Future Generation Computer Systems 131 (2022) 75–90

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

MLS-ABAC: Efficient Multi-Level Security Attribute-Based Access Control scheme

Seyed Farhad Aghili [a,*], Mahdi Sedaghat [a], Dave Singelée [a], Maanak Gupta [b]

[a] imec-COSIC, KU Leuven, Leuven, Belgium
[b] Tennessee Technological University, USA

ARTICLE INFO

Article history:
Received 25 August 2021
Received in revised form 10 December 2021
Accepted 5 January 2022
Available online 31 January 2022

Keywords:
Access control
Attribute-Based Encryption
Multi-Level Security
Authenticated encryption
Dynamic attributes

ABSTRACT

Realizing access control to sensitive data offloaded to a Cloud is challenging in the Internet of Things, where various devices with low computational power and different security levels are interconnected. Despite various solutions, the National Institute of Standards and Technology (NIST)'s Attribute-Based Access Control (ABAC) model is one of the preferred techniques in the literature. In this model, users who satisfy access policies using both static and dynamic attributes are allowed to access the data. However, NIST's ABAC model does not support encryption and therefore does not satisfy data confidentiality. Attribute-Based Encryption (ABE) is a known cryptographic primitive that enables fine-grained access control over encrypted data. However, currently the existing ABE schemes do not meet NIST's ABAC requirements or are not computationally efficient enough for IoT applications. In this paper, we propose a Multi-Level Security ABAC (MLS-ABAC) scheme that satisfies the requirements of NIST's ABAC model. Our construction is efficient and relies on a decryption outsourceable Ciphertext-Policy ABE scheme. Additionally, based on realistic application scenarios, only the authorized data users can decrypt the ciphertext, and check the integrity of the retrieved message. Furthermore, we present both conceptual and formal models for our proposed MLS-ABAC architecture along with performance metrics. The experimental results show that the proposed MLS-ABAC achieves a constant ciphertext size of ∼230 bytes and with encryption and decryption running times of ∼18 and ∼10 ms, respectively, independent of the number of attributes.

1. Introduction

The Internet of Things (IoT) is an evolving paradigm that enables wide-range connectivity among smart devices. With the intelligent nature of IoT networks and the openness of their associated applications, such as the Internet of Medical Things (IoMT), users can conduct real-time interactions any time and anywhere to take advantage of services such as tele-care [1]. This immediate connectivity undoubtedly raises information security and privacy challenges. Controlling user's conduct based on multiple access levels, i.e., multi-level access control mechanism, is critical. For example, only an authorized entity that fulfills the access control criteria for accessing the information in a healthcare network (e.g., roles and resources) can acquire access.

Besides, in the IoT networks, access control schemes should be energy-efficient. It means that the scheme proposed for being used in these systems must be lightweight to not impose high computational costs on resource-constrained elements of these systems. Using lightweight functions instead of heavy-weight functions, such as lightweight hash functions, lightweight authenticated encryption function, etc. makes the access control scheme efficient.

Background: Envision an IoT system where the data owner (e.g., patient) stores his/her sensitive information in either local servers or cloud systems, while the end-user (e.g., employees in the hospital) can read that data with some pre-defined access control rules. In this scenario where the system's access control is static (define once — used multiple times), the system administrator would typically use a classic access control model like Role-Based Access Control (RBAC) [2]. In traditional RBAC-based systems, the roles such as a doctor, nurse, and healthcare provider are assigned to the users, and the permissions of the roles are defined, i.e., to which objects/data in the system access is granted.

Generally, RBAC models cannot meet all the requirements of an IoT environment, such as heterogeneity and real-time. For instance, using personal mobile devices – the concept of Bring-Your-Own-Device (BYOD) – presents new security and privacy threats to the IoT that are not dealt with by RBAC. Luckily, there is an alternative solution called the Attribute-Based Access

* Corresponding author.
E-mail addresses: seyedfarhad.aghili@esat.kuleuven.be (S.F. Aghili), ssedagha@esat.kuleuven.be (M. Sedaghat), dave.singelee@esat.kuleuven.be (D. Singelée), mgupta@tntech.edu (M. Gupta).

https://doi.org/10.1016/j.future.2022.01.003

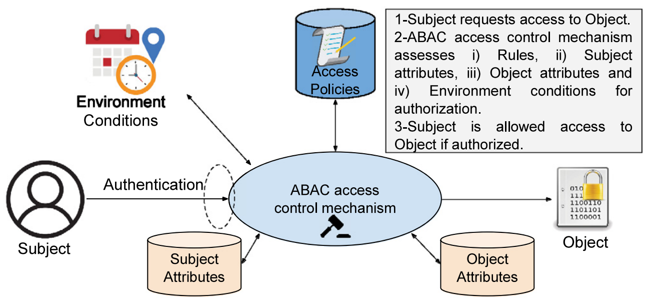

Environment Conditions
Access Policies
1-Subject requests access to Object.
2-ABAC access control mechanism assesses i) Rules, ii) Subject attributes, iii) Object attributes and iv) Environment conditions for authorization.
3-Subject is allowed access to Object if authorized.
Authentication
ABAC access control mechanism
Subject
Subject Attributes
Object Attributes
Object

Fig. 1. Basic ABAC model.

Control (ABAC) model. In ABAC, users can access the data if and only if they have all the associated attributes with the data. The basic ABAC model is based on a set of pre-defined and static attributes, such as "user identity," "user profession," and "user company", through which permissions are given to the users if they present all of the owned attributes. However, in this kind of ABAC system, a person with authorization will be able to access data for a lifetime, and from anywhere, with a mobile device. Some cases can cause privacy and security issues. For instance, assume the authorized user would access the confidential data in a public place, where there are many entities in close proximity, such as humans and espionage devices. There is a high risk that confidential data gets leaked in such a setting due to, for example, shoulder surfing or a camera recording. Alternatively, even worse, an attacker could access the data by stealing the mobile device. The problem here is that (i) there are no dynamic attributes that provide real-time verification and (ii) all the static attributes are stored inside the mobile device. As depicted in Fig. 1, the National Institute of Standards and Technology (NIST)'s ABAC model [3] consists of three kinds of entity attributes: (i) subject attributes, e.g., the role or the job title of the data user; (ii) object attributes, e.g., resource and files created by the data owner; and (iii) environment attributes, e.g., location and time. Using the full potential of NIST's ABAC model, i.e., employing both static and dynamic attributes (e.g., "time" and "location"), could overcome these problems. Dynamic attributes are derived from user behavior, so access is granted to the user who meets static attributes and provides dynamic attributes.[1]

However, we need to go even one step further. Due to the sensitivity of the information (e.g., healthcare records) outsourced to the cloud, this information should be securely stored (i.e., encrypted). Only an authorized user who has the security key should be able to read the information. Thus, integrating ABAC into Attribute-Based Encryption (ABE) provides us a possibility to encrypt the outsourced information and protect these from the cloud system itself. Using ABE, even if a cloud system were honest but curious, it would not be able to read the sensitive information.

Besides, it is not desirable in many applications that all information is accessible to all authorized users. This could cause unauthorized access to users who have sufficient features but only a lower security level. Specifying multi-level security (MLS) for the users and data enables two features: (i) users can access the information that is stored with the same security level as assigned to them, and (ii) at the same time prevents users from getting access to information for which they cannot satisfy the security level. Thus, MLS-based access control should be in place to ensure that access to confidential data is strictly controlled.

[1] This is controversial in practice since some of the dynamic attributes, like biometrics, may be noisy. A fuzzy extraction method can be used to overcome this challenge [1].

Typically, MLS-based access control implicates Mandatory Access Control (MAC) such as the Bell–LaPadula (BLP) model [4] that uses security labels for both subjects and objects. However, there are several problems with MLS, such as "sanitization", "covert channels", and "bypass" that should be considered. Besides confidentiality, also data integrity is an essential issue in IoT applications. In fact, the data should not have been altered and an authorized user who decrypts the information must be sure of its integrity, validity and consistency. Moreover, for schemes where the decryption is outsourced, the correctness of this operation should be checked as well. Employing an authenticated encryption schemes would be an excellent solution to overcome these issues.

Recently, Perazzo et al. in [5] evaluate the performance of the Bethencourt et al.'s representative CP-ABE scheme [6], considering the worst-case scenario on two popular IoT platforms, namely ESP32 [7] and RE-Mote [8]. It is demonstrated that if the CP-ABE scheme in [6] employs up to 10 attributes in ciphertexts and leverage hardware cryptographic acceleration, it can actually be used on IoT applications with resource-constrained devices. Thus, if the ABE scheme can have a constant execution time (e.g., with a better performance than [6] with 10 attributes) for its key generation, encryption, and decryption algorithms over total number of the attributes, it can easily be employed as a building block in ABAC for IoT applications. In the performance analysis section, we will discuss it in detail.

Our Contributions: In particular, we propose a Multi-Level Security Attribute-Based Access Control (MLS-ABAC) scheme with below functionalities.

- *Robust against compromised attacks*: MLS-ABAC ensures that a data user who can present all the necessary static and dynamic attributes can access the information for which they can satisfy the security level.
- *Compatible with IoT networks*: MLS-ABAC is lightweight and can be used in the context of IoT. This efficiency gain primarily comes from our proposed efficient decryption outsourceable CP-ABE construction, which is fast, flexible and lightweight.
- *Data Integrity*: The data integrity of the plaintext in case of outsourced decryption is preserved using authenticated encryption scheme.
- *MLS access control*: By using the attribute token and access token, only the user who satisfies the security level embedded inside the access token and meets the required policies can get access the ciphertext labeled with this security level.
- *Practical scheme*: The effectiveness and efficiency of MLS-ABAC are evaluated using the charm-crypto framework.

Moreover, we formalized our proposed access control model to demonstrate that it can be applied in real-world applications and is compatible with the NIST's distributed ABAC model.

Roadmap: The rest of the paper is organized as follows. The related work concerning the CP-ABE schemes is discussed in Section 2. The high-level overview of the proposed scheme is presented in Section 3. The necessary preliminaries and definitions and the main cryptographical building blocks and their security properties are formally defined in Section 4 along with our proposed decryption outsourceable CP-ABE. The proposed MLS-ABAC construction is defined in Section 5. The security requirements and the intuition behind the security along with the performance analysis are discussed in Section 6. In Section 7, we provide our implementation results and show how the MLS-ABAC construction is compatible with the IoT context. In Section 8, we discuss the introduced MLS-ABAC model in detail and its main applications and use-cases. Finally, we conclude the main contributions of the paper in Section 9.

2. Related work

Although NIST's ABAC model has many deceptive benefits, this model faces various security vulnerabilities, such as attribute forgery and permission counterfeiting. This is especially noteworthy for open environments such as IoT. In what follows, we list the related works in the literature.

2.1. ABE schemes

An excellent solution for this problem is using cryptographic access control. An ABE scheme is a superior cryptographic tool for fine-grained access control of encrypted data, which is utilized in various applications [9]. Goyal et al. in [10], formulated two types of ABE schemes: (i) Key-Policy ABE (KP-ABE) and (ii) Ciphertext-Policy ABE (CP-ABE).

- In KP-ABE, the secret keys are labeled by a function $f(.)$, and the sender is unable to alter it. So, KP-ABE allows the access policies to be determined in the setup phase and cannot be modified afterward.
- The CP-ABE embeds a (policy) function $f(.)$ in the ciphertext to dictate which recipients are permitted access to the message's contents. In this method, the ciphertext gets a label depending on an arbitrary function $f(.)$, and secret keys are paired with attributes in the domain of $f(.)$. The plaintext can be decrypted by the receivers if and only if the data users' attribute set meets the condition $f(.)$, that is, $f(\mathbb{A}) = 1$. So, CP-ABE allows the data owner to define the access policies and data control.

In our paper, we use the CP-ABE for the design of our ABAC model, which is more suitable for access control schemes.

The first CP-ABE scheme, proposed by Bethencourt et al. in [11], enabled the protected data to satisfy access policies arbitrarily defined by the data owners with respect to any monotonic formula. The main drawback of this scheme is that it is based on the Generic Group Model (GGM), which is regarded as a weak security model. As a subsequent work, Cheung et al. [12] constructed a variant of the CP-ABE scheme in the standard model that is capable of dealing with constraints generated from any AND-gates circuit. Despite the short ciphertext size of this scheme, its policy function is limited to AND-gate circuits only. The decryption algorithm of this scheme requires a linear number of pairings which makes it inefficient in practice. Waters in [13] proposed a CP-ABE scheme that is asymptotically efficient in the standard model. Its design is based on a Linear Secret Sharing Scheme (LSSS) to construct an arbitrary access policy. As this scheme also requires a linear number of pairings to decrypt, it is not efficient in practice. In [14], the authors proposed a CP-ABE scheme that supports users' revocation based on lattices. However, their revocation mechanism is based on the principle that every time access permissions of a particular node need to be changed, the corresponding keys need to be updated for that node. This mechanism is not efficient.

The authors in [15] proposed an anonymous decentralized CP-ABE scheme based on bilinear pairings and secure outsourcing. However, their scheme does not support user revocation and delegation. They also employ cryptographic accumulators to verify the user's attributes anonymously. Moreover, the authors only consider static attributes. Cui et al. in [16] propose a key regeneration-free ABE scheme based on CP-ABE. This scheme is based on bilinear pairings and provides user key traceability for a user who delegates his/her decryption key. This scheme does not support dynamic attributes and user revocation.

In [17], the authors proposed a hierarchical ABE scheme that allows a trusted server to delegate part of its responsibility to other authorities, which can independently decide on the semantics of their attributes. Hierarchical ABE also allows for proxy re-encryption, but it forces a fixed structure for the access policies, limiting the flexibility of the access control system. In [18], Sedaghat et al. presented a decryption outsourceable Ciphertext-Policy Attribute-Based Signcryption (CP-ABSC) construction based on linear secret sharing schemes. Though authorized end users are given a constant computational complexity to decrypt a message, they must store a linear number of keys in the memory. Moreover, their architecture cannot accommodate the multi-level security paradigm. The CP-ABE scheme proposed in [19] can verify the correctness of the transformed ciphertext for authorized users and unauthorized users. This scheme is based on bilinear pairings and Message Authentication Codes (MAC). This scheme also does not provide user revocation. In Table 1, we compare the related ABE schemes in different aspects.

Table 1
Comparison of CP-ABE schemes.

Scheme	AS	DO	DI	SM	Attributes
BSW07 [6]	Tree	✗	✗	CPA	Static
Waters11 [13]	LSSS	✗	✗	CCA	Static
Cui et al. [16]	LSSS	✗	✗	CPA	Static
Nasiraee et al. [15]	Tree	✓	✗	RCPA	Static
Xiong et al. [20]	LSSS	✓	✗	CPA	Static
Sedaghat et al. [18]	LSSS	✓	✓	CCA	Static
Li et al. [19]	AND	✓	✓	RCPA	Static
Our scheme	AND	✓	✓	CPA	Static/Dynamic

AS stands for Access structure, DO stands for Decryption Outsourceable, DI stands for Data Integrity, SM stands for Security Model. When a scheme lacks a property, it will be mentioned by ✗.

2.2. Multi-level schemes

The authors of [21,22] both propose new ABE schemes and claim that their proposed schemes provide multi-level access control. As in classical ABE schemes, these schemes require that the data user satisfy a set of attributes to be able to decrypt any given encrypted data considering the access structure. But their proposed schemes does not offer multi-level access to encrypted data based on the security level of the data users. To address this issue, the authors of [23] proposed a new ABAC model with two layers of encryption on the data owner side and on the cloud side. However, in this model, the cloud performs an extra fine-grained encryption on top of the encrypted data, which is inefficient. Additionally, these schemes are not conform to NIST's ABAC model.

2.3. ABAC schemes

Another aspect to highlight is the mathematically grounded policy based on formal access control models. Our literature review for ABAC schemes based on ABE suggests that little or no effort has been made to provide a conceptual model. Gupta and Sandhu [24] proposed an administration model for Hierarchical Group and Attribute-Based Access Control (HGABAC) to control the assignment and removal of attributes, using three sub-models: User Attribute Assignment (UAA), User-Group Attribute Assignment (UGAA) and User to user-Group Assignment (UGA). However, they did not propose any scheme for their model. Bhatt et al. [25] formalized a restricted HGABAC model called rHGABAC, and implemented their model in the NIST Policy Machine (PM) tool [26–28]. The authors of [29] propose an ABAC model based on a broadcast key management scheme for attribute-based access control based on Single-Layer Encryption (SLE) schemes. In this scheme, the data owner in the encryption phase enforces access control policies (ACPs). Despite SLE's improvement over its predecessors, it consumes a lot of computational and storage resources for changing the group key [30] regularly. The approach also requires that all ACPs to be enforced by encryption, both initially and subsequently after users are added or revoked. This encryption work must be done by the data owner, which incurs high communication and computation costs. In this paper, we formalize our proposed access control scheme by providing a conceptual and formal model, in contrast to prior work in ABE-based ABAC.

Additionally, it is our primary objective to minimize the ciphertext size, while existing CP-ABE schemes suffer from either a large ciphertext size or a high computational overhead. To accomplish this, CP-ABE schemes based on AND-gate circuits are considered promising candidates. Based on the concept of AND-gate circuits, several constant-size ciphertext CP-ABE schemes are proposed [31–33], their ciphertext contains at least one element of the target group.

3. MLS-ABAC scheme overview

This section provides an overview of the proposed scheme, including the proposed architecture and a high-level description of the proposed MLS-ABAC, which complies with NIST's ABAC model.

As illustrated in Fig. 2, NIST's ABAC model has four main functional components: (i) Policy Enforcement Point (PEP), (ii) Policy Decision Point (PDP), (iii) Policy Information Point (PIP), and (iv) Policy Administration Point (PAP). In this model, policies are converted into Digital Policies (DPs), which are access control rules. Based on the NIST standard, the ABAC model also requires Meta policies (MPs)-policies about policies — for managing DPs, such as the assignment of priorities. More precisely, the NIST's ABAC model works as follows: (i) PEP interprets an access request from an authenticated subject and sends the request to PDP. It finally responds to the request based on the PDP's decision. (ii) PDP assesses the access policy obtained from PAP in terms of the attribute values fetched by querying PIP to determine if access should be granted. This evaluation is based on DPs and MPs. (iii) PIP presents attribute values (e.g., environmental conditions or static attributes) upon receiving a request from the PDP component. (iv) PAP verifies and manages DPs and MPs the PDP later uses for evaluation.

Although the NIST's ABAC model supports all the ABAC functionalities, it does not provide data confidentiality. To overcome this shortcoming, we proposed the MLS-ABAC model, which not only works based on the original workflow of the standard ABAC model but also ensures data confidentiality.

Further, as stated in [34], in the ABAC models the cloud server may be required to search the entire database in order to grant one data to a data user. Consequently, retrieval times can be very long and, as a result databases are difficult to maintain. As part of our proposed MLS-ABAC scheme, the cloud server first determines the data user's security level, then searches the database based on the security level. Therefore, MLS-ABAC is more practical than traditional ABAC models.

3.1. MLS-ABAC architecture

The architecture of our solution is illustrated in Fig. 3, there are five main entities, including:

- **Attribute Authority Server (AAS)** is a trusted party that generates the system parameters and also the secret keys for the data users.

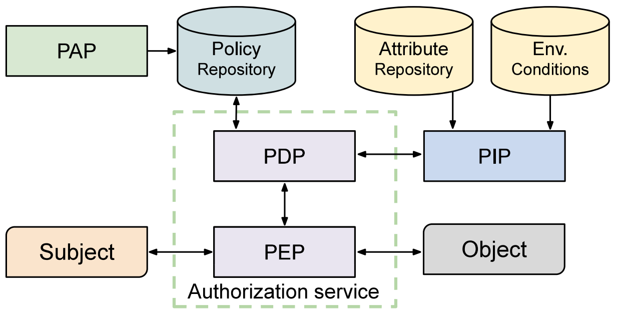


Fig. 2. NIST's ABAC model.

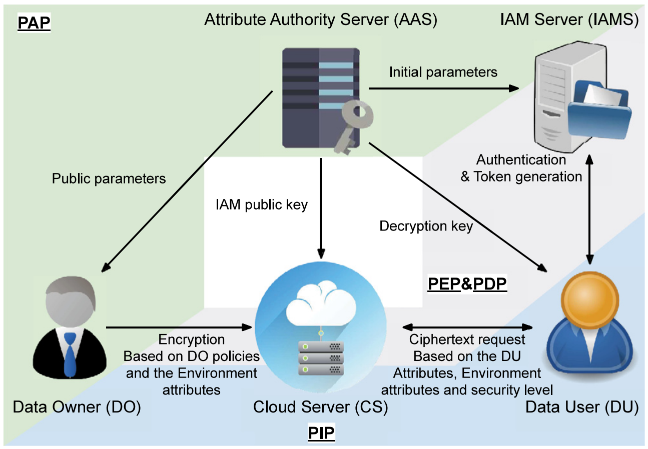

PAP
Attribute Authority Server (AAS)
IAM Server (IAMS)
Initial parameters
Public parameters
Authentication & Token generation
IAM public key
Decryption key
PEP&PDP
Encryption Based on DO policies and the Environment attributes
Ciphertext request Based on the DU Attributes, Environment attributes and security level
Data Owner (DO)
Cloud Server (CS)
Data User (DU)
PIP

Fig. 3. System Architecture of MLS-ABAC.

- **Identity and Access Management Server (IAMS)** creates tokens for users corresponding to their access security level based on sets of security levels. It is noteworthy that AAS creates, deletes and updates these sets and then submits them to the IAMS.
- **Cloud Server (CS)** acts as a repository and stores ciphertexts that are offloaded from IoT data producers. CS also checks the validity of the received tokens from another IoT device as a data consumer. Then CS decides whether access should be granted based on the defined predicate functions. It is assumed CS is honest-but-curious, which means it follows the protocol while curious to learn sensitive data.
- **Data Owner (DO)** is an IoT data producer who wants to share a sensitive message amongst users by defining a security level. To this end, it generates a ciphertext that carries some metadata as well. Therefore, to prevent data leakage, the DO chooses the security level for the ciphertext and uploads both ciphertext and associated security level to CS.
- **Data User (DU)** is a lightweight IoT device that wants to read data from the CS using its credentials (i.e., data consumer). To this aim, it requests a token from the IAMS and then transmits it to CS. If the DU passes the verification phase, it can obtain the ciphertext. Finally, if DU's attributes satisfy the access policy, the DU can decrypt the ciphertext to learn the entire message; otherwise, it learns nothing.

As depicted in Fig. 3, in the proposed MLS-ABAC model, the PAP component relies on the AAS and DO, the PIP component relies on the DO, CS and DU and finally, IAMS, CS and DO are acting as PEP and PDP components that are entirely consistent with the standard ABAC model.

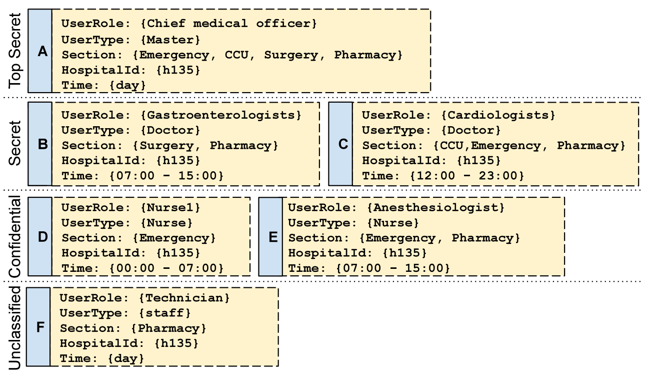


Fig. 4. Multi-level Security with ABAC Example.

3.2. *Multi-level security for ABAC*

First, we would like to provide an example of MLS-ABAC as illustrated in Fig. 4.

In this example, {UserType, HospitalId} is the set of static attributes and {Section, Time} is the set of dynamic attributes. Our proposed model has a set of four security levels which are Top Secret, Secret, Confidential, and Unclassified where User A belongs to security level Top Secret, User B and User C belong to Secret, User D and User E belong to Confidential, and User F belongs to Unclassified security level. The admin of the system (here, IAMS) defines this partial order hierarchy among users and will update them when the system policy changes.

In this system, the entity User E can download the information which are uploaded in to the security levels Confidential and Unclassified, and can decrypt the ones that are wrapped with the User E's static and dynamic attributes. Further, if the security level for User E is upgraded to Secret, it can download the information which are uploaded in to the security levels Secret, and below, and can decrypt the ones that wrapped with the Anesthesiologist's static and dynamic attributes.

Based on the above scenario, the admin of the system easily can *revoke* or *delegate* the access right of the User E to the wrapped information uploaded in to the security level Secret.

As demonstrated above, the MLS-ABAC scheme should ensure that a DU who can present all the necessary static and dynamic attributes is allowed to access the information for which they can satisfy the security level. Besides, the information should be stored securely to the CS. Thus, the ABE scheme should encrypt the information in such a way that the encrypted data can only be accessed by DU who have the credentials for the defined attributes. Moreover, the data integrity of the plaintext in the case of outsourced decryption should be preserved using an encryption scheme that simultaneously verifies the confidentiality and authenticity of data. Therefore, a lightweight authenticated encryption scheme should be deployed. Therefore, by combining the observations above, we can now summarize the main building blocks in our proposed MLS-ABAC scheme: (i) CP-ABE, (ii) authenticated encryption and cryptographic hash function. In the next section, we will zoom in on these different building blocks.

4. Building-block collection

In this section, we first provide definitions and preliminaries, and then provide the main building-block of the MLS-ABAC scheme including our proposed Decryption Outsourceable CP-ABE scheme, ASCON Cipher Suite [35] and JSON Web Token (JWT) [36].

4.1. *Preliminaries and definitions*

Throughout the paper, we take λ as a predetermined security parameter, where $\mathsf{negl}(\lambda)$ denotes a negligible function. The ith component scalar of each set $\mathbb{A}$ is represented by A_i. The notation $x \leftarrow\!\$\ \chi$ indicates that x is sampled uniformly according to a distribution χ. The integer numbers between 1 and n (i.e, the set of $\{1, \ldots, n\}$) is expressed by $[n]$. For a randomized algorithm $(\text{out}) \leftarrow \mathsf{alg}(\text{in})$, the term $\mathsf{alg}(\text{in}; r)$ assigns a certain random value r to the algorithm. For a given prime order field $\mathbb{F}$, the set of univariate polynomials of degree less than d over $\mathbb{F}$ is denoted by $\mathbb{F}_{<d}[X]$. Finally, PPT stands for "Probabilistic Polynomial Time".

Definition 1 (*Binary Representation of a Subset*)**.** For a given universe set $\mathbb{U}$ of size n, each subset $\mathbb{A}$ can be represented as a binary string of length n. Particularly, the ith element of a binary string for the set $\mathbb{A} \subseteq \mathbb{U}$ is equal to 1 (i.e., $a[i] = 1$) if $A_i = U_i$. We denote a binary representation set as a binary tuple $(a[1], \ldots, a[n]) \in \mathbb{Z}_2^n$.

Definition 2 (*Zero-polynomial*)**.** For a finite set of $\mathbb{U} = \{k_1, \ldots, k_n\}$, we define the zero-polynomial $Z_{\mathbb{A}}(X)$ for a nonempty subset of $\mathbb{A} \subset \mathbb{U}$ as $Z_{\mathbb{A}}(X) := \prod_{i=1}^{n} (X - k_i)^{\overline{a[i]}}$, where $\overline{a[i]}$ is the binary representation of the complement set $\overline{\mathbb{A}}$. In other words, this univariate polynomial vanishes on all the components of the set $\mathbb{U}$ such that the binary representation of the subset $\mathbb{A}$ is zero.

Definition 3 (*Bilinear Groups [37]*)**.** Let a Type-III[2] bilinear group generator $\mathcal{BG}(1^{\lambda})$ under security parameter λ, returns $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathsf{p}, \hat{e})$, such that $\mathbb{G}_i$, $i \in \{1, 2, T\}$ are cyclic groups of the same prime order p with generators $[1]_i$, and $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is an efficiently computable bilinear map with the following properties.

- $\forall\, a, b \in \mathbb{Z}_p$, $[a]_1 \bullet [b]_2 = [ab]_T = [b]_1 \bullet [a]_2$.
- $[1]_T \neq 1$.
- For all $[1]_1$ and $[1]_2$, there exists an efficient algorithm for computing $[1]_T$.

Where we use bracket notation and for randomly selected generators $g \in \mathbb{G}_1$ and $h \in \mathbb{G}_2$ and $[x]_1$ denotes $x \cdot g \in \mathbb{G}_1$. Moreover, we denote the pairing with square bracket notation as $\hat{e}(g^a, h^b) = [a]_1 \bullet [b]_2$.

Definition 4 ((l, m, t) - *MSE-DDH Assumption [38]*)**.** Under security parameter λ, we select an asymmetric bilinear group generator $\mathcal{BG}(\lambda) = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathsf{p}, \hat{e})$. For given three integers l, m, t, let f and h be two univariate composite polynomials of degree l and m that vanishes on pairwise distinct points $\vec{x} = (x_1, \ldots, x_l)$ and $\vec{y} = (y_1, \ldots, y_m)$, respectively. For randomly chosen integers $\alpha, \delta, k \leftarrow\!\$\ \mathbb{Z}_p^*$, the (l, m, t)-Multi-Sequence of Exponents Diffie–Hellman assumption states that no PPT adversary $\mathcal{A}$ can distinguish between $\Gamma = [kf(\alpha)]_T$ and a random element $\Gamma \leftarrow\!\$\ \mathbb{G}_T$ with a non-negligible advantage, when given four tuples $\vec{v_1} = \left([1]_1, [\alpha]_1, \ldots, \left[\alpha^{l+t-2}\right]_1, [k\alpha f(\alpha)]_1\right)$, $\vec{v_2} = \left([\delta]_1, [\delta\alpha]_1, \ldots, \left[\delta\alpha^{l+t}\right]_1\right)$, $\vec{v_3} = \left([1]_2, [\alpha]_2, \ldots, \left[\alpha^{m-2}\right]_2\right)$ and $\vec{v_4} = \left([\delta]_2, [\delta\alpha]_2, \ldots, \left[\delta\alpha^{2m-1}\right]_2, [kh(\alpha)]_2\right)$. The advantage of a PPT adversary $\mathcal{A}$ for solving the (l, m, t)-MSE-DDH assumption is,

$$\left| \Pr\left[\mathcal{A}\left(\vec{x}, \vec{y}, \vec{v_{1-4}}, \Gamma = [kf(\alpha)]_T\right) = 1\right] - \Pr\left[\mathcal{A}\left(\vec{x}, \vec{y}, \vec{v_{1-4}}, \Gamma \leftarrow\!\$\ \mathbb{G}_T\right) = 1\right] \right| \leq \mathsf{negl}(\lambda).$$

[2] In this type, $\mathbb{G}_1 \neq \mathbb{G}_2$ and also there is no efficient algorithm to compute nontrivial homomorphism in both directions $\mathbb{G}_1$ and $\mathbb{G}_2$.

Definition 5 (*Access Structure* [39]). Let $\mathcal{P}$ be a set of parties $\mathcal{P} = \{P_1, \dots, P_n\}$. A collection $\mathcal{A} \subset 2^{\mathcal{P}}$ is called a monotone collection if $\forall B, C$ such that if $B \in \mathcal{A}$ *and* $B \subseteq C$, *then* $C \in \mathcal{A}$. Consequently a monotone collection of non-empty subset of $\mathcal{P}$ is defined as a monotone access structure in which all the sets in this monotone collection are called the authorized sets. On the contrary, all the sets that are not in this monotone collection are called unauthorized sets.

4.2. Decryption outsourceable CP-ABE scheme

Next we propose a decryption outsourceable CP-ABE scheme with constant key and ciphertext size. This construction is under the formal definition of decryption outsourceable CP-ABE schemes which is formalized as follows.

Definition 6 (*Decryption Outsourceable CP-ABE*)**.** A decryption outsourceable CP-ABE scheme over message space $\mathcal{M}$, ciphertext space $\mathcal{C}$ and a Boolean relation $\mathsf{Br} : \Sigma_k \times \Sigma_c \to \{0, 1\}$, where Σ_k is the set of key indices and Σ_c denotes the ciphertext indices, consists the following PPT algorithms.

- $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathcal{ABE}.\mathsf{PGen}(\lambda, \mathbb{U})$: This probabilistic algorithm takes the security parameter λ and an attribute space as inputs and generates the pair of public parameters and master secret key $(\mathsf{pp}, \mathsf{msk})$.
- $(\mathsf{dk}_x) \leftarrow \mathcal{ABE}.\mathsf{KGen}(\mathsf{msk}, x)$: This randomize algorithm takes a key index $x \in \Sigma_k$ and master secret key msk, and returns a private decryption key dk_x underlying key index $x \in \Sigma_k$.
- $(\mathsf{Ct}) \leftarrow \mathcal{ABE}.\mathsf{Enc}(\mathsf{pp}, m, y)$: The encryption algorithm to encrypt secret message $m \in \mathcal{M}$, gives public parameters pp and a ciphertext index $y \in \Sigma_c$ as inputs. It returns ciphertext Ct along with the underlying access rights y.
- $(\mathsf{tk}_x, \epsilon) \leftarrow \mathcal{ABE}.\mathsf{Tgen}(\mathsf{pp}, \mathsf{dk}_x)$: The token generation algorithm is a probabilistic algorithm that takes pp and a secret decryption key dk_x and returns a corresponding token tk_x along with some auxiliary information ϵ as outputs.
- $((\mathsf{pd}_x, \epsilon'), \perp) \leftarrow \mathcal{ABE}.\mathsf{PDec}(\mathsf{pp}, \mathsf{tk}_x, \mathsf{Ct})$: The partial decryption algorithm takes pp, a token $\mathsf{tk}_{\mathbb{B}}$ and underlying auxiliary information ϵ along with a ciphertext under access policy $y \in \Sigma_c$. If $\mathsf{Br}(x, y) = 1$, it computes the partially decrypted value pd_x and auxiliary information ϵ', otherwise, it responses by $\perp$.
- $(m', \perp) \leftarrow \mathcal{ABE}.\mathsf{Dec}(\mathsf{pp}, \mathsf{pd}_x, \epsilon, \mathsf{dk}_x)$: This deterministic algorithm to decrypt a ciphertext with access policy y takes pp and a valid secret decryption key dk_x. It responses by $m' \in \mathcal{M}$ iff $\mathsf{Br}(x, y) = 1$, else returns $\perp$.

The main security requirements for a decryption outsourceable CP-ABE scheme are Correctness and INDistinguishability against Chosen Plaintext Attack (IND-CPA).

Definition 7 (*Correctness*)**.** We call a decryption outsourceable CP-ABE scheme consistent if we have,

$$\Pr[\mathsf{Dec}\,(\mathsf{pp}, \mathsf{Enc}(\mathsf{pp}, m, y), \mathsf{dk}_x) = m : \mathsf{Br}(x, y) = 1] \approx_c 1$$

Definition 8 (IND-CPA)**.** Let us have a decryption outsourceable CP-ABE scheme over the Boolean relation $\mathsf{Br} : \Sigma_k \times \Sigma_c \to \{0, 1\}$, message space $\mathcal{M}$ and ciphertext space $\mathcal{C}$. For the security parameter λ and a PPT adversary $\mathcal{A}$, we recall the selectively-secure INDistinguishability game under Chosen Ciphertext Attack (IND-CPA) as follows.

- **Initialization:** Adversary $\mathcal{A}$ chooses a query set Q and a challenge access structure y^* such that $\mathsf{Br}(\{x_i\}_{x_i \in Q}, y^*) = 0$. The challenger $\mathcal{B}$ samples the pair of public parameters and the master secret key by running the algorithm $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{PGen}(\lambda, \mathbb{U}, Q)$ and publishes pp and keeps msk secure.
- **1st Query phase.** The adversary $\mathcal{A}$ can send a polynomially bounded number of queries to the decryption key and encryption oracles.
- **Challenge:** $\mathcal{A}$ sends two challenge messages of the same length $(m_0, m_1) \overset{\$}{\leftarrow} \mathcal{M} \times \mathcal{M}$ to $\mathcal{B}$. Then $\mathcal{B}$ flips a fair coin and produces a random bit $b \overset{\$}{\leftarrow} \{0, 1\}$, and executes $\mathsf{Ct}^* \leftarrow \mathsf{Enc}(m_b, y^*)$ and returns back Ct^*.
- **2nd Query phase.** Upon receiving the challenge ciphertext Ct^*, $\mathcal{A}$ can query the oracles in the first query phase within the restriction that she cannot request the decryption of Ct^* under key indices $x_i \in \mathcal{Q}$ and challenge access policy y^*.
- **Guess.** $\mathcal{A}$ guesses the bit $b' \in \{0, 1\}$ and we say a PPT adversary $\mathcal{A}$ wins the IND-CPA security game if she guesses the random bit b better than by chance.

For the advantage
$Adv^{\text{IND-CPA}}_{\Pi_{\text{CP-ABE}},\mathcal{A}}(1^\lambda, b) = 2\,|Pr\,[\mathcal{A} \text{ wins the IND-CPA game}] - \frac{1}{2}|$,
we say $\Pi_{\text{CP-ABE}}$ is IND-CPA-secure if for all the PPT adversaries $\mathcal{A}$.

$$\Big|\Pr[Adv^{\text{IND-CPA}}_{\Pi_{\text{CP-ABE}},\mathcal{A}}(1^\lambda, b = 0) = 1] - \Pr[Adv^{\text{IND-CPA}}_{\Pi_{\text{CP-ABE}},\mathcal{A}}(1^\lambda, b = 1) = 1]\Big| \approx_c 0. \tag{1}$$

For a given Type-III bilinear group pairing $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathsf{p}, \hat{e})$, the introduced CP-ABE scheme over AND-gate access structures (Definition 5) consists the following PPT algorithms.

- $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathcal{ABE}.\mathsf{PGen}(1^\lambda, \mathbb{U})$: For a given attribute space $\mathbb{U} = (A_1, \dots, A_n)$ and security parameter λ, it calculates $k_i = \mathsf{H}(A_i)$ such that $\mathsf{H}(.)$ is a collision-resistant hash function in the random oracle model. It uniformly samples random $s \overset{\$}{\leftarrow} \mathbb{Z}^*_{\mathsf{p}}$, and computes $g_2 = \left[s^2\right]_1$ and $h_j = \left\{\left[s^j\right]_2\right\}_{j=0}^{n}$ as the standard basis under secret value s in the second cyclic group. Finally, it returns the master secret key and public parameters as $\mathsf{msk} = \{[1]_1, s\}$ and $\mathsf{pp} = \{g_2, h_j, [s]_T, \mathsf{H}\}$, respectively.
- $(dk_{\mathbb{B}}) \leftarrow \mathcal{ABE}.\mathsf{KGen}(\mathsf{msk}, \mathbb{B})$: This algorithm takes the master secret key msk and a key index $\mathbb{B} \subset \mathbb{U}$, and then returns the secret decryption key $\mathsf{dk}_{\mathbb{B}}$ as output. It samples a unique random integer $t_u \overset{\$}{\leftarrow} \mathbb{Z}^*_p$ and computes the zero-polynomial $Z_{\mathbb{B}}(x) = \prod_{j=1}^{n}(x - k_j)^{1-b_j}$. At this point, it returns the secret decryption key corresponding to subset $\mathbb{B} \subset \mathbb{U}$ as $\mathsf{dk}_{\mathbb{B}} = \big([1/Z_{\mathbb{B}}(s)]_1\big)$.
- $(\mathsf{Ct}) \leftarrow \mathcal{ABE}.\mathsf{Enc}(\mathsf{pp}, m, \mathbb{P})$: The encryption algorithm takes public parameters pp as inputs and encrypts a secret message $m \in \mathcal{M}$ under an access structure $\mathbb{P} \subset \mathbb{U}$. More precisely, it samples a random integer $r \overset{\$}{\leftarrow} Z^*_{\mathsf{p}}$, computes the Zero-polynomial $Z_{\mathbb{P}}(x) = \prod_{i=1}^{n}(x - k_i)^{1-p[i]} = \sum_{i=0}^{n} z_i x^i$ and calculates $\mathsf{Ct} = (C, C_1, C_2) = \big(m\,[rs]_T\,, g_2^{-r}, [rsZ_{\mathbb{P}}(s)]_2\big)$ and outputs Ct.
- $(\mathsf{tk}_{\mathbb{B}}, C'_1, \epsilon) \leftarrow \mathcal{ABE}.\mathsf{Tgen}(\mathsf{pp}, \mathsf{dk}_{\mathbb{B}}; \mu)$: The token generation algorithm is a probabilistic algorithm that takes pp and a secret decryption key $\mathsf{dk}_{\mathbb{B}}$ and then samples a uniformly random integer $\mu \overset{\$}{\leftarrow} \mathbb{Z}^*_q$. It finally returns $\mathsf{tk}_{\mathbb{B}} = \mathsf{dk}^{\mu}_{\mathbb{B}}$, $C'_1 = C^{\mu}_1$ and secret auxiliary value $\epsilon = \mu$ as outputs.
- $(\mathsf{pd}_{\mathbb{B}}, C) \leftarrow \mathcal{ABE}.\mathsf{PDec}(\mathsf{pp}, \mathsf{tk}_{\mathbb{B}}, \mathsf{Ct})$: The partial decryption algorithm takes pp, a token $\mathsf{tk}_{\mathbb{B}}$ and a ciphertext under access policy $\mathbb{P}$ and checks $\mathbb{P} \subseteq \mathbb{B}$. It computes the univariate polynomial $L_{\mathbb{P},\mathbb{B}} = \prod_{j=1}^{n}(x - k_j)^{c[j]} = \sum_{i=0}^{n} \ell_i x^i$, where $c[j] =$

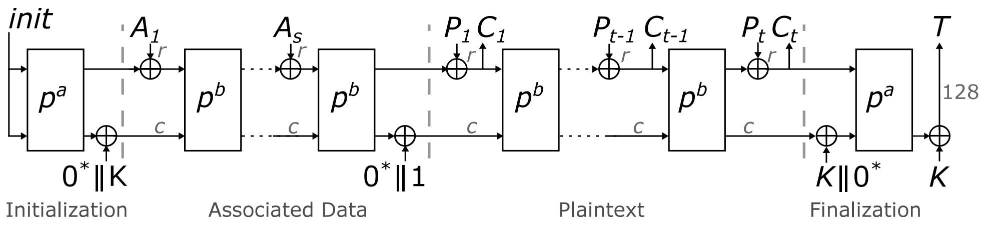


Fig. 5. Ascon-128 Encryption [35].

$b[j]-p[j]$. Then, it computes $\mathsf{pd}_{\mathbb{B}} = \left((C'_1 \bullet V)(\mathsf{tk}_{\mathbb{B}} \bullet C_2)\right)^{1/\ell_0} = [-rs\mu]_T$, where $V = \prod_{i=1}^{n} h_{i-1}^{\ell_i}$ and returns $(\mathsf{pd}_{\mathbb{B}}, C)$ as output.

- $(m, \perp) \leftarrow \mathcal{ABE}.\mathsf{Dec}(\mathsf{pp}, \mathsf{pd}_{\mathbb{B}}, C, \epsilon)$: The DU, who knows the secret auxiliary value of ϵ, can run this deterministic algorithm to obtain message m. This algorithm takes the partially decrypted value $\mathsf{pd}_{\mathbb{B}}$ and C as inputs and calculates $m := C \cdot \mathsf{pd}_{\mathbb{B}}^{1/\mu}$.

Theorem 1 (*Correctness*). *The proposed decryption outsourceable CP-ABE is consistent.*

Proof. The proof is provided in A.1. □

Theorem 2 (IND-CPA). *Under the hardness of (l, m, t)-MSE-DDH, no PPT adversary $\mathcal{A}$ can win the defined IND-CPA game with a non-negligible advantage for the proposed CP-ABE scheme.*

Proof. We prove this theorem in A.2. □

4.3. ASCON Cipher Suite

In order to enable message integrity and prevent sending target group elements as part of the ciphertext, we integrate the ASCON cipher suite to combine provably secure authenticated encryption (Ascon-128) and hashing (Ascon-Hash) algorithms with our proposed ABE scheme [35]. Ascon-128 provides 128-bit security for confidentiality of plaintext as well as the integrity of both plaintext and associated data. Also, Ascon-Hash has output length of 256-bit and provides 128-bit security against collision and (second) pre-image attacks [35]. Our MLS-ABAC scheme consists of $Ascon.\mathsf{AENC}(\cdot)$, $Ascon.\mathsf{H}(\cdot)$ and $Ascon.\mathsf{NGen}(\cdot)$ algorithms which are Ascon-128, Ascon-Hash and nonce generation functions described as follows.

Ascon-128 is a lightweight authenticated encryption/decryption scheme based on a sponge construction which uses 320-bit permutation boxes. Fig. 5 demonstrates all four states of the Ascon-128 encryption. The decryption process in Ascon-128 is the same as the encryption process with the only difference that after the associated data phase, the ciphertext is used instead of the plaintext in the third stage. The encryption process gets as input a plaintext of arbitrary length, a 128-bit secret key, a 128-bit nonce, and an associated data of arbitrary length. It then returns the ciphertext with the same length as the plaintext and 128-bit tag as the output. The tag will be used in the decryption process to authenticate both associated data and plaintext.

Initialization state. Initialization state of Ascon-128 consists of 320 bits named *init* and formed by the 128-bit secret key, a 128-bit nonce, as well as an initial vector (IV).

Associated data. Associated data is additional data that is optional. Associated data is not encrypted and its integrity will be confirmed in the decryption process. The encryption and decryption processes of Ascon-128 are as below.

Decryption process. Algorithm 1 presents this process. This algorithm returns the plaintext P if the tag T_C is verified.

Algorithm 1 Ascon-128 Decryption Algorithm.

```
1: **Input.** $C, T_C$, nonce $N$, associated data $A$, secret key $K$
2: **Output.** $P$ or $\perp$
3: $(P, T'_C) \leftarrow \mathsf{ADEC}(K, N, A, C)$
4: **if** $T'_C == T_C$:
5:   **return** $P$
6: **else return** $\perp$
```

The Ascon-Hash function is based on sponges [40]. This hash function has an output length of 256-bit and uses the Algorithm 2 to generate this output, where p^a is the a-round permutation. The architecture of the Ascon-Hash function is shown in Fig. 6.

Algorithm 2 Ascon-Hash Hashing Algorithm.

```
1: **Input.** message $M$
2: **Output.** $H \in \{0,1\}^l$
3: $\mathcal{S} \leftarrow p^a(init_{Hash})$
4: $M_1 \ldots M_s \leftarrow M\|1\|0^\star$
5: **for** $i = 1, \ldots, s$:
6:   $\mathcal{S} \leftarrow p^a((\mathcal{S}_r \oplus M_i)\|\mathcal{S}_{256})$
7: **for** $i = 1, \ldots, t = \lceil \frac{256}{r} \rceil$:
8:   $H_i \leftarrow \mathcal{S}_r$
9:   $\mathcal{S} \leftarrow p^a(\mathcal{S})$
10: **return** $\lfloor H_1\| \ldots \|H_t \rfloor_{256}$
```

Initialization. In Ascon-Hash, initialization state consists of a 320-bit bitstring named $init_{Hash}$. This string contains an initial vector IV as well as 256-bit 0.

4.4. JSON Web Token Structure

JWT is a string comprising three parts, namely a header, a payload, and a signature. The header contains ``alg'' and ``kid'' which are the signing algorithm being used and a hint for the identity of the key used to secure the JWT, respectively. Since the IAMS has only one private/public key pair, ``kid'' is not used in our JWT. In our scheme we use the RS256 algorithm in which the digital signature algorithm is RSASSA-PKCS1-v1_5 using SHA-256[3]. The payload contains claims stored inside the JWT which are the security levels of the DU, the identity of the IAMS, and the expiration time/date of the Token. The last part of the JWT contains the signature of the header and the payload using the private key of the IAMS [36]. We use this Token (TK_{it}) for the information exchange process between DU and IAMS as well as the authentication process between DU and CS. Fig. 7 shows an example of a header and payload in our scheme.

JWT is an encoded signed string that includes the DU claims as ``sl'':``security level(s)'', ``aud'':``identity of the IAMS'' and "exp": "Token expiration time". We use this Token (TK_{it}) for the information exchange process between DU

[3] RSASSA-PKCS1-V1_5 uses the RSA private key to sign the message and defined in www.ietf.org/rfc/rfc3447.txt.

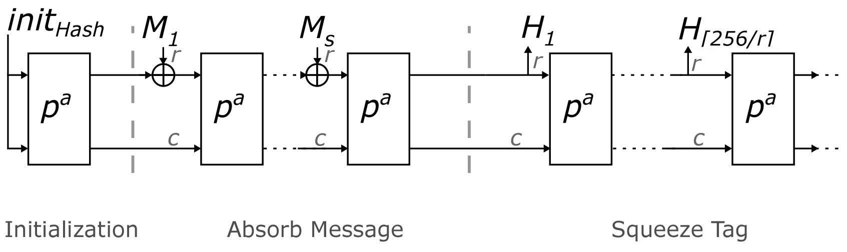


Fig. 6. Ascon-Hash Function, $c = 256$ [35].

```
Header:
{
 "typ": "JWT",
 "alg": "RS256"
}
Payload:
{
 "sl": "security level(s)",
 "aud": "identity of the IAM server",
 "exp": a number that indicates expiration time
}
```

Fig. 7. Header and Payload of JWT in MLS-ABAC Scheme.

and IAMS, as well as the authentication process between DU and CS. Due to the expiration time included in the JWT, it is only valid for a limited amount of time and it will fail to authenticate if the JWT is expired. An JWT scheme can be formalized using the following algorithms, which are based on our specifications.

- $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathcal{JWT}.\mathsf{svGen}(\lambda)$: This algorithm takes the security parameter λ as input and returns the pair of signing and verification keys $(\mathsf{sk}, \mathsf{vk})$.
- $TK_{it} \leftarrow \mathcal{JWT}.\mathsf{Tgen}(\mathsf{sk}, sl, aud, exp)$: This algorithm takes the signing key sk, DU's security level sl, the aud which is the identity of the IAMS, and the expiration date exp as inputs, and then returns the token TK_{it} as output.
- $(1, \perp) \leftarrow \mathcal{JWT}.\mathsf{Vf}(\mathsf{vk}, TK_{it})$: This algorithm takes the token verification key vk and the token TK_{it} as inputs. It checks the validity of the IAMS signature and the token expiration times and returns 1 in the case of acceptance, otherwise it responses by $\perp$.

5. Proposed MLS-ABAC scheme

In this section, we first explain both Ascon-128 and Ascon-Hash initialization states, and then we describe our proposed MLS-ABAC scheme which consists of seven PPT algorithms.

5.1. Ascon-Hash and Ascon-128 initialization

The initialization phase of Ascon-Hash consists of 320 bits named $init_{Hash}$ and is defines as follows: $init_{Hash} = p^a(0^8 \| ns \| nd \| 0^8 \| h \| 0^{256})$ in which ns and nd denote the number of static attributes and dynamic attributes respectively. Each of these are defined as an 8 bits integer. Moreover, $h = 256$ is 32 bits integer and p^a is the a-round permutation. Also, the initialization phase of Ascon-128 consists of 320 bits named $init$. In the proposed scheme, $init$ can be derived as $init = \mathsf{H}^{64} \| K \| N$ in which H^{64} is the first 64 bits of $\mathsf{H} = Ascon.\mathsf{H}(\psi_1 \| \psi_2 \| \dots \| \psi_m \| d)$, K is 128 bits secret key and N is 128 bits nonce. Where, $\{\psi_i\}_{i \in [m]}$ is a bit-string representation of the dynamic attributes and d is an element in the target group $\mathbb{G}_T$.

5.2. MLS-ABAC construction

The proposed MLS-ABAC scheme is presented in Algorithm 3 and described as follows.

- **AAS Parameter Generation**: AAS by taking the security parameter λ and the attribute space $\mathbb{U}$ executes this probabilistic algorithm and returns the master secret key and public parameters of the system.
- **IAMS Key Generation**: This algorithm is executed by IAMS. It takes the security parameter λ and returns the pair of signing and verification keys $(\mathsf{sk}, \mathsf{vk})$ and publishes vk while keeps secure the signing key sk.
- **Key Generation**: This randomized algorithm is executed by AAS and produces the secret decryption key as output under the static attribute set $\mathbb{B} \subset \mathbb{U}$.
- **Encryption**: The encryption algorithm is executed by the DO and it is based on $Ascon.\mathsf{AENC}(\cdot)$ and $\mathcal{ABE}.\mathsf{Enc}(\cdot)$ algorithms described in Sections Section 4.3 and 4.2, respectively. The DO determines the access rights to the secret encrypted message in both static and dynamic attribute sets and grants access to the ciphertext only to a specific group of DUs. In this algorithm, $\psi_1 \| \psi_2 \| \cdots \| \psi_m$ are the bit-string representation of the m numbers of dynamic environmental attributes.
- **Token Generation**: In this probabilistic algorithm, the DU first sends the *query* to the IAMS and requests a JSON Web Token TK_{it} to get access to the stored ciphertexts in the CS. At this point, the IAMS decides whether DU is allowed to get access to a ciphertext with a certain security level (i.e., DU $\in sl$). To this end, IAMS generates the JWT token TK_{it} signed with sk using $\mathcal{JWT}.\mathsf{Tgen}(\cdot)$ algorithm including security level and also expiration time and returns them back the DU. Next, the DU uses a valid private decryption key and executes the $\mathcal{ABE}.\mathsf{Tgen}(\cdot)$ phase of the proposed decryption outsourceable CP-ABE to generate $(\mathsf{tk}_{\mathbb{B}})$ to outsource the resource-intensive part of the computations to CS.
- **Partial Decryption**: The CS checks the validity of the DU's token by running the $\mathcal{JWT}.\mathsf{Vf}(\mathsf{vk}, TK_{it})$ algorithm first. If the DU is allowed to have access to the stored ciphertext based on the given token TK_{it}, then the ciphertext is partially decrypted and returned to the DU, otherwise it returns $\perp$. It is important to stress that the $\mathcal{JWT}.\mathsf{Vf}(\cdot)$ algorithm verifies the IAMS signature and the token expiration date and the condition $[sl \in Ct_{sl}]$ verifies whether the ciphertext that the DU requested has the security level mentioned in the Token's claim.
- **Decryption**: This algorithm uses a partially decrypted value along with a hidden auxiliary information μ that is used in the token generation phase to securely decrypt data. In this algorithm, $\{\psi'_i\}_{i \in [m]}$ is a bit-string representation of the DU's dynamic attributes that should be equal to $\{\psi_i\}_{i \in [m]}$. Also, H'^{65_192} and H'^{193_256} mean 128 bits of H' from 65-*th* bit to 192-*th* bit and the last 64 bits of H', respectively.

Algorithm 3 Our construction.

```
1: Function AAS Parameter Generation($\lambda, \mathbb{U}, \mathcal{D}$):
2:   Actor. AAS
3:   $(\mathsf{msk}, \mathsf{pp}) \leftarrow \mathcal{ABE}.\mathsf{PGen}(\lambda, \mathbb{U})$ / ABE PGen algorithm
4:   return (pp, msk)

5: Function IAMS Key Generation($\lambda$):
6:   Actor. IAMS
7:   $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathcal{JWT}.\mathsf{svGen}(\lambda)$ / IAMS s/v KGen algorithm
8:   return (sk, vk)

9: Function Key Generation(pp, msk, $\mathbb{B}$, $i$):
10:  Actor. AAS
11:  $(\mathsf{dk}_{\mathbb{B}}) \leftarrow \mathcal{ABE}.\mathsf{KGen}(\mathsf{pp}, \mathsf{msk}, \mathbb{B})$ / ABE KGen algorithm
12:  return $(\mathsf{dk}_{\mathbb{B}})$

13: Function Encryption(pp, $m$, $\mathbb{P}$, $\beta$):
14:  Actor. DO
15:  $r \leftarrow\$ \mathbb{Z}_p^*$ / Samples a random integer
16:  $(\mathsf{Ct}' = (d, C_1, C_2)) \leftarrow \mathcal{ABE}.\mathsf{Enc}(\mathsf{pp}, m' = [1]_T, \mathbb{P}; r)$
17:  $\mathsf{H} \leftarrow Ascon.\mathsf{H}(\psi_1 \| \psi_2 \| \ldots \| \psi_m \| d)$ / Ascon hash function
18:  $N \leftarrow\$ Ascon.\mathsf{NGen}(\lambda, 128)$ / Ascon nonce generation algorithm
19:  $(C, T_C) \leftarrow Ascon.\mathsf{AENC}(\mathsf{H}^{65\_192}, N, \mathsf{H}^{193\_256}, m)$
20:  $\mathsf{Ct} = (C, C_1, C_2, N)$
21:  return $(\mathsf{Ct}, T_C)$

22: Function Token Generation(pp, $sl$, $aud$, $exp$, sk):
23:  Actors. DU, IAMS
24:  if DU $\in sl$:
25:      $TK_{it} \leftarrow \mathcal{JWT}.\mathsf{Tgen}(\mathsf{sk}, sl, aud, exp)$
   / IAMS generates the JWT token
26:      $\epsilon := \mu \leftarrow\$ \mathbb{Z}_p^*$
   / DU samples a uniformly random integer
27:      $(\mathsf{tk}_{\mathbb{B}}) \leftarrow \mathcal{ABE}.\mathsf{Tgen}(\mathsf{pp}, \mathsf{dk}_{\mathbb{B}}; \mu)$
   / DU runs the ABE token generation phase
28:      return $(TK_{it}, \mathsf{tk}_{\mathbb{B}}, \epsilon)$
29:  else   return 0

30: Function Partial Decryption(pp, Ct, $\mathsf{tk}_{\mathbb{B}}$, $TK_i$, vk):
31:  Actor. CS
32:  if $true \leftarrow \mathcal{JWT}.\mathsf{Vf}(\mathsf{vk}, TK_{it}) \wedge sl \in Ct_{sl}$:
33:     $(\mathsf{pd}_{\mathbb{B}}) \leftarrow \mathcal{ABE}.\mathsf{PDec}(\mathsf{Ct}, \mathsf{tk}_{\mathbb{B}})$
   / CS runs ABE partial decryption algorithm
34:     return $(\mathsf{pd}_{\mathbb{B}}, \mathsf{Ct}, T_C)$
35:  else return $\perp$

36: Function Decryption($\mu, T_C, C, N$):
37:  Actor. DU
38:  $d \leftarrow \mathcal{ABE}.\mathsf{PDec}(\mathsf{pp}, \mathsf{pd}_{\mathbb{B}}, \mathsf{Ct}; \mu)$
39:  $H' \leftarrow Ascon.\mathsf{H}(\psi_1' \| \psi_2' \| \ldots \| \psi_m' || d)$ / Ascon hash function
40:  $(m', T_C') \leftarrow Ascon.\mathsf{ADEC}(\mathsf{H}'^{65\_192}, N, \mathsf{H}'^{193\_256}, C)$
   / Ascon decryption algorithm
41:  if $T_C' == T_C$:
42:     return $m = m'$
43:  else return $\perp$
```

6. Security and performance

In this section, we briefly discuss the security intuition behind the proposed MLS-ABAC scheme and also provide a discussion about its performance.

6.1. Security analysis

We informally discuss three main security features for the proposed MLS-ABAC construction in terms of correctness, the secrecy of the plaintext and token unforgeability.

Correctness: Informally, an MLS-ABAC scheme is defined to be complete if and only if an authorized DU passes the token verification phase as well as the target recipients successfully decrypt the ciphertext and learn the plaintext correctly. This can be concluded under the correctness of underlying primitives including the proposed decryption outsourceable CP-ABE scheme (Theorem 1), the completeness of Jason Web Token (Section 4.4) and correctness of Ascon construction 4.3.

Confidentiality: This security feature makes it impossible for an adversary to learn meaningful information when (s)he does not comply with the ciphertext access rights. It is mainly derived from the concepts behind hard cryptographic assumptions like Discrete Logarithm and MSE-DDH assumptions and one-wayness of the underlying Ascon-hash function. Recall the IND-CPA security of the proposed CP-ABE, where no PPT adversary can win the defined indistinguishability security game. It is to be expected that an adversary without having knowledge on value d must correctly guess the inputs of the Ascon-hash function to execute the Ascon encryption successfully. Since the first cyclic group generator is hidden and the ciphertext is blinded by an initial random number, r, only those users who satisfy the ciphertext access right (i.e., $[\mathbb{P} \subseteq \mathbb{B}]$) can obtain the correct partially decrypted value $\mathsf{pd}_{\mathbb{B}}$. Moreover, since the decryption tokens are generated using a uniformly random integer μ, only the users who generated this token can deduce $d^{\mu^{-1}}$ by knowing μ and $\mathsf{pd}_{\mathbb{B}}$. In addition, CS has no way to learn information from decryption tokens since they are re-randomized by the DU under a random integer μ. We assume Distributed Denial-of-Service (DDOS) and Replay attacks [41,42] resistance to be important yet orthogonal to our work.

Unforgeability: This security property ensures that an adversary $\mathcal{A}$ cannot generate a verifiable JSON web token unless the token is created by the IAMS. This naturally comes from the unforgeability nature of the intended digital signature used in JWT. Moreover, since the proposed outsourceable CP-ABE scheme is secure against the collude attack, then an adversary in the CP-ABE's token generation phase cannot generate a decryption token beyond the authorized private decryption key and underlying attributes.

6.2. Performance analysis

MLS-ABAC, as mentioned earlier, is based on our proposed decryption outsourceable CP-ABE scheme. This means that the performance of our CP-ABE construction determines the efficiency of the MLS-ABAC system. This section describes how the proposed CP-ABE scheme performs. More precisely, we compare our proposed scheme with the existing decryption outsourceable ABE schemes in terms of communication and computation costs. Additionally, we will evaluate the obtained results for the proposed MLS-ABAC experiment in Section 7.

It can be seen from Table 2 that our proposed decryption outsourceable CP-ABE employed in our MLS-ABAC scheme is more efficient and has less communication overhead in comparison with the existing IoT-friendly ABE schemes. Precisely, the time required to execute the encryption phase in our proposed CP-ABE varies by the total number of attributes minus the number of attributes intended to the policy without any pairing. In addition, since the heavy computational parts of decryption are outsourced to the CS, a DU only needs to compute one exponentiation to

Table 2
Computation and communication comparison.

Scheme	Enc. Cost	Dec. Cost	pp. size	dk. length	Ct. length
Nasiraee et al. [15]	$(2\mathcal{S}+1)\mathbf{E}+\mathbf{P}$	$\mathcal{S}\mathbf{M}+2\mathbf{E}$	$(4\mathcal{U}+4)\lvert\mathbb{G}\rvert$	$(2\mathcal{S})\lvert\mathbb{G}\rvert$	$(2\mathcal{L})\lvert\mathbb{G}\rvert+\lvert\mathbb{G}_T\rvert$
Sedaghat et al. [18]	$(3\ell+6)\mathbf{E}+2\mathbf{P}$	$(\mathcal{I}+5)\mathbf{E}+5\mathbf{P}$	$(2+\mathcal{U})\lvert\mathbb{G}\rvert+\lvert\mathbb{G}_T\rvert$	$(\mathcal{S}+3)\lvert\mathbb{G}\rvert$	$\lvert\mathbb{G}_T\rvert+(2\mathcal{L}+4)\lvert\mathbb{G}\rvert$
Xiong et al. [20]	$\mathbf{P}+(48\mathcal{S}+5)\mathbf{E}$	$3\mathbf{P}+3\mathbf{E}$	$(11+2y)\lvert\mathbb{G}\rvert$	$(2+5\mathcal{S})\lvert\mathbb{G}\rvert$	$(3+6r)\lvert\mathbb{G}\rvert$
Our scheme	$(\mathcal{U}-\mathcal{L}+3)\mathbf{E}$	$1\mathbf{E}$	$(\mathcal{U}+1)\lvert\mathbb{G}\rvert+\lvert\mathbb{G}_T\rvert$	$\mathbf{1}\lvert\mathbb{G}\rvert$	$2\lvert\mathbb{G}\rvert+\lvert\mathbb{G}_T\rvert$

$|\mathbb{G}|$ and $|\mathbb{G}_T|$ denote the bit-length of elements in base group $\mathbb{G}$ and target group $\mathbb{G}_T$, respectively. $\mathcal{U}$ and $\mathcal{S}$ denote the number of attributes in the universe attribute set and DU attribute set, respectively. $\mathcal{L}$ expresses the number of attributes in the access policy and y is the maximum number of elements in DU's attribute set in the system. $\mathcal{F}$ and $\mathcal{I}$ are the elements of the set of attribute mapping functions and the number of attributes used for decryption, respectively. ℓ is the bit-length of an access policy for the access matrix X and r is the number of the rows of X. Finally, **P**, **M**, and **E** are pairing, multiplication and exponentiation costs, respectively.

decrypt a ciphertext. The length of public parameters grows linearly with the total number of attributes in the network. As well, a DU only requires to store one single base group's element in the storage, while in the other constructions, this size increases linearly with the number of attributes of the DU. The proposed scheme also introduces only a small and constant ciphertext length that infects the communication overhead.

We will next evaluate the achieved experiments for the proposed MLS-ABAC with some well-known CP-ABE schemes in practice.

7. Our results

In this section, we compare the performance of our proposed MLS-ABAC construction with Bethencourt et al. [6], BSW07 in short, and two recent works in the literature; Nasiraee et al. [15], NA20 in short, and Xiong et al. [20], XZPZY20 in short. We have chosen to evaluate our work with the BSW07 scheme because it is widely used in practical implementations. Moreover, as we mentioned before (cf. Section 1), BSW07 can be employed in an IoT context if it uses up to 10 attributes in ciphertexts. In this section, we show that our proposed CP-ABE construction is more efficient than BSW07 and has a constant execution time for encryption, decryption, and key generation. We also explain that our proposed CP-ABE is more efficient than the existing IoT-based construction such as BA20 and XZPZY20. Also note that, based on the applications, our construction is based on AND-gate circuits, whereas both NA20 and XZPZY20 schemes are proposed under a stronger Linear Secret Sharing Scheme (LSSS) approach.

We obtained the benchmarks on Ubuntu 20.04.2 LTS with an Intel Core i7-9850H CPU @ 2.60 GHz and 16 GB of memory. We implemented the construction by using the Charm-Crypto framework [43], a Python library for Pairing-based Cryptography.[4] We utilized a super-singular curve "SS512" with a base field size of 512 bits and an embedding degree of 2. Fig. 8 consists of six graphs depicting the following relationships.

- *Total number of Attributes versus Setup time (top left graph)*: This graph displays the time required to generate the parameter of the system over total number of the attributes. The setup time, as an offline phase, in BA20 and XZPZY20 schemes increases linearly with the number of attributes, while BSW07 has a constant computational cost; our proposed construction has less complexity than BA20 and XZPZY20 and insignificantly more than BSW07. It is worth noting that this phase can be run offline and when considering for example a generous number of 1000 attributes, it only requires ∼250 milliseconds (msec).
- *Number of attributes versus secret decryption key size (top center graph)*: In this graph if we look at the size of the secret keys over the total number of attributes owned by the users. We can see that this relationship is linear for BSW07, NA20 and XZPZY20 schemes. However our proposed scheme requires a quasi constant storage. For instance, in our scheme the required memory for secret keys is ∼1.3 kilobytes (kbytes) for 1000 attributes which is significantly lower the other schemes.
- *Number of attributes versus key generation time (top right graph)*: This graph shows the relationship between the number of attributes of each user and the time to generate decryption key. As can be seen, this time for our scheme is constant – equal to ∼1 msec – while it grows linearly for the other schemes over the number of the attributes.
- *Number of attributes versus Encryption time (bottom right graph)*: This graph displays the relationship between the number of embedded attributes in the policy and the Encryption time. As can be seen, the time required to encrypt a ciphertext in our scheme is constant (less than ∼20 msec), while in the other schemes, it grows linearly.
- *Number of attributes versus ciphertext size (bottom left graph)*: This graph displays the relationship between length of ciphertext in link between sender and the cloud and the number of attributes in the access policy. As it is illustrated, our scheme achieves a constant size of ciphertext (less than ∼230 bytes) while in other schemes, it grows linearly with the number of attributes in the access policy. Noted this result is primarily due to the circuit level applied to the constructions.
- *Number of attributes versus Decryption time (bottom center graph)*: This graph shows the relationship between the total number of attributes of each user and the decryption time. As can be seen, the time required to decrypt a ciphertext in all three decryption outsourceable constructions (Ours, BA20 and XZPZY20) is constant and is always less than ∼15 msec, while this overhead grows in BSW07 linearly with the number of attributes in the access policy.

Our MLS-ABAC scheme is lightweight because of two reasons: our proposed lightweight CP-ABE construction and the lightweight ASCON Cipher Suite. With our CP-ABE, the encryption phase is pairing-free and the heavy parts of the decryption phase can be outsourced to the CS (i.e., DU only needs to compute one exponentiation to decrypt a ciphertext). The DU only stores one element of the base group in the storage, regardless of the number of DU's attributes. Additionally, the proposed scheme has a constant ciphertext length, which does not increase with the number of attributes in the access policy. As a result, our proposed CP-ABE, which is used in our MLS-ABAC scheme, is more efficient and has fewer communication overheads than the existing constructions and provides the largest efficiency gain.

In addition to comparing the proposed CP-ABE scheme with other constructions, we would like to point out that our proposed scheme is more efficient than the BSW07 construction [6]. This is important because, based on the results presented in [5], any

[4] The source code can be found at https://github.com/MLSABAC.

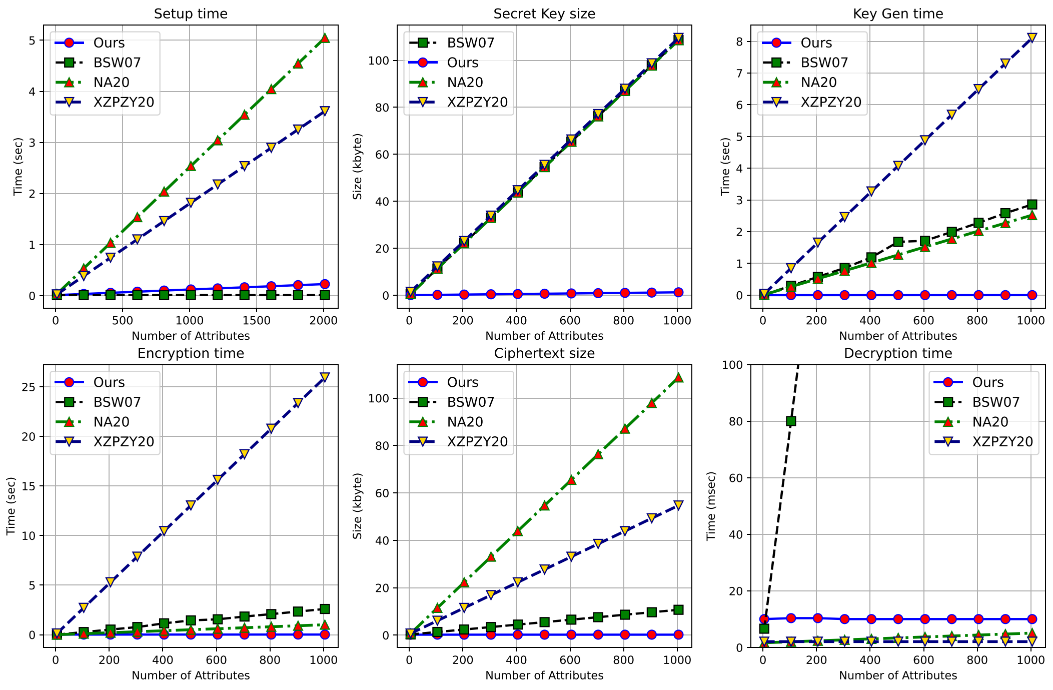


Fig. 8. Performance Analysis of the Proposed MLS-ABAC Scheme.

CP-ABE scheme with better performance than BSW07 can easily be implemented in an IoT context. In [5], the authors analyze the performance of the BSW07 construction, considering the worst-case scenario on two popular IoT platforms, namely the ESP32 [7] and RE-Mote [8]. By leveraging hardware cryptographic acceleration, the CP-ABE scheme in [6] can be used on IoT applications with resource-constrained devices if it employs up to 10 attributes in ciphertexts. According to the above discussion and the information presented in Table 2, both DO and DU have low computation and communication costs. Hence, our proposed MLS-ABAC scheme, based on our CP-ABE construction, as well as a lightweight ASCON Cipher Suite, is efficient enough to be applied in the IoT context.

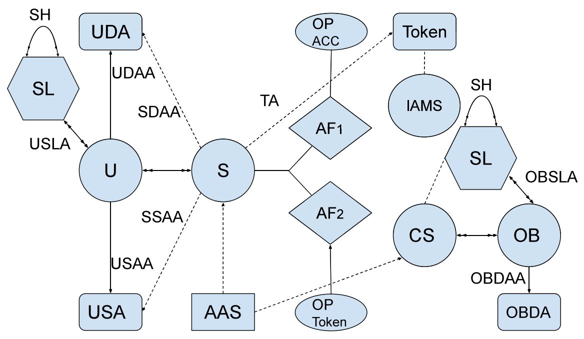


Fig. 9. The Conceptual Model for MLS-ABAC.

8. Conceptual and formal model for MLS-ABAC

In Fig. 9, we illustrate the conceptual MLS-ABAC model followed by formal definitions in Table 3. The access control components involved in this model are Users (U), Subjects (S), Objects (OB), Security levels (SL), Identity and Access Management Services (IAMS), Attribute Authority Services (AAS), Cloud server services (CS), JWT Tokens (T), Operations on IAMS (OP_{ACC}), and Operations on objects in cloud server services (OP_{Token}). User Static Attributes (USA) and User Dynamic Attributes (UDA) are the set of user attributes for users and their associated subjects. Object Dynamic Attributes (OBDA) is the set of dynamic attributes assigned to data objects (OB). These attributes (which are only set-values specified by attType) can be directly assigned to users and objects to values from the set of atomic values (denoted by R(att)). These attributes are same as the dynamic attributes assigned to the subjects (UDA).

The attribute function att_u is the attribute function in USA ∪ UDA that maps a user to a set of values in the power set of $R(att_u)$ and assigns attributes to the users. A similar function (att_{obd}) is defined for the objects associated with cloud services. In our MLS-ABAC model, users and their associated subjects are assigned to multiple security levels, which are denoted by the many to many function directUSL. The security levels hierarchy (SH) (the self-loop on SL), is a partial order relation on SL and presented by $\succeq_{sl}$. The equation $sl_1 \succeq_{sl} sl_2$ denotes that sl_1 is senior to sl_2 and sl_1 inherits all the security properties of sl_2. A token (T) is assigned by the IAMS to a subject based on the witness provided by the subject. This token will have set of security levels, a timestamp and a creator IAMS, which will be assigned to the subject, as denoted by the one to one relation TA. The function security_levels takes as input the subject along with the assigned token, and maps it to a set of security levels for which the subject is allowed to access the objects. Function issued_by defines the IAMS which created the token. A subject s created by the user u (denoted by SubCreator) has a subset of the attribute values of its creator.

In Table 3, two authorization functions of our proposed MLS-ABAC are presented. The first one is the IAMS service token request authorization function ($Auth_{op}(s : S, sr : IAMS)$) where $op \in OP_{ACC}$ and the second one is the data and service authorization function ($Auth_{op}(s : S, cs : CS, ob : OB)$) where $op \in OP_{Token}$ operations. More precisely, for $Auth_{op}(s : S, sr : IAMS)$, a subject $s \in S$ is permitted to perform an operation $op \in OP_{ACC}$ (i.e., receiving the access token) when the security level

Table 3
Formal model definitions for MLS-ABAC.

Basic sets and functions

- U, S, OB, SL, IAMS, AAS and CS are finite sets of users, subjects, objects, security levels, identity and access management, attribute authority services and cloud server services respectively.
- JWT Token (T) are finite set of security tokens provided by AAS to the subject created by user based on witness.
- OP_{ACC} and OP_{Token} are finite sets of operations on IAMS services based on accumulator and objects in CS based on access token respectively.
- USA, UDA and OBDA are finite sets of users' static and dynamic attributes, and objects dynamic attributes respectively.
- For each attribute att in USA ∪ UDA ∪ OBDA, R(att) represents the attributes' range, a finite set of atomic values.
- attType(att) defines the attribute type value which can be set valued only.
- $\forall\ att_u \in USA \cup USDA.\ att_u: U \to 2^{R(att_u)}$ maps users to a set of static and dynamic attribute values.
- $\forall\ att_{obd} \in OBDA.\ att_{obd}: OB \to 2^{R(att_{obd})}$ maps objects a set of dynamic attribute values.
- directUSL: $U \to 2^{SL}$ maps each user to a set of security levels, equivalently USLA $\subseteq$ U $\times$SL.
- SH $\subseteq$ SL $\times$SL, a partial order relation $\succeq_{sl}$ on SL.
- directOBSL: OB $\to$ SL maps each data object to a security level defined in CS, equivalently OBSLA $\subseteq$ OB $\times$SL.
- SH $\subseteq$ SL $\times$SL, a partial order relation $\succeq_{sl}$ on SL.
- T $\subseteq 2^{SL}$ $\times$IAMS $\times$TimeStamp is a set of tokens having security levels, an IAMS which created, and validity timestamp.
- TA $\subseteq$ T $\times$S, a one to one relation for binding a token to subject.
- security_levels : S $\times$T $\to 2^{SL}$, mapping each subject and its assigned token to allowed security levels. Formally, security_levels (s,t) = {sl $\in$ SL | (sl, iams, timestamp) = t $\in$ T $\wedge$ (t,s) $\in$ TA}.
- issued_by : T $\to$ IAMS, mapping a token to its issuer IAMS. Formally, issued_by (t : T) = iams $\in$ IAMS such that (sl, iams, timestamp) = t.
- SubCreator : S $\to$ U, mapping each subject to its creator user.
- $\forall\ att_u \in USA \cup USDA.\ att_u: S \to 2^{R(att_u)}$ maps subjects to a set of static and dynamic attribute values. It is required that : $att_u(s) \subseteq att_u(SubCreator(s))$.
- directUSL: S $\to 2^{SL}$ maps each subject to a set of security levels.
  It is required that : directUSL (s) $\subseteq$ directUSL(SubCreator(s)).

Authorization functions

- IAMS Service Authorization Function: For each op $\in OP_{ACC}$, $Auth_{op}$(s:S, sr:AAS) is a propositional logic formula returning true or false, which is defined using the following policy language:
  - $\alpha ::= \alpha \wedge \alpha \mid \alpha \vee \alpha \mid (\alpha) \mid \neg\alpha \mid \exists x \in set.\alpha \mid \forall x \in set.\alpha \mid set \triangle set \mid atomic \in set \mid atomic \notin set$
  - $\triangle ::= \subset \mid \subseteq \mid \not\subseteq \mid \cap \mid \cup$
  - set ::= $att_u(s)$ | directUSL(s) for $att_u \in USA \cup UDA$
  - atomic ::= value
- Cloud Service Authorization Function: For access operation on cloud service cs from subject s using the token t, $Auth_{access}$(s:S, t:T, cs:CS) is a propositional logic formula returning true or false and defined using the above policy language stated above with the following changes:
  - set ::= $att_u(s)$ | issued_by(t) | security_levels(s,t) for $att_u \in USA \cup UDA$
  - atomic ::= value
- Data Objects Authorization Function: For each op $\in OP_{Token}$ on object ob $\in$ OB belonging to cloud server service cs $\in$ CS, $Auth_{op}$(s:S, cs:CS, ob:OB) is a propositional logic formula returning true or false and defined using the above policy language stated above with the following changes:
  - set ::= $att_u(s)$ | $att_{obd}(ob)$ | directOBSL(ob) for $att_u \in USA \cup UDA$, for $att_{obd} \in OBDA$
  - atomic ::= value

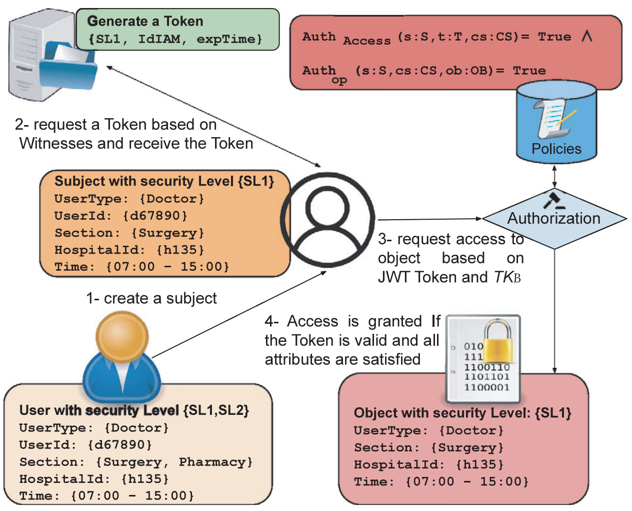
Access operation decision

- A subject $s \in S$ is allowed to perform an operation op $\in OP_{ACC}$ to receive an access token which is created by a service sr $\in$ IAMS, if the policies stated in the authorization function, i.e the security levels of the subject assigned by the AAS as a part of witness and the accumulator parameters in IAMS satisfy $Auth_{op}$(s:S, sr:IAMS) . Formally, $Auth_{op}$(s:S, sr:IAMS) = True.
- A subject s $\in$ S is permitted to perform an operation op $\in OP_{Token}$ on an object ob $\in$ OB in cloud server service cs $\in$ CS with a token t $\in$ T, if subject s is permitted to access services cs based on the Token t and the attributes of subject and object satisfy the required policies. Formally, $Auth_{access}$(s:S, t:T, cs:CS) $\wedge$ $Auth_{op}$(s:S, cs:CS, ob:OB) = True.

in IAMS is satisfied (i.e., the accumulator parameters from the subject and its security level satisfied the policies stated in IAMS). Similarly, for $Auth_{op}$(s : S, cs : CS, ob : OB), a subject s $\in$ S is permitted to perform an operation op $\in OP_{Token}$ (i.e., accessing to the object ob $\in$ OB) when the security level of the object in CS is satisfied (i.e., $Auth_{access}$(s : S, t : T, cs : CS) = True, meaning the security level stated in the token satisfied the policies stated in CS) and the attributes of s $\in$ S satisfied the attributes of object ob. Thus, a subject s is permitted to perform an operation on an object ob in a service cs iff $Auth_{receive\_token}$(s : S, iams : IAMS), $Auth_{access}$(s : S, t : T, cs : CS) and $Auth_{op}$(s : S, cs : CS, ob : OB) returns True.

8.1. Use-case and application for MLS-ABAC

An example MLS-ABAC configuration is illustrated in Fig. 10. Here, a user who wants to access the data creates a subject (here, DU) with the subset of the user's attributes. Then, DU sends the token request along with the witness to the IAMS and receives the token TK_{it} including the user's security level. At this point, DU generates the token $tk_{\mathbb{B}}$ based on his attributes and sends it to the CS along with the token TK_{it}. In response, if these tokens can pass both the authentication and authorization steps, CS partially decrypts the ciphertext and returns this value to the DU for decryption. Finally, DU who has already generated the token $tk_{\mathbb{B}}$



Generate a Token
{SL1, IdIAM, expTime}
Auth Access (s:S,t:T,cs:CS)= True ∧
Auth op (s:S,cs:CS,ob:OB)= True
Policies
2- request a Token based on Witnesses and receive the Token
Subject with security Level {SL1}
UserType: {Doctor}
UserId: {d67890}
Section: {Surgery}
HospitalId: {h135}
Time: {07:00 - 15:00}
Authorization
3- request access to object based on JWT Token and TKB
1- create a subject
4- Access is granted If the Token is valid and all attributes are satisfied
User with security Level {SL1,SL2}
UserType: {Doctor}
UserId: {d67890}
Section: {Surgery, Pharmacy}
HospitalId: {h135}
Time: {07:00 - 15:00}
Object with security Level: {SL1}
UserType: {Doctor}
Section: {Surgery}
HospitalId: {h135}
Time: {07:00 - 15:00}

Fig. 10. Access Request Flow Example (comply with the NIST's ABAC model).

can decrypt the partially decrypted message and access the data. The details of this example that explains the policies $\mathtt{Auth}_{\text{access}}$ and $\mathtt{Auth}_{\text{op}}$ are as follows.

A user Bob has attributes `UserType(Bob) = {Doctor}`, `UserId(Bob) = {d67890}`, `Section(Bob) = {Surgery, Pharmacy}`, `HospitalId(Bob) = {h135}`, and `Time(Bob) = {07:00--15:00}`. An initialized subject S has attributes which are the subset of Bob attributes who is its creator. In this subset, the dynamic attribute `Section` only includes value `'Surgery'` meaning this subject only can decrypt the encrypted data (read objects) that are wrapped with the dynamic attribute `Section` with value `'Surgery'` and `Time` with value {07:00–15:00}. After creating the subject, it is permitted to receive the access token `'t'` from IAMS if the subject can prove its membership in at least one security level (SL). We have a security hierarchy SH, a total order relation on SL such that `SH = {(SL1, SL2), (SL2, SL3), (SL3, SL4)}`.

A subject is permitted to download, decrypt and perform an operation op on an object ob in cloud server service cs if the policy $\mathtt{Auth}_{\text{access}}\mathtt{(s:S,\ t:T,\ cs:CS)=True} \wedge \mathtt{Auth}_{\text{op}}\mathtt{(s:S,\ cs:CS,\ ob:OB)=True}$ is satisfied meaning that the subject s is permitted to access services cs based on the token 't' and the effective attributes of subject s (the subject that has static and dynamic attributes can generate $\mathrm{Tk}_{\mathbb{B}}$ token), attributes of cloud server services (the security level), and the attributes of object ob (both static and dynamic attributes). For example, a policy $\mathtt{Auth}_{\text{access}}\mathtt{(s:S,\ t:T,\ cs:CS)} \equiv \mathtt{\{SL2,\ SL3,\ SL4\}} \in \mathtt{security\_levels(s,\ t)} \wedge \mathtt{issued\_by(t)} = \mathtt{iams} \in \mathtt{IAMS}$, meaning that the subject received the valid token with security levels SL2, SL3, and SL4 which was created by an IAMS, and can access the wrapped data objects stored in SL2, SL3, and SL4 levels of the cloud service cs, i.e all data objects ob such that directOBSL(ob) is equal to SL2, SL3 or SL4. Another policy $\mathtt{Auth}_{\text{read}}\mathtt{(s:S,\ cs:CS,\ ob:OB)} \equiv \mathtt{Auth}_{\text{access}}\mathtt{(s:S,\ t:T,\ cs:CS)=\ True} \wedge \mathtt{\{07:00\text{--}15:00\}} \in \mathtt{Time(s)} \wedge \mathtt{Surgery} \in \mathtt{Section(s)} \wedge \mathtt{Doctor} \in \mathtt{UserType(s)} \wedge \mathtt{h135} \in \mathtt{HospitalId(s)} = \mathtt{True}$, determines that read operation on medical data (decryption) is allowed by a subject s on an object ob in service cs if first, subject s allowed to access wrapped data objects stored in the security levels SL2, SL3, and SL4 (i.e., $\mathtt{Auth}_{\text{access}}\mathtt{(s:S,\ t:T,\ cs:CS)=\ True}$) and then if the attributes `Time`, `Section`, `UserType`, and `Hospital` of the accessing user have values `07:00--15:00`, `Surgery`, `Doctor`, and `h135`, respectively.

9. Conclusion

In this paper, we proposed a novel outsourceable MLS-ABAC scheme based on both static and dynamic user attributes in combination with the assigned security level that satisfies the requirements of NIST's ABAC model. Our proposed ABE construction is based on CP-ABE with a constant decryption key size. Adding a security level verification before partial decryption facilitates the management of the access control and improves the security and privacy of IoT systems. Using lightweight functions instead of heavyweight functions, such as lightweight Ascon-hash and Ascon authenticated encryption functions, as well as our proposed lightweight CP-ABE, makes MLS-ABAC efficient. Besides considering security level verification and dynamic attributes, MLS-ABAC also protects the data integrity of the plaintext in case of outsourced decryption, using an authenticated encryption scheme. Moreover, to demonstrate the use-case and application of our proposed MLS-ABAC scheme, we formalize our proposed access control model by providing a conceptual and formal model as well as performance metrics. The results of our implementation and a brief comparison of proposed scheme performance with predecessor work provide an excellent incentive to use MLS-ABAC in some real-world IoT scenarios. We leave the implementation of our proposed MLS-ABAC on an IoT platform as future work. Another open question is the construction of an efficient MLS-ABAC under a general Boolean relation.

CRediT authorship contribution statement

Seyed Farhad Aghili: Conceptualization, Methodology, Software, Validation, Investigation, Evaluation, Writing – original draft, Writing – review & editing. **Mahdi Sedaghat:** Conceptualization, Methodology, Software, Validation, Investigation, Evaluation, Formal analysis, Writing – original draft, Writing – review & editing. **Dave Singelée:** Methodology, Validation, Investigation, Writing – original draft, Writing – review & editing. **Maanak Gupta:** Conceptualization, Validation, Investigation, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: This work was financed in part by the European Union's Horizon 2020 Research and innovation program, under grant agreement No. 826284 (ProTego).

Acknowledgments

We would like to thank anonymous reviewers for their valuable comments. This work was supported in part by the European Union's Horizon 2020 Research and Innovation Program, under Grant 826284 (ProTego). This work was supported in part by the Research Council KU Leuven C1 on Security and Privacy for Cyber-Physical Systems and the Internet of Things with contract number C16/15/058 and by CyberSecurity Research Flanders with reference number VR20192203.

Appendix. Security proofs

A.1. Proof of Theorem 1

Proof. To prove the correctness of the proposed CP-ABE, we demonstrate that a receiver with an authorized set of attributes $\mathbb{B} \in \Sigma_k$ can retrieve message encrypted under access policy $\mathbb{P} \in \Sigma_c$ iff $\mathbb{P} \subseteq \mathbb{B}$.

$$m' := C \cdot \mathsf{pd}_{\mathbb{B}}^{1/\mu} = C \cdot ((C_1' \bullet V)(\mathsf{tk}_{\mathbb{B}} \bullet C_2))^{-1/\mu\ell_0} = C \cdot (([-r\mu s^2]_1 \bullet [(L_{\mathbb{P},\mathbb{B}}(s) - \ell_0)/s]_2) \cdot ([\mu/Z_{\mathbb{B}}(s)]_1 \bullet [rsZ_{\mathbb{P}}(s)]_2))^{(-\mu\ell_0)^{-1}} = C \cdot ([rs\mu(l_0 - L_{\mathbb{P},\mathbb{B}}(s))]_T \cdot [rs\mu L_{\mathbb{P},\mathbb{B}}(s)]_T)^{(-\mu\ell_0)^{-1}} = C \cdot [rs\ell_0\mu]_T^{-1/\mu\ell_0} = m \cdot [rs]_T \cdot [-rs]_T = m \quad \square \qquad (A.1)$$

A.2. Proof of Theorem 2

Proof. We plan to prove this theorem by the reduction. Let there exists a Probabilistic Polynomial Time (PPT) adversary, $\mathcal{A}$, who can break the proposed scheme in the introduced security game in Definition 8 with a non-negligible advantage of ϵ. Then we are going to show that how a PPT adversary $\mathcal{B}$ can solve (l, m, t)-MSE-DDH problem defined in 4 with a non-negligible advantage of at least $\frac{\epsilon}{2}$. In fact, $\mathcal{B}$ takes on the role of the challenger and utilizes the adversary $\mathcal{A}$ in order to solve the mentioned hard problem.

Let the challenger $\mathcal{C}$ of Decisional (l, m, t)-MSE-DDH hard assumption runs the asymmetric bilinear group generator $\mathcal{BG}(\lambda)$ for the security parameter λ and takes $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathsf{p}, \hat{e})$ such that $[1]_1$, $[1]_2$ and $[1]_T$ be the generators of the defined cyclic groups. The challenger $\mathcal{C}$ first chooses three integers l, m, t, along with two univariate coprime polynomials f and h of degree l and m with pairwise distinct roots $\vec{x} = (x_1, \ldots, x_l)$ and $\vec{y} = (y_1, \ldots, y_m)$. It samples integers $\alpha, \delta k \leftarrow\$ \mathbb{Z}_p^*$ uniformly at random and then flips a fair coin, $\beta \leftarrow\$ \{0, 1\}$, outside $\mathcal{B}$'s view. If $\beta = 0$, $\mathcal{C}$ sets $\Gamma = [kf(\alpha)]_T$, otherwise, it sets $\Gamma = R$, where R is a random element of the target cyclic group $\mathbb{G}_T$. The challenger $\mathcal{C}$ sends Γ and the pair of vectors $\vec{x}$ and $\vec{y}$ along with $\vec{v}_1 = ([1]_1, [\alpha]_1, [\alpha^2]_1, \ldots, [\alpha^{l+t-2}]_1, [k\alpha f(\alpha)]_1)$, $\vec{v}_2 = ([\delta]_1, [\delta\alpha]_1, [\delta\alpha^2]_1, \ldots, [\delta\alpha^{l+t}]_1)$, $\vec{v}_3 = ([1]_2, [\alpha]_2, [\alpha^2]_2, \ldots, [\alpha^{m-2}]_2)$ and $\vec{v}_4 = ([\delta]_2, [\delta\alpha]_2, [\delta\alpha^2]_2, \ldots, [\delta\alpha^{2m-1}]_2, [kh(\alpha)]_2)$ to the adversary $\mathcal{B}$.

Initialization: In this phase, the simulator $\mathcal{B}$ sets the universe attribute set of $\mathbb{U}$ as the all possible attributes in the defined network and a collision-resistant Hash function $\mathsf{H} \leftarrow\$ \mathcal{H}$. Then $\mathcal{B}$ publishes $\mathbb{U}$ and receives back the challenge access policy $\mathbb{P}^*$ along with the query set Q as a group of attribute sets $\mathbb{B}_i \subseteq \mathbb{U}$ for $i \leq s$, such that $|\mathbb{B}_i| \leq e$ from $\mathcal{A}$ and also $\mathbb{P}^* \not\subseteq \{\mathbb{B}_i\}_{i=1}^s$ (i.e., $\mathsf{Bf}(\mathbb{B}_i, \mathbb{P}^*) = 0$ for $1 \leq i \leq s$. At this point, $\mathcal{B}$ computes the following polynomial that is the multiplication of zero-polynomials corresponding for the chosen subsets $\mathbb{B}_i$ in the query set Q.

$$Y_Q(x) = \prod_{i=1}^{s} Z_{\mathbb{B}_i}(x) = \prod_{i=1}^{s}\left(\prod_{j=1}^{n-e}(x - k_j)^{\overline{b_i[j]}}\right). \qquad (A.2)$$

Where $b_i[j]$ represents the jth binary representation of subset $\mathbb{B}_i$. The degree of univariate polynomial $Y_Q(x)$ is bounded by $s(n-e)$. Moreover, since $\mathcal{B}$ knows $h(x) = \prod_{i=0}^{m}(x - y_i)$, it is assumed $Z_{\mathbb{P}^*}(x) = Y_Q(x)h(x)$ such that $|\mathbb{P}^*| = n - (s(n-e) + m)$. This can be feasible by defining hash function H in the Random Oracle Model. Although the generator of the first cyclic group is not public, she assumes $G = [f(\alpha)Y_Q(\alpha)]_1$ as a new generator for the cyclic group $\mathbb{G}_1$. To generate g_2 under the new generator it computes $f(x)Y_Q(x) = \sum_{j=0}^{l+s(n-e)} p_j x^j$ and we have, $g_2 = \left(\prod_{j=0}^{l+s(n-e)} [\alpha^{j+2}]_1^{p_j}\right) = [f(\alpha)Y_Q(\alpha)]_1^{\alpha^2} = G^{\alpha^2}$.

Then $\mathcal{B}$ defines $h_i = [\alpha^i]_2$ for $0 \leq i \leq n$. Finally, the public parameters based on the new generator G are $\mathsf{pp} = \{\{h_i\}_{i=0}^n, g_2, [\alpha f(\alpha)Y_Q(\alpha)]_T, \mathsf{H}\}$. While she keeps secure the master secret key $\mathsf{msk} = \{G\}$ by herself.

1st Query phase. The adversary $\mathcal{A}$ after receiving the public parameters has access to the following oracles for a polynomially bounded number of queries.

Simulating the $\mathcal{O}_{\mathsf{KGen}}(\mathbb{B}_i)$ oracle. The adversary $\mathcal{A}$ has access to this oracle which is provided by $\mathcal{B}$, to receive private decryption key corresponding to the attribute set $\mathbb{B}_i \in Q$. In this end, $\mathcal{B}$ calculates the univariate polynomial $\Lambda_i(x)$ to simulate secret decryption key, such that $f(x)Y_Q(x) = \Lambda_i(x) \cdot Z_{\mathbb{B}_i}(x)$. Based on the definition of polynomial $Y_Q(x)$, we know it is divisible by $Z_{\mathbb{B}_i}(x)$ and we can rewrite the above equation as $\Lambda_i(x) = (f(x)Y_Q(x))/Z_{\mathbb{B}_i}(x)$. Since the polynomial $\Lambda_i(x)$ is not rational then we can take the coefficients in the standard basis as $\Lambda_i(x) = \sum_{j=0}^{q} \lambda_j x^j$ and $f(x)Y_Q(x) = \sum_{j=0}^{l+s(n-e)} p_j x^j$. Finally, the challenger returns the following equation as the simulated secret decryption key $\mathsf{dk}_{\mathbb{B}_i}$ corresponding to key index $\mathbb{B}_i$.

$$\mathsf{dk}_{\mathbb{B}_i} = \prod_j [\alpha^j]_1^{\lambda_j} = [f(\alpha)Y_Q(\alpha)]_1^{1/Z_{\mathbb{B}_i}(\alpha)} = G^{1/Z_{\mathbb{B}_i}(\alpha)}. \qquad (A.3)$$

Simulating the $\mathcal{O}_{\mathsf{Enc}}(m, \mathbb{P})$ oracle. The adversary $\mathcal{A}$ can adaptively request to encrypt arbitrary messages from the message space $\mathcal{M}$ under a certain access structure $\mathbb{P}$. The challenger $\mathcal{B}$ samples a random integer $r \leftarrow\$ \mathbb{Z}_p^*$, uniformly and computes the following equations and sends back the tuple $\mathsf{Ct} = (\mathbb{P}, C, C_1, C_2)$ to $\mathcal{A}$.

$$(C, C_1, C_2) = \left(m[r\alpha f(\alpha)Y_Q(\alpha)]_T, g_2^{-r}, [r\alpha, Z_{\mathbb{P}}(\alpha)]_2\right). \qquad (A.4)$$

The only condition is that $(m-2) \geq n - |\mathbb{P}| + 1$, i.e., $|\mathbb{P}| \geq n - m + 3$.

Simulating the $\mathcal{O}_{\mathsf{Tgen}}(\mathsf{dk}_{\mathbb{B}}, C)$ oracle. It takes a correctly generated decryption key $\mathsf{dk}_{\mathbb{B}}$ and the cipher value C as inputs. It selects a random integer $\mu \leftarrow\$ \mathbb{Z}_{\mathsf{p}}^*$. It returns $\mathsf{tk}_{\mathbb{B}} = \mathsf{dk}_{\mathbb{B}}^{\mu}$ along with $C_1' = C_1^{\mu}$.

Simulating the $\mathcal{O}_{\mathsf{PDec}}(\mathsf{tk}_{\mathbb{B}}, \mathsf{Ct})$ oracle. It takes a valid token pair $(\mathsf{tk}_{\mathbb{B}}, C_1')$ and underlying ciphertext Ct encrypted under policy $\mathbb{P}$. If $\mathsf{Br}(\mathbb{B}, \mathbb{P}) = 1$, by assuming $c[j] = b[j] - p[j]$, it computes the univariate polynomial $L_{\mathbb{P},\mathbb{B}} = \prod_{j=1}^{n}(x - k_j)^{c[j]} = \sum_{i=0}^{n} l_i x^i$ and $V = \prod_{i=1}^{n} h_{i-1}^{l_i}$ and returns $\mathsf{pd}_{\mathbb{B}} = \left((C_1' \bullet V)(\mathsf{tk}_{\mathbb{B}} \bullet C_2)\right)^{1/l_0} = [-rs\mu]_T$ along with C, else it responses by $\perp$.

Simulating the $\mathcal{O}_{\mathsf{Dec}}(\mathsf{pd}_{\mathbb{B}}, C; \mu)$ oracle. The adversary $\mathcal{A}$ has access to this oracle to receive the full decryption of Ct by providing partially decrypted value $\mathsf{pd}_{\mathbb{B}}$. It returns $m' = C \cdot \mathsf{pd}_{\mathbb{B}}^{1/\mu}$ as output.

Challenge: The adversary $\mathcal{A}$ chooses two same length plaintexts $\{m_0, m_1\} \leftarrow\$ \mathcal{M} \times \mathcal{M}$ and sends them to $\mathcal{B}$. Then $\mathcal{B}$ flips a fair coin to have the biased bit $b \leftarrow\$ \{0, 1\}$, and computes the challenge ciphertext $\mathsf{Ct}^* = (\mathbb{P}^*, C^*, C_1^*, C_2^*)$ as follows,

$$C^* = m_b\Gamma, C_2^* = g_2^{-r}, C_2^* = [kh(\alpha)]_2. \qquad (A.5)$$

It is assumed $r^* = k/(\alpha Y_Q(\alpha))$ as the randomness for the challenge ciphertext. To put it in a nutshell, based on (l, m, t)-MSE-DDH assumption, there are two cases for the received challenge Γ with the same probability 1/2. If $\Gamma = [kf(\alpha)]_T$ then $C^* = m_b[kf(\alpha)]_T = m_b[r^*\alpha f(\alpha)Y_Q(\alpha)]_T$ and also $C_2^* = [kh(\alpha)]_2 = [r^*\alpha Y_Q(\alpha)h(\alpha)]_2 = [r^*\alpha Z_{\mathbb{P}^*}(\alpha)]_2$ are in the correct format. While in the case of an independent and random element in the group $\mathbb{G}_T$, the computed C^* is a random element out of the construction and the adversary can distinguish by chance.

2nd Query phase. The adversary $\mathcal{A}$ after receiving the challenge ciphertext Ct* has access to the defined queries in the first phase on the condition that she cannot query the partial decryption and decryption oracles.

Guess. Afterwards, $\mathcal{A}$ returns back a bit b'. Let β' be the value that is guessed by $\mathcal{B}$ about the value of β. If $b' = b$, $\mathcal{B}$ outputs $\beta' = 0$, i.e, $\Gamma = [kf(\alpha)]_T$. Otherwise, $\mathcal{B}$ outputs $\beta' = 1$, which indicates that it receives a random element in the cyclic group $\mathbb{G}_T$. When $\beta = 1$, the adversary $\mathcal{A}$ obtains no information about b. So she can guess by chance and we have $\Pr[b' = b \mid \beta = 1] = 1/2$. On the other hand, when $b' \neq b$, the guess value by the adversary $\mathcal{B}$ is $\beta' = 1$, so we have $\Pr[\beta' = \beta \mid \beta = 1] = 1/2$. Particularly, if $\beta = 0$, $\mathcal{A}$ can distinguish with a non-negligible advantage ϵ because she has received the true format of the ciphertext for the message m_b. Thus, we have $\Pr[b' = b \mid \beta = 0] \geq \epsilon + 1/2$. As adversary $\mathcal{B}$ correctly guesses the value of β when $\beta = 0$, we have $\Pr[\beta' = \beta \mid \beta = 0] \geq \epsilon + 1/2$. Therefore, the overall advantage of the adversary $\mathcal{B}$ in solving the (l, m, t)-MSE-DDH problem is,

$$\begin{aligned} &Adv_{\mathcal{B}}^{\mathtt{MSE-DDH}} = \Pr[\beta = 0]\Pr[\beta' = \beta \mid \beta = 0] + \\ &\Pr[\beta = 1]\Pr\left[\beta' = \beta \mid \beta = 1\right] - 1/2 \\ &\geq 1/2\,(\epsilon + 1/2) + (1/2 \cdot 1/2) - 1/2 \geq \frac{\epsilon}{2}\ . \end{aligned} \tag{A.6}$$

Therefore, the adversary $\mathcal{B}$ can play the (l, m, t)-MSE-DDH game with non-negligible advantage $\frac{\epsilon}{2}$. By contradiction, since we know there is no PPT adversary $\mathcal{B}$ to break the (l, m, t)-MSE-DDH assumption with a non-negligible advantage, then the proposed CP-ABE scheme in 4.2 is secure in the IND-CPA game. □

References

[1] S.F. Aghili, et al., LACO: LIghtweight three-factor authentication, access control and ownership transfer scheme for e-health systems in IoT, Future Gener. Comput. Syst. 96 (2019) 410–424.

[2] R.S. Sandhu, et al., Role-based access control models, Computer 29 (2) (1996) 38–47.

[3] V.C. Hu, D. Ferraiolo, R. Kuhn, A.R. Friedman, A.J. Lang, M.M. Cogdell, A. Schnitzer, K. Sandlin, R. Miller, K. Scarfone, et al., Guide to attribute based access control (abac) definition and considerations (draft), NIST Spec. Publ. 800 (162) (2013) 1–54.

[4] D.E. Bell, L.J. La Padula, Secure Computer System: Unified Exposition and Multics Interpretation, Tech. rep. Mitre Corp Bedford Ma, 1976.

[5] P. Perazzo, F. Righetti, M. La Manna, C. Vallati, Performance evaluation of attribute-based encryption on constrained iot devices, Comput. Commun. 170 (2021) 151–163.

[6] J. Bethencourt, et al., Ciphertext-policy attribute-based encryption, in: 2007 IEEE Symposium on Security and Privacy (SP'07), IEEE, 2007, pp. 321–334.

[7] Espressif ESP32 datasheet, 2020, https://bit.ly/2qW8yj1, accessed: 2020-01-15.

[8] Zolertia RE-mote datasheet, 2020, https://bit.ly/2OkilYY, accessed: 2020-01-15.

[9] A. Sahai, B. Waters, Fuzzy identity-based encryption, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2005, pp. 457–473.

[10] V. Goyal, et al., Attribute-based encryption for fine-grained access control of encrypted data, in: Proc. of CCS'06, 2006, pp. 89–98.

[11] J. Bethencourt, et al., Ciphertext-policy attribute-based encryption, in: 2007 IEEE Symposium on Security and Privacy (SP'07), IEEE, 2007, pp. 321–334.

[12] L. Cheung, C. Newport, Provably secure ciphertext policy ABE, in: Proc. of CCS'07, 2007, pp. 456–465.

[13] B. Waters, Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization, in: PKC '14, Springer, 2011, pp. 53–70.

[14] K. Yang, et al., Attribute based encryption with efficient revocation from lattices, IJ Netw. Secur. 22 (1) (2020) 161–170.

[15] H. Nasiraee, M. Ashouri-Talouki, Anonymous Decentralized Attribute-Based Access Control for Cloud-Assisted Iot, FGCS, 2020.

[16] H. Cui, et al., Key regeneration-free ciphertext-policy attribute-based encryption and its application, Inform. Sci. 517 (2020) 217–229.

[17] G. Wang, et al., Hierarchical attribute-based encryption for fine-grained access control in cloud storage services, in: Proc. of ACM'17 on Computer and communications security, 2010, pp. 735–737.

[18] S.M. Sedaghat, et al., An efficient and secure data sharing in smart grid: Ciphertext-policy attribute-based signcryption, in: 2017 Iranian Conference on Electrical Engineering (ICEE), IEEE, 2017, pp. 2003–2008.

[19] J. Li, et al., Full verifiability for outsourced decryption in attribute based encryption, IEEE Trans. Serv. Comput. (2017).

[20] H. Xiong, et al., Partially policy-hidden attribute-based broadcast encryption with secure delegation in edge computing, vol. 97, FGCS, 2019, pp. 453–461.

[21] Z. Li, S. Huan, Multi-level attribute-based encryption access control scheme for big data, in: MATEC Web of Conferences, 173, EDP Sciences, 2018, pp. 03047.

[22] N. Kaaniche, M. Laurent, Attribute based encryption for multi-level access control policies, in: SECRYPT 2017: 14th International Conference on Security and Cryptography, vol. 6, Scite Press, 2017, pp. 67–78.

[23] M. Nabeel, E. Bertino, Privacy preserving delegated access control in public clouds, IEEE Trans. Knowl. Data Eng. 26 (9) (2013) 2268–2280.

[24] M. Gupta, R. Sandhu, The $GURA_G$ administrative model for user and group attribute assignment, in: Proc. of NSS'10, Springer, 2016, pp. 318–332.

[25] S. Bhatt, et al., Abac with group attributes and attribute hierarchies utilizing the policy machine, in: Proc. of ABAC Workshop, ACM, 2017, pp. 17–28.

[26] N. I. of Standards, Technology, Policy Machine, Tech. rep., U.S. Department of Commerce, Washington, D.C.

[27] D. Ferraiolo, et al., The policy machine: A novel architecture and framework for access control policy specification and enforcement, J. Syst. Archit. 57 (4) (2011) 412–424.

[28] D.F. Ferraiolo, et al., Policy Machine: Features, Architecture, and Specification, Tech. rep., 2015.

[29] M. Nabeel, N. Shang, E. Bertino, Privacy preserving policy-based content sharing in public clouds, IEEE Trans. Knowl. Data Eng. 25 (11) (2012) 2602–2614.

[30] P. Pandiaraja, P. Vijayakumar, V. Vijayakumar, R. Seshadhri, Computation efficient attribute based broadcast group key management for secure document access in public cloud., J. Inf. Sci. Eng. 33 (3) (2017).

[31] M. Sedaghat, B. Preneel, Cross-domain attribute-based access control encryption, in: M. Conti, in: M. Stevens, S. Krenn (Eds.), Cryptology and Network Security, Springer International Publishing, Cham, 2021, pp. 3–23.

[32] P.V.X. Tran, T.N. Dinh, A. Miyaji, Efficient ciphertext-policy ABE with constant ciphertext length, in: 2012 7th International Conference on Computing and Convergence Technology (ICCCT), IEEE, 2012, pp. 543–549.

[33] F. Guo, Y. Mu, W. Susilo, D.S. Wong, V. Varadharajan, CP-ABE with constant-size keys for lightweight devices, IEEE Trans. Inf. Forensics Secur. 9 (5) (2014) 763–771.

[34] S. Namasudra, P. Roy, P. Vijayakumar, S. Audithan, B. Balusamy, Time efficient secure dna based access control model for cloud computing environment, Future Gener. Comput. Syst. 73 (2017) 90–105.

[35] C. Dobraunig, et al., Ascon v1. 2 submission to nist, NIST round 2, 2019, https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/ascon-spec-round2.pdf.

[36] M. Jones, et al., Json Web Token (Jwt), Tech. Rep., 2015.

[37] D. Boneh, M. Franklin, Identity-based encryption from the weil pairing, in: Annual International Cryptology Conference, Springer, 2001, pp. 213–229.

[38] C. Delerablée, D, Pointcheval, dynamic threshold public-key encryption, in: Annual International Cryptology Conference, Springer, 2008, pp. 317–334.

[39] A. Beimel, et al., Secure Schemes for Secret Sharing and Key Distribution, Technion-Israel Institute of technology, Faculty of computer science, 1996.

[40] G. Bertoni, et al., Sponge functions. Ecrypt Hash Workshop, 2007.

[41] C. Zhang, R. Green, Communication security in internet of thing: preventive measure and avoid ddos attack over iot network, in: Proceedings of the 18th Symposium on Communications & Networking, 2015, pp. 8–15.

[42] J.-X. Hu, et al., An intelligent and secure health monitoring scheme using iot sensor based on cloud computing, J. Sens. 2017 (2017).

[43] J.A. Akinyele, et al., Charm: a framework for rapidly prototyping cryptosystems, J. Cryptogr. Eng. 3 (2) (2013) 111–128.

Seyed Farhad Aghili received his Ph.D. degree in Information Technology Engineering from the Faculty of Computer Engineering, University of Isfahan, in 2019. He received his Master's degree in Electrical Engineering from Shahid Rajaee Teacher Training University (SRTTU) in 2013. In July 2018, he joined to SPRITZ Security & Privacy Research Group at the University of Padua as a visiting Ph.D. researcher. From April 2019 to January 2020, he was an exchange Ph.D. student at Computer Networks and Telematics, Institute of Computer Science, Faculty of Engineering, University of Freiburg. He is currently a Post Doctoral Research Fellow, COSIC, KU Leuven, Belgium. His current research interests include access control models, authentication protocols, applied cryptography, and IoT systems security.

Mahdi Sedaghat is a third year Ph.D. student at imec-Cosic, KU Leuven, Leuven, Belgium. He received his master's degree in Electrical Engineering with a major in secure telecommunication and cryptography from the Sharif University of Technology, Tehran, Iran. His research involves privacy-preserving distributed systems and Blockchain technology, and he is especially interested in Zero-Knowledge proofs.

Dr. Ir. Dave Singelée received the Master's degree of Electrical Engineering and a Ph.D. in Applied Sciences in 2002 and 2008 respectively, both from KU Leuven (Belgium). He is currently an industrial research manager (IOF) at the research group COSIC. His main research interests are cryptography, security and privacy of wireless communication networks, key management, distance bounding, cryptographic authentication protocols, and security solutions for medical devices.

Maanak Gupta is an Assistant Professor in Computer Science at Tennessee Technological University, Cookeville, USA. He received M.S. and Ph.D. in Computer Science from the University of Texas at San Antonio (UTSA) and has also worked as a postdoctoral fellow at the Institute for Cyber Security (ICS) at UTSA. His primary area of research includes security and privacy in cyber space focused in studying foundational aspects of access control and their application in technologies including cyber physical systems, cloud computing, IoT and Big Data. He has worked in developing novel security mechanisms, models and architectures for next generation smart cars, smart cities, intelligent transportation systems and smart farming. He is also interested in machine learning based malware analysis and AI assisted cyber security solutions. His research has been funded by the US National Science Foundation (NSF), NASA, US Department of Defense (DoD) and private industry. He holds a B.Tech degree in Computer Science and Engineering, from India and an M.S. in Information Systems from Northeastern University, Boston, USA.