a request $H_1' = H_1(TK||T_e||ID_i)$to the IoT device, it verifies if $H_1' = H_1$. If succeeds the IoT device performs the instruction, *I,* sent remotely by the data consumer.

Algorithm 4 Decryption

Input: the ciphertext *CT*, the signature σ, the public parameters *PK*, the set of attributes *S,* the contextual token $T_x^{c_j}$, the secret key SK_U
Output: the plaintext K_s'

1. The context manager 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. $F_{x,T_x^{c_j}} = \left(e(h.C_x', L).e(C, D_i)\right)^{T_x^{c_j\prime}}$
$= \left(e(g,g)^{r\beta}.e(H_1(att_i)^{-s}, g^r).e(H_1(att_i)^r, g^s)\right)^{T_x^{c_j\prime}}$
$= e(g,g)^{r\beta T_x^{c_j\prime}}$
7. **Else if** x is an attribute leaf node **then**
8. **if** $att_i \in S$ **then**
9. $F_x = e(C_x'.C_x, L).e(C, D_i)$
$= e\left(H_1(att_i)^{-s}.h^{q_x(0)}, g^r\right).e(H_1(att_i)^r, g^s)$
$= e(g,g)^{r\beta q_x(0)}$
10. **Else** return $\perp$
11. **End if**
12. **Else**
13. **For** each child z of x **do**
14. $F_z = \text{DecryptNode (CT}, \sigma, SK_u, \text{z}))$
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 = index(z) and S_x' = {index(z) : $z \in S_x$ }
19. Return F_x
20. **Else**
21. Return $F_x = \perp$
22. **End if**
23. **End if**
24. **End function**

25. **If** x is a root node **then**
26. $A = \text{DecryptNode}(\text{CT}, \sigma, SK_u, \text{R})$
$= e(g,g)^{r\beta s}$
27. **End if**
28. The decryption is performed as follows:
29. $K'_s = \frac{\hat{C}.A}{e(g^s, g^{\alpha}.h^r)}$
30. **If** $e(\sigma, g) = e(H_1(K'_s), g^{\gamma_{GW}})$ **then**
31. K'_s is valid
32. **End if**

V. CAABAC Performance and Security Analysis

In this section, we evaluate the effectiveness of the proposed scheme through a security analysis. Then, a comparative study of benchmarking approaches will be presented to assess the communication, the storage, and the computation cost of the CAABAC scheme.

A. Security Analysis

- *Data confidentiality*: Data confidentiality of the proposed CAABAC invokes the security of both CP-ABE [12] and identity-based encryption (IBE) [13] algorithms. It has been proved that these algorithms are secure under the DBDH assumption. On one hand, the contextual information is embedded into the access structure as a set of special attributes, thus this integration does not affect the structure of the CP-ABE scheme. Hence, the property of CP-ABE confidentiality is conserved. On the other hand, the contextual tokens are generated based on IBE. Therefore, the security of access tokens can be demonstrated in random oracle model. In addition, the ciphertext cannot be decrypted without a valid access token. Hence, the proposed scheme ensures confidentiality.
- *Context-aware privacy*: In our construction, the encryption algorithm involves the contextual information to determine who can access what and under which context. In fact, a data consumer can decrypt the ciphertext only if she satisfies the context requirements and she has a valid token to access data.
- *Mutual authentication*: During the authentication and authorization process, the authentication between the IoT gateway and the data consumer is performed using a challenge-response technique. Once the first authentication is achieved, each smart thing authenticates the data consumer using an authentication token.
- *Resilience against escrow problem*: In the proposed scheme, the private keys of users are generated based on the cooperation between KGC and AA. Thus, any authority cannot reveal the whole secret key of the user.
- *Replay attack*: To resist the replay attack, the IoT gateway sends a random nonce r with the ciphertext to the user. The response message $H_1^{'} = H_1(M^{'}||r)$ cannot be used by another user to get an authentication token. In addition, the expiration time T_e added to the authentication token TK ensures the validity and freshness of the communicated messages.

B. Performance Analysis

In this subsection, we evaluate performance characteristics of CAABAC scheme through quantitative analysis. The proposed scheme is compared with those of H-CLSC [10], CP-ABE [7] and PPDAS [14] in terms of storage, communication, and computation cost. We notice that the compared schemes apply different methods to design the access control algorithm. 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 [7], 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. As [10], the size of an element in group $\mathbb{G}_1$ can be compressed to 65 bytes.

1) Storage overhead

The storage overhead is related to the size of users' secret keys. In the CAABAC scheme, the data consumer needs to store $\{SK_u = \{D = g^{\alpha}h^r, L = g^r, D_i = H_1(att_i)^r, \forall att_i \in S\}\}$, whose size is (|att|+2) $|\mathbb{G}_1|$, where |att| is the cardinality of the attribute set. As shown in Table 2, the user in the proposed scheme requires less storage overhead than other schemes [14] [7] using CP-ABE algorithm.

Scheme	storage overhead
CP-ABE [7]	(2*|att|+1) $\|\mathbb{G}_1\|$
H-CLSC [10]	$\|\mathbb{G}_1\|$=65 bytes
PPDAS[14]	(3*|att|+13) $\|\mathbb{G}_1\|$+2$\|Z_q^*\|$
CABAAC	(|att|+2) $\|\mathbb{G}_1\|$

Table 2. Storage overhead comparisons

2) Communication overhead

The ciphertext is stored in the IoT gateway and transmitted to data consumers when requested. In this analysis we consider the exchanged messages between the data consumer, the IoT gateway and the smart thing. In fact, the IoT gateway has to send to the data consumer $< CT, \sigma, r, ID_i, TK, T_e, AES(K_s, M) >$ whose size is $|T| + |\widetilde{C}| + |C| + |C'_x| + |T_x^{c_j}| + |C_x| + |\sigma| + |r| + |ID_i| + |TK| + |T_e| + |AES(K_s, M)|$. We assume as [11] that $|ID_i|$, $|T_e|$, $|T|$ have 1-byte, 1-byte, 4-bytes, respectively. In addition, the IoT gateway has to send to the smart thing the message $< ID_i, TK, T_e >$ whose size is $|ID_i| + |TK| + |T_e|$. The smart thing has only to encrypt the challenge M with a symmetric key and send it to the IoT gateway, so, the message size is $|AES(K_s, M)|$. Compared to H-CLSC [10], CAABAC has higher communication overhead. However, in H-CLSC this cost linearly increases with the number of users (n) but in our scheme it is independent of the number of users.

Scheme	IoT gateway (bytes)	Smart thing (bytes)
CP-ABE [7]	5|p|+ 24=236.5	10|p| + 76=501
H-CLSC[10]	2$\|\mathbb{G}_1\|$+n $\|Z_q^*\|$+|M|+w=180 (1 user)	-
PPDAS[14]	27|p|+31=1178.5	|p|+1=43.5
CABAAC	6|p|+ 4$\|Z_q^*\|$+24=359	$\|AES(K_s, M)\|$=16

Table 3. Communication overhead comparisons

3) Computation Cost

In this subsection, we assess the computation overhead of the proposed CAABAC scheme compared to benchmarking schemes. As the operations on pairing, exponentiation and multiplication mainly affect the computational overhead, we only consider them. We denote T_E the time consumed for one exponentiation operation, T_M the time consumed for one scalar multiplication in $\mathbb{G}_1$, and T_P the time for one pairing operation.

In CAABAC scheme, the encryption process in the IoT gateway requires seven Tate pairing operations. The computational cost of the different comparative schemes is presented in Table 4. As in [15], to evaluate the running time of the operations, the algorithms are implemented on an Intel PXA270 processor at 624 MHz installed on the Linux personal digital assistant. The running time of the different operation are T_E = 53.85ms, T_M = 30.67ms, and T_P = 96.20ms, respectively.

Scheme	IoT gateway (ms)	Smart thing (ms)
CP-ABE [7]	$5T_P$=481	$10T_P$=962
H-CLSC[10]	$3T_P+6T_M$=472.62	-
PPDAS[14]	$11T_P + 25T_E+4T_M$ =2520.38	$1T_E$=53.85
CAABAC	$7T_p$=673.4	-

Table 4. Computation cost comparisons

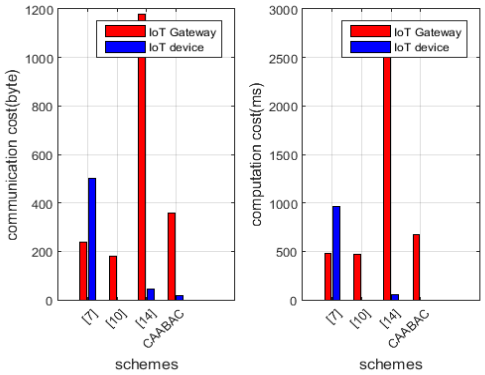

Fig 3. Communication and Computational cost comparisons

As shown in Fig 3, the proposed scheme is more efficient than the other schemes using CP-ABE algorithm. However, it has more computational cost compared to the H-CLSC scheme. But, in H-CLSC [10], 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. Therefore, the computation cost will be linear to the context changing and the number of users. Nevertheless, in the proposed model only a unique token will be generated to define each context.

VI. Conclusion

In this paper, we have proposed a novel Context-Aware Attribute-Based Access Control (CAABAC) scheme to provide dynamic and context-aware access control. The proposed approach incorporates the contextual information as a set of special attributes in the CP-ABE scheme. From a security perspective, the proposed scheme meets the different security requirements and solves the key escrow problem of CP-ABE algorithm. The performance analysis has proven that CAABAC outperforms the existing access control schemes using CP-ABE algorithm.

Acknowledgment

This work is achieved as part of the European project ITEA PARFAIT [16], 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," in *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22-32, Feb. 2014.

[2] D. E. Kouicem, B. Abdelmadjid and L. Hicham, "Distributed Fine-Grained Secure Control of Smart Actuators in Internet of Things," *(ISPA/IUCC)*, Guangzhou, 2017, pp. 653-660.

[3] S. Gusmeroli, S. Piccione, and D. Rotondi, "A capability-based security approach to manage access control in the internet of things," Mathematical and Computer Modelling, vol. 58, no. 5-6, pp. 1189–1205, September 2013.

[4] P. N. Mahalle &al. "Identity Establishment and Capability based Access Control (iecac) scheme for Internet of Things," in Proc. *The 15th International Symposium on Wireless Personal Multimedia Communications*, Taipei, 2012, pp. 187-191.

[5] M. P. Pawlowski et al., "Towards a Lightweight Authentication and Authorization Framework for Smart Objects ," in *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 4, pp. 690-702, April 2015.

[6] 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, March 2017.

[7] Chunqiang Hu, Hongjuan Li, Yan Huo, Tao Xiang, , and Xiaofeng Liao, " Secure and Efficient data communication protocol for Wireless Body Area Networks", IEEE. Trans. On Multi-scale Computing, vol.2, no.2, June.2016

[8] J. Hong, K. Xue, W. Li, and Y. Xue, "LABAC: A Location-aware Attribute-based Access Control Scheme for Cloud Storage," in Proc. *IEEE Global Communications Conference (GLOBECOM)*, Washington, DC, 2016, pp. 1-6.

[9] J. Hong, K. Xue, W. Li, and Y. Xue, "TAFC: Time and attribute factors combined access control on time-sensitive data in public cloud," in Proc. *IEEE Global Communications Conference (GLOBECOM)*, San Diego, CA, 2015, pp. 1-6.

[10] 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," *IEEE International Conference on Communications (ICC)*, Kansas City, MO, 2018, pp. 1-7.

[11] S. Wang, K. Liang, J. K. Liu, J. Chen, J. Yu and W. Xie, "Attribute-Based Data Sharing Scheme Revisited in Cloud Computing," in *IEEE Trans. Info. Forensics and Security*, vol. 11, no. 8, pp. 1661-1673, Aug. 2016.

[12] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in Proceedings of the *IEEE Symposium on Security and Privacy (SP '07)*, Berkeley, CA, 2007, pp. 321-334.

[13] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in Advances in Cryptology (CRYPTO2001), pp. 213–229, Springer, 2001

[14] M. Jahan, S. Seneviratne, B. Chu, A. Seneviratne and S. Jha, "Privacy preserving data access scheme for IoT devices," *IEEE 16th International Symposium on Network Computing and Applications (NCA)*, Cambridge, MA, 2017, pp. 1-10.

[15] Zhang, A.; Wang, L.; Ye, X.; Lin, X. Light-Weight and Robust SecurityAware D2D-Assist Data Transmission Protocol for Mobile-Health Systems. IEEE Trans. Inf. Forensics Secur. 2017, 12, 662–675.

[16] ITEA3-PARFAIT.https://itea3.org/project/parfait.html.