# Attribute based encryption for multi-level access control policies

Nesrine Kaaniche, Maryline Laurent

# Attribute based Encryption for Multi-level Access Control Policies

Nesrine Kaaniche[1], Maryline Laurent[1]
[1]*SAMOVAR, Telecom SudParis, CNRS, University Paris-Saclay*
*Member of the Chair Values and Policies of Personal Information*
{*nesrine.kaaniche, maryline.laurent*}*@telecom-sudparis.eu*

Keywords: multi-level access control, attribute-based encryption, flexible and scalable access policies, data secrecy, user privacy

Abstract: The economy and security of modern society relies on increasingly remote and distributed infrastructures. This trend increases both the complexity of access control to outsourced data and the need of privacy-preserving mechanisms. Indeed, access control policies should be flexible and distinguishable among users with different privileges. Also, privacy preservation should be ensured against curious storage system administrators, for outsourced data, as well as access requestors identities if needed.
In this paper, we propose a multi-level access control mechanism based on an original use of attribute based encryption schemes. Our construction has several advantages. First, it ensures fine-grained access control, supporting multi-security levels with respect to different granted access rights for each outsourced data file. Second, relying on an attribute based mechanism, key management is minimized, such that users sharing the same access rights are not required to collaborate to extract the secret enciphering key. Third, our proposal is proven to provide efficient processing and communication overhead, compared to classical usage of attribute based encryption schemes.

## 1 Introduction

The increasing need and complexity of access control to outsourced data, along with the ever growing privacy concerns, has given rise to encryption mechanisms that combine privacy aspects, such as anonymity and unlinkability, with credentials that go beyond asserting a simple identity of a user, but rather for a full set of attributes. These mechanisms, referred to as Attribute based Encryption (ABE) and Attribute based Signature (ABS) mechanisms, used to encrypt and sign data files with respect to an access policy computed on a set of attributes. For example, in a hospital setting, access to a patient's records can be provided to the patient, his doctors, nurses, or to the administrative staff for the billing service. This can be formalized by an access policy based on users' attributes. This access policy must restrict access of actors to a subset of patient's records, to avoid nurses to access to personal information of the patient (e.g. name, address), and administrative staff to know his disease or health condition.

In this paper, we present a novel encryption scheme based on attribute based mechanisms for multi-level access policies. Our scheme ensures a selective access to data based on users' granted privileges. Practically, when a party encrypts a data file, she specifies an access structure and a certain number of security levels. Thus, a user is able to decrypt a sub-set of data blocks related to a security level $k$ if that user's private keys satisfy the sub-set of attributes related to the $k$-security level.

***Paper Organization*** **–** the remainder of this paper is organized as follows. First, section 2 discusses related works and highlights security and functional requirements. Then, section 3 introduces our definitions and gives background on access structures and Lagrange Interpolation. Afterwards, we introduce our system and threat models in section 4 and detail our concrete construction in section 5. Finally, we prove the security of our construction in section 6 and we evaluate the scheme performances in section 7 before concluding in section 8.

## 2 Related Work and Security Requirement Analysis

Sharing sensitive data between different involved actors is often an issue, due to the complexity of access control policies' management in remote and dis-

tributed infrastructures. Indeed, different deciphering keys can be distributed to different users that are allowed to access the corresponding data content, with respect to their granted privileges. However, the translation of an access control list into an equivalent multi-level policy remains the main issue of these schemes.

To forbid access to some parts of data, some processes propose to black out or remove these parts. These processes are referred to as *redaction* mechanisms (Miyazaki et al., 2006) (Steinfeld et al., 2002), (Johnson et al., 2002), (Ateniese et al., 2005).
Generally, the proposed schemes rely on malleable cryptographic primitives (e.g; chameleon hash functions instead of the usual hash functions) in order to allow redactors having their own secret key to modify some portions of the originally signed data file. Although these techniques permit selective access to some parts of data, they are also still inefficient with multi-level access privileges.

In 2010, Di Vimercati et al. (Di Vimercati et al., 2010) present a selective authorization policy model based on graph theory in order to ensure *read* privilege. In their proposal, the authors consider a dynamic group of users sharing data stored in remote cloud servers and assume that each data content may only be accessed by a subset of users. Indeed, for controlling data access, (Di Vimercati et al., 2010) relies on the use of both a key agreement algorithm and a key derivation algorithm that enables a key to be derived from another key and a public token. The combination of these two algorithms is able to correctly convert access policies defined by data owners into encryption policies. Afterwards, in 2012, Raykova et al. (Raykova et al., 2012) present an access control scheme that additionally supports the modification of the accessed data file. That is, in order to differentiate between *read* and *write* privileges, a public-private key pair for each data file is provided at the fine-grained level. Further, two token trees are built to distribute the private and public keys, respectively used to enforce read and write privileges. Recently, Di Vimercati et al. (di Vimercati et al., 2013) present another approach to support modification of outsourced data files. The basic idea of this approach is to associate each content with a write tag. The remote server allows a user to perform a write operation on a file if he correctly shows the corresponding write tag. A crucial concern of the (di Vimercati et al., 2013) scheme is that the keys used to encrypt write tags have to be shared between authorized users and the server. Although the attractive advantages of the proposed solutions (Di Vimercati et al., 2010), (di Vimercati et al., 2013), (Raykova et al., 2012) to support selective access control, they do not support multi-level access structure on the same data content.

Along with the different emerging techniques supporting multi-level access control to encrypted data, Attribute based Encryption (ABE) has been often presented as a solution to provide flexible data sharing (Sahai and Waters, 2005), (Bethencourt et al., 2007). In 2005, Sahai and Waters introduced the concept of ABE as a new technique for encrypted access control (Sahai and Waters, 2005). Contrary to traditional public key encryption mechanisms, both users' private keys and ciphertexts are associated with a set of attributes or a structure over attributes. The user is able to decrypt a ciphertext if there is a match between his private key and the ciphertext. Several works rely on ABE to realize fine grained access control for outsourced data (Hur and Noh, 2011),(Yu et al., 2010),(Jahid et al., 2011), (Ruj, 2014), (Horváth, 2015), (Huang et al., 2016). Although these schemes proposed efficient solutions to protect outsourced data from unauthorized access, they are still inefficient with multi-level access policies, where users have to share the same data content with different access rights to distinct parts of the data file.

## 2.1 E-health Scenario

In a real e-health scenario, different medical organisations and actors can be involved such as hospitals, research laboratories, pharmacies, health ministry as well as doctors, nurses and patients. On one hand, the shared data have to be protected from unauthorized access while ensuring fine grained access control for different authorized actors. Thus, the data confidentiality must be preserved against malicious users. As such, encryption should be applied while supporting flexible sharing of encrypted outsourced data among dynamic group of users, with fine-grained access control policies.
On the other hand, the private identifying information of the involved users, such as doctors and patients, must not be revealed to unauthorized actors. For instance, the system should not reveal any private information related to a doctor, such as his professional card, as well as his patients' personal data. Indeed, the disclosure of such information may be used to produce targeted advertisement related to the health condition of the patients or to run statistical surveys.

Let us consider the following use case: a doctor wants to partially share parts of the medical record of his patient with respect to different access control policies. For instance, he shares the health status of his patient with other doctors working in the cardi-

ology or infectious diseases departments, in order to have specialist advices on other related health problems. Similarly, he shares some attributes of the patient personal information (i.e; billing information) with hospital administrative staff to enable efficient billing services. The doctor also shares test blood results with laboratory staff and hospital administrative staff, to have detailed reports on blood tests. Finally, he shares care information with caregivers or nurses as well as emergency information with emergency physicians to ensure proper monitoring of the patient.
Thus, the aforementioned group of users define the following access control policies:

- access to billing information – (*hospital administrative staff* `and` *executive*) `and` *hospital Z*;
- access to medical information – (*doctor* `and` *infectious diseases department*) `and` (*hospital Y* `or` ( *hospital X* `and` *cardiology*));
- access to care instructions – (*head nurse* `or` *caregiver*) `and` (*hospital Y* `or` ( *hospital X* `and` *cardiology*));
- access to test blood results – *laboratory staff* `and` (*hospital administrative staff* `and` *executive*) `and` *hospital Z*;
- access to emergency information – *emergency physician* `and` (*hospital Y* `or` ( *hospital X* `and` *cardiology*)).

This use-case points out that in real-life scenarios access controls lists can be overlapped by introducing duplicated access attributes, to different parts of outsourced data. Hence, the management of access policies becomes more complex and the burden of enciphering keys' management rises mainly with dynamic group of users.

## 2.2 Naive Approach

ABE is usually considered to be the most suitable technique if authorized users have the same access rights to the whole data content. Nonetheless, as introduced in section 2.1, for the depicted e-health use case, authorized users do not have the same access privileges.

To enable access to encrypted data, the available option related to the use of ABE mechanisms is based on naive computing. For instance, the doctor creates an access structure for each part of the medical record which will be then encrypted, with respect to every group of authorized users. Hence, the major disadvantage of this approach is that it generates a processing overhead, mainly due to redundant subtrees and to the calculation of several secrets related to each independent access tree. Another shortcoming is that this approach considerably raises the size of the encrypted data file, generating a heavy communication overhead.
It is worthy noticeable that computation and communication costs should be minimized especially for e-health applications, where access and outsourcing of emergency information have to be optimized.

## 2.3 Security and Functional Requirements

Our objective is to design a new ABE scheme, which ensures a multi-level access control policies for the same data content. Our idea consists in creating an aggregate access tree permitting a multi-level access to the data file.
The design of our scheme is motivated by providing the support of both robustness and efficiency while fulfilling the following properties:

- ***data confidentiality*** **–** the proposed scheme has to protect the secrecy of encrypted data contents against malicious users, even in case of collusions.
- ***flexible access control*** **–** our proposal should ensure flexible security policies among dynamic groups of users with different granted privileges.
- ***privacy*** **–** preserving users' privacy is multifold. First, it is useful in a context where anonymity should be enforced to forbid any user's identification or personal information leakages (e.g. sex, age, address). Second, unauthorized users should not be able to deduce information about the redacted part of the data file, based on available parts of files or to link the encrypted content to a specific entity. Third, access to encrypted data should not reveal identifying information of the requesting entity.
- ***low processing cost*** **–** the encryption algorithm should have a low computational complexity to minimize the impact of security on the efficiency of e-health record processing.
- ***low communication overhead*** **–** our multi-level encrypted data file should be short-sized as the transmission overhead is important in the emerging infrastructure context.

# 3 Preliminaries

In this section, we provide some prerequisites, namely access structures and Lagrange interpolation.

**Definition 3.1.** *(**Access Structure** (Beimel, 2011)) Let $\mathcal{P} = \{P_1, P_2, \cdots, P_n\}$ be a set of parties, and a collection $\mathbb{A} \subseteq 2^{\{P_1,P_2,\cdots,P_n\}}$ is called monotone if $\forall B, C \subseteq 2^{\{P_1,P_2,\cdots,P_n\}}$ : if $B \in \mathbb{A}$ and $B \subseteq C$ then $C \in \mathbb{A}$. An access structure is a collection $\mathbb{A}$ of non-empty subsets of $\{P_1, P_2, \cdots, P_n\}$ ; i.e. $\mathbb{A} \subseteq 2^{\{P_1,P_2,\cdots,P_n\}} \setminus \{\emptyset\}$. The sets in $\mathbb{A}$ are called authorized sets, and the sets not in $\mathbb{A}$ are called unauthorized sets.*

We note that in several ABE schemes, these parties are considered as the attributes. In this paper, we consider a monotone access structure with multi-threshold security levels. The construction of such a access structure is detailed in section 5.1.

**Definition 3.2.** *(**Lagrange Interpolation**) Given a set of $(k+1)$ distinct points $\{(x_0, y_0), \cdots, (x_k, y_k)\}$, the Lagrange polynomial is a linear combination $L(x) = \sum_{j=0}^{k} y_j \delta_j(x)$ of Lagrange coefficients $\delta_j(x) = \prod_{0 \leq i \neq j \leq k} \frac{x - x_i}{x_j - x_i}$.*

# 4 Overview

Our flexible access control scheme is relying on ABE in the sense that clients' keys and decryption capabilities are related to the attributes they possess. In our proposal, the plaintext is comprised of a set of messages and the client's credentials (certified attributes) determine *which subset* of data blocks can be decrypted. More precisely, we propose to conceive a scheme as follows: the encrypting entity expresses an access structure with respect to $n$ attributes, while defining multi-security levels $\{k_l\}_{l \in \{1,c\}}$, where $k_l$ is the $k_l$-security level and $c$ is the number of defined security levels.
We note that each security level $k_l$ corresponds to $n_l$ identified sub-trees that permit to reconstruct a secret key $v_l$ needed to decrypt a subset of data blocks.

Referring to the e-health use-case, the encrypting entity represents the doctor who wants to share the medical record of his patient with different users. The authorized users to decipher outsourced data based on their granted credentials are referred to as decrypting entities or requestors. Encrypted data files are outsourced in remote servers, such as cloud servers. Thus, a cloud service provider is responsible for the system management.

## 4.1 System Model

Our multi-level attribute-based encryption scheme for a message space $\mathcal{M}$ and an access structure space $\mathcal{G}$ consists of four randomized algorithms, defined as follows:

- `setup` – this algorithm is performed by the central authority (i.e; the master entity). It takes as input the security parameter $\kappa$ and outputs the public parameters *params* and the master key *msk*.
- `encrypt` – this algorithm is executed by the enciphering entity. It takes as input the public parameters *params*, an access structure $\Gamma$ over the universe of attributes $\mathbb{S}$, the set of security levels $\{k_l\}_{l \in [1,c]}$, where $c$ is the number of security levels and a message $M = \{m_l\}_{l \in [1,c]}$. This algorithm encrypts the message $M$ with respect to the different security levels and outputs a ciphertext $CT = \{\Gamma, \forall k_l : \{ST_i\}_l, CT_l\}$, where $l \in [1,c]$ and $\{ST_i\}_l$ is the set of subtrees that have to be satisfied by each security level $k_l$. The encryption is performed such that only a user that possesses a set of attributes with regard to a security level $k_l$ that satisfies the required subtrees $\{ST_i\}_l$ can decrypt the enciphered message $CT_l$.
- `keygen` – the key generation algorithm is executed by the master entity. It takes as input the public parameters *params*, the master key *msk* and a set of attributes $\mathcal{S}$ and outputs the related secret key *sk*.
- `decrypt` – this algorithm is executed by the deciphering entity. It takes as input the public parameters *params*, the ciphertext $CT$, which contains an access policy $\Gamma$, the security level $k_l$, the set of required subtrees $\{ST_i\}_l$ and the secret key *sk* related to the set of attributes $\mathcal{S}$. The set of attributes has to satisfy the access structure $\Gamma$, with respect to a security level $k_l$ and the related subtrees $\{ST_i\}_l$, to be able to decrypt the corresponding ciphertext $CT_l$ and retrieve the message $m_l$.

The correctness property requires that for all security parameter $\kappa$, all universe descriptions $\mathbb{S}$, all $(params, msk) \in \texttt{setup}(\kappa)$, all $\mathcal{S} \subseteq \mathbb{S}$, all $M \in \mathcal{M}$, all $\Gamma \in \mathcal{G}$, all $sk \in \texttt{keygen}(params, msk, \mathcal{S})$, all $k_l \in \mathcal{K}$ ($\mathcal{K}$ is the security level space) and all $CT \in \texttt{encrypt}(params, \Gamma, M, \{k_l\}_{l \in [1,c]})$, if $\mathcal{S}$ satisfies $\Gamma$ with respect to a security level $k_l$ and the related subtrees $\{ST_i\}_l$, then the decryption algorithm $\texttt{decrypt}(params, CT, k_l, \{ST_i\}_l, sk)$ outputs $m_l$.

## 4.2 Security and Privacy Model

For our security and privacy model, we first assume that authorized users know, through an application, which policy needs to be applied on several data contents. Second, we suppose that data are organized into

several categories, to which the same access rights apply. Among the category, data might be of different types, but each category might at least include $k$ different types of data, so in case the category is present in a patient's record, it is not possible to infer information with regard to the type of information. For increased protection against inference, there might be interest in setting a range of possible size for each category.

For designing the most suitable security solution, we have to consider realistic threat models. That is, we point out two adversaries: *malicious* user and *honest but curious* server.

- *malicious user adversary* – a malicious user tries to override his rights. That is, he may attempt to deviate from the protocol or to provide invalid inputs. As such, we consider the malicious user adversary mainly against the confidentiality property.
- *honest but curious server adversary* – this storage server honestly performs the operations defined by our proposed scheme, but it may actively attempt to gain extra-knowledge about the outsourced sensitive data, and/or the identifying information of the requestors. Hence, we consider the honest but curious server adversary against the privacy property.

To prove that our scheme is secure against both *honest but curious* and *malicious* adversaries, we consider the security experiment $Exp_{\mathcal{A}}(1^{\kappa})$ between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. First, $\mathcal{A}$ selects a challenge access structure $\Gamma^*$, such that he can ask for any private keys generation of a set of attributes $\mathcal{S}$ as well as decryption queries of ciphertexts $CT$ that do not satisfy $\Gamma^*$. The security game $Exp_{\mathcal{A}}(1^{\kappa})$ is formally defined as follows:

SETUP – the challenger $\mathcal{C}$ runs the `setup` algorithm and gives the public parameters to the adversary $\mathcal{A}$.

QUERY PHASE – the adversary can repeatedly make any of the following queries:

- `obtain` : for each session $j$, $\mathcal{A}$ requests the private key $\{sk_j\}_{j\in[1,t]}$ associated to a set of attribute $\{\mathcal{S}_j\}_{j\in[1,t]}$ with respect to the security levels $\{k_{l,j}\}$. Note that if $\mathcal{C}$ already extracted a private key $sk_i$ for $\mathcal{S}_i$, then $\mathcal{C}$ returns $sk_i$.
- `cordec` : for each session, the adversary submits $(CT_j, \mathcal{S}_j)$ and asks for the decryption result of the ciphertext $CT_j$, under the private key for $\mathcal{S}_j$. If $\mathcal{C}$ has not previously extracted the private key $sk_j$ for $\mathcal{S}_j$, then $\mathcal{C}$ does the extraction based on the `obtain` algorithm. Then, the adversary $\mathcal{A}$ receives the output of the decryption algorithm of $CT_j$ with respect to the security levels $\{k_{l,j}\}$.

CHALLENGE – the adversary $\mathcal{A}$ submits two equal length messages $M_0$ and $M_1$. In addition, the adversary gives the access structure $\Gamma^*$ and the set of security levels $\{k_l\}^*$, such that none of the previous sets $\{\mathcal{S}_j\}_{j\in[1,t]}$ satisfies the access structure for $\{k_l\}^*$. The challenger flips a random coin $b$ and encrypts $M_b$ under $\Gamma^*$, with respect to $\{k_l\}^*$. The resulting ciphertext $CT^*$ is given to $\mathcal{A}$.

GUESS – the adversary outputs a guess $b'$ of $b$. The output of the experiment is 1 if and only if $b = b'$.

**Definition 4.1. CCA-1 security with respect to $Exp_{\mathcal{A}}(1^{\kappa})$ –** *Our multi-level access control scheme is selectively CCA-1 secure (i.e; selectively secure against chosen-ciphertext attacks) for an attribute universe $\mathbb{S}$ if for all probabilistic polynomial-time adversaries $\mathcal{A}$, there exists a negligible function $\varepsilon$, such that:*

$$Pr[Exp_{\mathcal{A}}(1^{\kappa}) = 1] = \frac{1}{2} \pm \varepsilon$$

# 5 Multi-Level Attribute based Encryption Construction

In this section, we detail our multi-level attribute-based construction (subsection 5.2) after introducing the model of the considered access tree (subsection 5.1).

## 5.1 Access Tree Model

Let $\Gamma$ be a tree representing the access structure. That is, $\Gamma$ is defined upon the following two levels:

- Level 1 – the first level presents the root node and its children. The root node is represented by the "AND "gate and it is defined as a $k_l$-*out-of-c* security levels. Each security level $k_l$ requires $p_l$ subsets of attributes and $n_l$ sub-trees of the root node for the reconstruction of the corresponding secret key $v_l$.
- Level 2 – the second level corresponds to interior nodes as well as leaf nodes. Each interior node of the tree is a threshold gate and the leaves are associated with attributes. We note that the second level corresponding to the different subtrees $\{\{ST_l\}_l\}$ is generated in the same way as in Bethencourt et al. construction(Bethencourt et al., 2007).

We note that we use the same notation as (Bethencourt et al., 2007) to describe the access tree. Each non-leaf node of $\Gamma$ is described by the number of its children $num_x$ and a threshold value $t_x$, where $1 \leq t_x \leq num_x$. If the threshold value $t_x = num_x$, then it is an "AND "gate, otherwise it is an "OR "gate.

As introduced in (Bethencourt et al., 2007), three additional functions are defined namely $\texttt{parent}(x)$, $\texttt{att}(x)$ and $\texttt{index}(x)$. The $\texttt{parent}(x)$ function denotes the parent of the node $x$, the $\texttt{att}(x)$ denotes the attributes associated with the leaf node $x$ and the $\texttt{index}(x)$ denotes a number associated with the node.

We denote by $\Gamma_x$ the subtree of $\Gamma$ rooted at the node $x$. If a set of attributes $\mathcal{S} = \{a_i\}_{i\in\{1,l\}}$, where $l$ is the number of attributes and $l \geq t_x$, satisfies the access tree $\Gamma_x$, it is referred to as $\Gamma_x(\mathcal{S}) = 1$.
Hence, depending on the number of the attributes $l$ and the required subtrees rooted by the root node, the user may decrypt the ciphertext $CT_l$, with respect to a security level $k_l$ and the related $\{ST_i\}_l$.
A user will be able to decrypt a ciphertext with a given key if and only if there is a match of attributes between the private key of the user and the nodes of the tree, such that the tree $\Gamma$ is satisfied.

## 5.2 Concrete Construction

Our multi-level attribute-based encryption scheme is based on the following algorithms:

$\texttt{setup}(\kappa)$ – this algorithm selects a bilinear group $(\hat{e}, \mathbb{G}_1, \mathbb{G}_2, g)$, such that $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$, $\mathbb{G}_1$ and $\mathbb{G}_2$ are two multiplicative groups of prime order $p$ and $g$ is a generator of $\mathbb{G}_1$.
The $\texttt{setup}$ algorithm selects two randoms $\alpha, \beta \in \mathbb{Z}_p$, and sets $X = \hat{e}(g,g)^{\alpha}$. The public parameters, considered as an auxiliary input to all the following algorithms, are defined as follows:

$$params = \{\mathbb{G}_1, \mathbb{G}_2, \hat{e}, g, h = g^{\beta}, X\}$$

The master key $msk$ is the pair $(\beta, g^{\alpha})$.

$\texttt{encrypt}(\Gamma, M, \{k_l\}_{l\in[1,c]})$ – this algorithm encrypts a message $M = \{m_l\}_{l\in[1,c]}$ under an access tree $\Gamma$, with respect to $k_c$ security levels. This algorithm first chooses a polynomial $q_x$ for each node $x$ and sets the degree $d_x$ of each polynomial as described in (Bethencourt et al., 2007), to be less than the threshold value such that $d_x = t_x - 1$ (i.e; $t_x$ is the threshold value of the node $x$).
We denote by $q_r$ the polynomial associated to the root node and defined as $q_r(x) = a_0 + a_1 x + \cdots + a_{d_r} x^{d_r}$.
We note that the degree of the root polynomial has to be at least equal to the total number of the root subtrees.
For each security level $k_l$, the encrypting entity defines the set of required subtrees $\{ST_i\}_l$ in order to derive the corresponding secret key $v_l$, such that $v_l = \sum_{i\in\{1,\cdots,n_l\}} q_r(index(x_i))$.

We suppose that $Y$ is the set of leaf nodes of the access tree $\Gamma$. The ciphertext is then defined as follows:

$$CT = \{\Gamma, \forall k_l : \{ST_i\}_l, \tilde{C}_{k_l} = m_l \cdot X^{v_l}, C_{k_l} = h^{v_l},$$

$$\forall y : C_y = g^{q_y(0)}, C'_y = \mathcal{H}(\texttt{att}(y))^{q_y(0)}\}$$

where $\mathcal{H}$ is a hash function, such that $\mathcal{H} : \{0,1\}^* \to \mathbb{G}_1$.

$\texttt{keygen}(msk, \mathcal{S})$ – this algorithm generates user's private keys related to the set of attributes $\mathcal{S}$, as defined in the Bethencourt et al. construction (Bethencourt et al., 2007). It first selects a random $r \in \mathbb{Z}_p$ and a set of random values $\{r_j\}$, where $j$ is the number of attributes in $\mathcal{S}$. The resulting key is represented as follows:

$$sk = \{D = g^{(\alpha+r)/\beta}, \forall a_j \in \mathcal{S} : D_j = g^r \cdot \mathcal{H}(j)^{r_j}, D'_j = g^{r_j}\}$$

$\texttt{decrypt}(CT, k_l, sk)$ – the decryption algorithm is based on two levels. Assume that the deciphering entity satisfies the $k_l$-security level with $n_l$ sub-trees of $\Gamma$ being satisfied. For the decryption algorithm, the deciphering entity starts by the second level:

- Level 2 – the algorithm works in a recursive manner, relying on the algorithm *DecryptNode* as presented in (Bethencourt et al., 2007).
- Level 1 – for the first level, we suppose that the deciphering entity satisfies the $k_l$-security level. Recall that the security level $k_l$ requires having the set $\{q_r(\texttt{index}(x_i))\}$ related to $\{ST_i\}_l$. As such, let $S_r$ be the set of a $n_l$-sized set of child nodes $x$ of the root node (i.e; $S_r$ corresponds to the set of required subtrees $\{ST_i\}_l$).
  To extract the deciphering key, the $\texttt{decrypt}$ algorithm computes $F_{R_{k_l}}$ such as:

$$F_{R_{k_l}} = \prod_{x\in S_r} \hat{e}(g,g)^{rq_x(0)} = \hat{e}(g,g)^{\sum_{x\in S_r} rq_x(0)} = \hat{e}(g,g)^{rv_l} \tag{1}$$

The $\texttt{decrypt}$ algorithm can now decrypt the ciphertext with respect to the $k_l$-security level, such as:

$$\frac{\tilde{C}_{k_l}}{\frac{\hat{e}(C_{k_l}, D)}{F_{R_k}}} = \frac{\tilde{C}_{k_l}}{\frac{\hat{e}(h^{v_l}, g^{(\alpha+r)/\beta})}{\hat{e}(g,g)^{rv_l}}} = \frac{\tilde{C}_{k_l}}{\frac{\hat{e}(g,g)^{(\alpha+r)v_l}}{\hat{e}(g,g)^{rv_l}}} = \frac{m_l \cdot X^{v_l}}{X^{v_l}} = m_l \tag{2}$$

# 6 Security Analysis

In this section, we prove the security of our multi-level attribute based encryption scheme with respect to the threat model detailed in section 4.2. First, we discuss that our construction ensures the confidentiality property in section 6.1. Then, we analyse the resistance of our scheme against privacy attacks in section 6.2.

## 6.1 Confidentiality

To ensure efficient access control, our construction mainly relies on the CP-ABE scheme proposed by Bethencourt et al. (Bethencourt et al., 2007). As such, the data confidentiality preservation is tightly related to the security of the used attribute based encryption algorithm.

**Theorem 6.1.** *Our multi-level access control scheme is secure against selective non-adaptive chosen ciphertext attacks in the Generic Group Model (GGM), with respect to the* $Exp_{\mathcal{A}}(1^{\kappa})$ *experiment.*

*Proof.* The indistinguishability property means that if an adversary has some information about the plaintext, he should not learn about the ciphertext. This security notion requires the computational impossibility to distinguish between two messages chosen by the adversary with a probability greater than a half. Indeed, in attribute-based encryption schemes, the adversary may lead an attack against the indistinguishability property either on his own or through a collusion attack.

On one hand, in order to decrypt a ciphertext with respect to a security level $k_l$, an adversary $\mathcal{A}$ may conduct an attack against the indistinguishability property. That is, he must recover $X^{v_l} = \hat{e}(g,g)^{\alpha \cdot v_l}$, where the secret sharing key $v_l$ is embedded in the ciphertext. For this purpose, $\mathcal{A}$ has to retrieve the corresponding $\tilde{C}_k$ and the related private key element $D$ from the user's private key.

To prove that our scheme ensures the confidentiality property, we first consider that the adversary $\mathcal{A}$ is running the $Exp^{conf}$ security game defined in Section 4.2 with an entity $\mathcal{B}$. This entity $\mathcal{B}$ is running the $Exp_{\mathcal{B}}$ Bethencourt et al. security game (Bethencourt et al., 2007), with the challenger $\mathcal{C}$. The objective of this proof is to show that the advantage of the adversary $\mathcal{A}$ to win the $Exp_{\mathcal{A}}(1^{\kappa})$ experiment is equivalent to the advantage of the entity $\mathcal{B}$ to win the Bethencourt et al. security game (Bethencourt et al., 2007).
Hereafter, $\mathcal{A}$ and $\mathcal{B}$ proceed as follows:

SETUP – the challenger $\mathcal{C}$ runs the `setup` algorithm and gives the public parameters to the adversary $\mathcal{B}$. Then, $\mathcal{B}$ sends `params` to $\mathcal{A}$.

QUERY PHASE – during this phase, $\mathcal{B}$ first initializes an empty table $T$. Then, the adversary can repeatedly make any of the following queries:

- `obtain` : for each session $j$, $\mathcal{A}$ requests the private key $\{sk_j\}_{j\in[1,t]}$ associated to a set of attribute $\{\mathcal{S}_j\}_{j\in[1,t]}$ with respect to the security levels $\{k_{l,j}\}$. The algorithm $\mathcal{B}$ uses the challenger $\mathcal{C}$ to generate and return the corresponding secret keys to the adversary $\mathcal{A}$. Recall that if $\mathcal{C}$ already extracted a private key $sk_j$ for $\mathcal{S}_j$, then $\mathcal{C}$ returns $sk_j$. The secret keys $\{sk_j, \mathcal{S}_j\}_{j\in[1,t]}$ are returned to $\mathcal{B}$. Afterwards, $\mathcal{B}$ sets a new entry with the pair $\{sk_j, \mathcal{S}_j\}_{j\in[1,t]}$ and returns the secret keys $\{sk_{j,GID}\}_{j\in N}$ to the adversary $\mathcal{A}$.
- `cordec` : for each session, the adversary submits $(CT_j, \mathcal{S}_j)$ and asks for the decryption result of the ciphertext $CT_j$, under the private key for $\mathcal{S}_j$. During this phase, $\mathcal{B}$ checks if an entry $sk_j$ for $\mathcal{S}_j$ does exist in table $T$ with respect to $\{\Gamma^*, k_{l,j}\}$. As such, if $\mathcal{C}$ has not previously extracted the private key $sk_j$ for $\mathcal{S}_j$, then $\mathcal{B}$ queries the extraction of $sk_j$, such that $\Gamma^*(\mathcal{S}_j, k_{l,j}) = 1$ based on the `obtain` algorithm. $\mathcal{B}$ receives $sk_j$ and deciphers $CT_j$, with respect to decryption algorithm defined in (Bethencourt et al., 2007). Then, the adversary $\mathcal{A}$ receives the output of the decryption algorithm of $CT_j$ with respect to the security level $k_{l,j}$.

CHALLENGE – the adversary $\mathcal{A}$ submits two equal length messages $M_0$ and $M_1$. In addition, the adversary gives the access structure $\Gamma^*$ and the set of security levels $\{k_l\}^*$, such that none of the previous sets $\{\mathcal{S}_j\}_{j\in[1,t]}$ satisfies the access structure for $\{k_l\}^*$. Once receiving the challenge access $\Gamma^*$, the algorithm $\mathcal{B}$ first selects $\Gamma_{\mathcal{B}}$ such that $\Gamma_{\mathcal{B}} \subseteq \Gamma^*$. We have to note that all pre-identified subtrees $ST_i$ required to satisfy the security level $\{k_l\}^*$ have to be included in the selected access structure $\Gamma_{\mathcal{B}}$.
Afterwards, $\mathcal{B}$ sends the access structure $\Gamma_{\mathcal{B}}$ and the two equal length messages $M_0$ and $M_1$, defined by the adversary $\mathcal{A}$. The challenger flips a random coin $b$ and encrypts $M_b$ under $\Gamma_{\mathcal{B}}$. The resulting ciphertext $\{CT_b\}^*$ is given to $\mathcal{A}$.

For our analysis, we distinguish two different cases for the $Exp_{\mathcal{A}}(1^{\kappa})$ experiment defined in section 4.2:

- **Case 0 –** we set only one security level ${k_l}^*$, during the SETUP phase. That is, we define a single security level, such that all queried private keys are re-

lated to the set of attributes $\mathcal{S}_i$ that decrypt ciphertexts, encrypted with respect to $k_l{}^*$, for each session $i$. This first case simulates a CCA game for a CP-ABE scheme as presented in (Bethencourt et al., 2007). In fact, the two first steps SETUP and QUERY PHASE of the $Exp_{\mathcal{A}}(1^\kappa)$ experiment are similar to the (Bethencourt et al., 2007) security game. In addition, the challenge access tree defined by the adversary $\mathcal{A}$ is equivalent to the access structure selected by the algorithm $\mathcal{B}$, such that $\Gamma_{\mathcal{B}} = \Gamma^*$, where all sub-trees of $\Gamma^*$ have to be included in $\Gamma_{\mathcal{B}}$.

- **Case 1 –** we set the maximum number of security levels $c^*$ such as $c^* > 1$, during the SETUP phase. For each session $i$, we suppose that $\mathcal{A}$ has access to $CT_i = \{CT_{l,i}\}_{l \in [1,c^*]}$, where $CT_{l,i}$ is an encrypted data block $m_{l,i}$ under a security level $k_{l,i}$. During the CHALLENGE phase, $\mathcal{A}$ submits two different messages $M_0$ and $M_1$ and asks the challenger $\mathcal{C}$ to encipher the selected message under a security level $k_l{}^*$ that has never been queried during the QUERY PHASE. As such, the algorithm $\mathcal{B}$ has to select $\Gamma_{\mathcal{B}}$ where identified subtrees $ST_i$ required to satisfy the security level $\{k_l\}^*$ of $\Gamma^*$ have to be included in $\Gamma_{\mathcal{B}}$.

First, let us make the following common consideration: in the aforementioned security game, including **Case 0** and **Case 1**, the challenge ciphertext has a component $\tilde{C}_k$ which is either $M_0 \cdot X^{v_l}$ or $M_1 \cdot X^{v_l}$, where $v_l$ is the enciphering secret. So that, we consider a modified game, defined in (Bethencourt et al., 2007), in which $\tilde{C}_k$ is either $\hat{e}(g,g)^{\alpha \cdot v_l}$ or $\hat{e}(g,g)^{\theta}$, where $\theta$ is selected uniformly at random. The adversary $\mathcal{A}$ has to determine which is the case. The adversary advantage is obviously equal to $\varepsilon$ in the original security game. But, in the modified game, the adversary advantage is at least $\varepsilon/2$.
In the following, we consider the adversary's advantage in the modified game.

As introduced in (Boneh et al., 2005), with respect to a generic group model, each element of $\mathbb{G}_1$ and $\mathbb{G}_2$ is encoded as a unique random. As such, $\mathcal{A}$ cannot test more than the equality property. The encoding properties of elements in $\mathbb{G}_i$ is presented by $\xi_0 : \mathbb{Z}_p \to \{0,1\}^*$ that maps all $a \in \mathbb{Z}_p$ to the representation $\xi_0(a)$ of $g^a \in \mathbb{G}_1$ and $\xi_T : \mathbb{Z}_p \to \{0,1\}^*$ that maps all $a \in \mathbb{Z}_p$ to the representation $\xi_T(a)$ of $\hat{e}(g,g)^a \in \mathbb{G}_2$. The adversary communicates with the oracles to perform actions in $\mathbb{G}_1$, $\mathbb{G}_2$ and $\hat{e}$ based on $\xi_0$ and $\xi_T$ representations.

For **Case 0**, during the SETUP phase, the challenger chooses two randoms $\alpha$ and $\beta \in \mathbb{Z}_p$ and sends the public parameters $\xi_0(1) = g, \xi_0(\beta) = h$ and $\xi_T(\alpha)$ to the adversary. Afterwards, $\mathcal{B}$ initializes an empty table $T$. And, during the QUERY PHASE, $\mathcal{A}$ queries several times `obtain` and `cordec` algorithms. For each `obtain` query, the challenger $\mathcal{C}$ simulates the $\mathcal{H}$ oracle function for each string $i \in \mathcal{S}_j$, queried in session $j$. The $\mathcal{H}$ oracle outputs $g^{t_i}$ for each different queried $i$. Consequently, for a session $j$, the `obtain` oracle chooses a random $r^{(j)}$, computes $D_k = h^{\alpha / r^{(j)}}$ and for each $i \in \mathcal{S}_j$, it provides $D_i = g^{r^{(j)} + t_i r_i{}^{(j)}}$ and $D'_i = g^{r_i{}^{(j)}}$. These values are then set as new entries in $T$, by $\mathcal{B}$ and sent to the adversary $\mathcal{A}$.
Then, for the `cordec` oracle, $\mathcal{A}$ sends the pair $(\mathcal{S}_j, CT_j)$ and asks for the decryption of $CT_j$, with respect to the predefined security level $k_l{}^*$. Thus, $\mathcal{B}$ checks if an entry $sk_j$ for $\mathcal{S}_j$ does exist in table $T$ with respect to $\{\Gamma^*, k_{l,j}\}$. Then, if $\mathcal{C}$ has not previously extracted the private key $sk_j$ for $\mathcal{S}_j$, then $\mathcal{C}$ does the extraction based on the `obtain` algorithm. Subsequently, $\mathcal{B}$ computes the decryption of a ciphertext $CT^{(j)}$ for each session $j$ and provides a message $M_j$ or an error message if the set of attributes does not pass the access structure with respect to the pre-defined security level.
Clearly, our multi-level access control scheme is close to the CP-ABE construction proposed by Bethencourt et al. in (Bethencourt et al., 2007). The main difference consists in the derivation of the embedded secret $v_l$ corresponding to a pre-defined security level $k_l{}^*$. Indeed, unlike the (Bethencourt et al., 2007) scheme based on Lagrange Interpolation, in our construction, the processing of Level 1 of the access structure $\Gamma$, with respect to $k_l{}^*$, requires the multiplication of the different elements of $S_r$, in order to get $v_l$, where the number of elements of $S_r$ is lower than the degree of the root polynomial $q_r$. More precisely, the main difference mainly consists in $X^{v_l} = \hat{e}(g,g)^{\alpha \cdot v_l}$, where $v_l = \sum s_i$ in the $Exp_{\mathcal{A}}(1^\kappa)$ experiment while $v_l = s$ computed with respect to Lagrange Interpolation in the (Bethencourt et al., 2007) construction.
To prove that **Case 0** is close to the (Bethencourt et al., 2007) construction, we consider an *absurdum* reasoning, where $\mathcal{A}$ can win the $Exp_{\mathcal{A}}(1^\kappa)$ experiment with a non negligible probability. Let us consider that the root polynomial in $Exp_{\mathcal{B}}(1^\kappa)$ is equal to $q_{r,Exp_{\mathcal{B}}}(x) = \sum_{i=0}^{p} a_i x^i$, where $a_0 = s$. Thus, we have to verify if there exists one polynomial $q_{r,Exp_{\mathcal{A}}}$, such that $q_{r,Exp_{\mathcal{A}}} = \sum_{i=0}^{p} a'_i x^i$ and $\sum_{j=1}^{p} \sum_{i=0}^{p} a'_i {x_j}^i = s$.
Let us consider $p-1$ random values $(a'_i)$, where $i \in [1, p-1]$. Thus, we have the following equality:

$$\sum_{j=1}^{p} \sum_{i=0}^{p} a'_i {x_j}^i = s = \sum_{j=1}^{p} \sum_{i=0}^{p-1} a'_i {x_j}^i + \sum_{j=1}^{p} a'_p {x_j}^p \quad (3)$$

In the sequel, from Equation 3, we deduce that:

$$a'_p = \frac{s - \sum_{j=1}^{p} \sum_{i=0}^{p-1} a'_i x_j{}^i}{\sum_{j=1}^{p} a'_p x_j{}^p} \quad (4)$$

From Equation 3 and Equation 4, we deduce that the polynomial $q_{r,Exp_{\mathcal{A}}}$ exists. Consequently, the adversary $\mathcal{A}$ receives the challenge ciphertext $CT_b = \{\Gamma^*, k_l{}^*, \tilde{C}_{k_l{}^*} = M_b \cdot X^{\sum s_i}, C_{k_l{}^*} = h^{\sum s_i}, \forall y : C_y, C'_y\}$.
If $\mathcal{A}$ can win the $Exp_{\mathcal{A}}(1^{\kappa})$ experiment with a non negligible probability, then $\mathcal{A}$ can guess $b'$ which is therefore sent to $\mathcal{B}$. In the sequel, $\mathcal{B}$ can win the security game $Exp_{\mathcal{B}}$, introduced in (Bethencourt et al., 2007), with a non negligible probability. This contradicts our assumption that (Bethencourt et al., 2007) is proved secure in GGM model.
In addition, noticing that for **Case 0** of $Exp_{\mathcal{A}}$, the SETUP, QUERY and CHALLENGE phases are based on one single security level, where the challenge message $M_b$ contains one single data block related to the security level $k_l$, such that $\Gamma_{\mathcal{B}} = \Gamma^*$, where all sub-trees of $\Gamma^*$ have to be included in $\Gamma_{\mathcal{B}}$. As such, the advantage of the adversary is at most equal to $O(\frac{q^2}{p})$, where $p$ is the order of an additive group $\mathbb{F}_p$ and $q$ is a bound on the total number of group elements received by any adversary $\mathcal{A}$ from its interaction with the $Exp_{\mathcal{A}}$ security game and the different oracles.

For **Case 1**, the SETUP phase is executed similarly as for **Case 0**. In fact, the challenger $\mathcal{C}$ sends the public parameters $\xi_0(1) = g, \xi_0(\beta) = h$ and $\xi_T(\alpha)$ to the adversary. For ease of presentation, we do not show the progress of SETUP and QUERY PHASE between $\mathcal{C}$, $\mathcal{B}$ and $\mathcal{A}$, where the outputs of `obtain` and `cordec` are closely similar to **Case 0**, considering $c^*$ ciphertexts related to $c^*$ security levels.
During the challenge phase, when $\mathcal{A}$ asks for the encryption of the challenge message with respect to a challenge access structure $\Gamma^*$, $\mathcal{C}$ does the following. $\mathcal{C}$ first chooses a random $a_0 \in \mathbb{F}_p$ and uses the linear secret sharing scheme associated with the access structure $\Gamma^*$ to construct the shares $\sigma_k$ and $\lambda_i$ of $s$ for all relevant sub-trees $k$ and attributes $i$, respectively. As explained in (Bethencourt et al., 2007), both $\lambda_i$ and $\sigma_k$ shares have to be chosen uniformly and independently at random from $\mathbb{F}_p$, in order to respect the linear conditions imposed by the secret sharing scheme presented in (Bethencourt et al., 2007). Afterwards, the simulation chooses $c^*$ randoms $\theta_l \in \mathbb{F}_p$, where $l \in [1, c^*]$.

Finally, $\mathcal{C}$ outputs the encryption of the challenge message such that: for each security level $k_l$, we have $\tilde{C}_{k_l} = \hat{e}(h,h)^{\theta_l}$ and $C_{k_l} = h^{v_l}$, where $v_l = \sum_{i \in \{1,\cdots,n_l\}} \sigma_i$. (cf. section 5). For each relevant attribute $i$, we have $C_i = g^{\lambda_i}$ and $C'_i = g^{t_i \lambda_i}$. These values are then sent to the adversary. We state that if $\mathcal{A}$ asks for a decryption key for a set of attributes that pass $\Gamma^*$ with respect to any security level, then $\mathcal{C}$ does not issue the key. Similarly, if $\mathcal{A}$ asks for $\Gamma^*$, with respect to any security level, such that one of the keys is already issued then the simulation aborts. In the sequel, the advantage of the adversary is at most equal to $O(c^* \frac{q^2}{p})$, due to the randomness of the choice of variable values in the simulation. Indeed, the adversary' view in this simulation is identically distributed for all security levels. In fact, the encryptions of data blocks of the challenge message $M_b$ are completely independent, thanks to the use of the encoding function $\xi_T$. As such, **Case 1** can be considered as $c^*$ random repetitions of **Case 0** simulation, with respect to $c^*$ security levels.

On the other hand, one of the main challenge to design our multi-level attribute based encryption scheme was to prevent collusion attacks between users. Hence, our scheme randomizes users' private keys, as introduced in (Bethencourt et al., 2007), such that they cannot be combined. In fact, each private key element $D_j$, associated with an attribute $j$, contains a random value $r$ related to the user, and $r_j$ associated to the attribute $j$, which prevents colluding users to override their rights and successfully perform a collusion attack. In addition, as discussed in the aforementioned **Case 0** and **Case 1**, the $Exp_{\mathcal{A}}(1^{\kappa})$ experiment and the (Bethencourt et al., 2007) security game are shown to provide equivalent adversaries' advantages, with respect to selective chosen ciphertext attacks. Consequently, our multi-level access control mechanism is resistant against collusion attacks.

As such, we prove that our multi-level access control construction is secure against selective non-adaptive chosen ciphertexts attacks in the Generic Group Model (GGM), with respect to $Exp_{\mathcal{A}}(1^{\kappa})$ experiment.

□

## 6.2 Privacy

As introduced in section 2.3, the privacy preserving requirement distinguishes the privacy of contents against malicious users and the privacy of legitimate users against honest but curious adversaries.

**Theorem 6.2.** ***Privacy*** *Our multi-level attribute based encryption scheme is private, against both malicious and honest but curious adversaries.*

*Proof.* The proof of Theorem 6.2 is twofold.

- **Case a –** it considers the case of malicious users against the privacy of contents. That is, the adversary $\mathcal{A}$ tries to override his rights in order to get access to the encrypted personal information, embedded in queried ciphertexts. In our construction, personal information, referred to as $pi_l$, is encrypted under a security level $k_l$, where an adversary attempts to get access to the embedded secret $v_l$.
  Obviously, **Case a** joins the confidentiality requirement (cf. Theorem 6.1) with respect to the $Exp_{\mathcal{A}}(1^{\kappa})$ experiment presented in section 6.1, where encrypted messages are considered as sensitive identifying information $pi_l$. In fact, when an adversary tries to override his rights in order to get access to encrypted messages, he concretely leads an attack either on his own or through a collusion attack, as shown in in section 6.1. As such, as discussed in the aforementioned **Case 0** and **Case 1**, the $Exp_{\mathcal{A}}(1^{\kappa})$ experiment and the (Bethencourt et al., 2007) security game are shown to provide equivalent adversaries' advantages, with respect to selective chosen ciphertext attacks.
  In addition, let us notice that **Case 1** of the $Exp_{\mathcal{A}}(1^{\kappa})$ experiment can be modeled in *multi-user setting*, such that there are multiple challenge ciphertexts that can be dependent. In our case, the challenge ciphertexts represent the different pieces of the challenge message $M^* = \{m_l{}^*\}_{l\in[1,c^*]}$. Thus, this case is considered as a generalization of selective CCA security in the multi-user setting and the adversary $\mathcal{A}$ can make multiple *Left-or-Right* queries. These challenge ciphertexts have to be created with the same selector $b$; i.e; all ciphertexts are encryption of the left input, or all ciphertexts are encryption of the right input.
  Despite the multi-user setting, the proof of Theorem 6.1 shows that the advantage of $\mathcal{A}$, against the indistinguishability property is negligible.
  Indeed, we state that the encryption of every set of data blocks related to any security level $k_l$ is completely independent, thanks to the use of the encoding function $\xi_T$. Hence, the encryption scheme does not convey any information about the set of data blocks that have been enciphered for each security level.
  Thus, our multi-level access control scheme ensures the privacy of contents, against malicious users.
- **Case b –** it considers the case of honest but curious adversaries against the privacy of users. That is, the attacker $\mathcal{A}$ attempts to distinguish between two legitimate requesting users $u_1$ and $u_2$, trying to deduce the identifying information related to each requestor, with respect to their related attributes. Note that each user $u_j$ possesses a set of attributes $\mathcal{S}_{u_j}$, where $\Gamma(\mathcal{S}_{u_j}) = 1$, $j \in \{1,2\}$ and $\mathcal{S}_{u_1} \neq \mathcal{S}_{u_2}$.
  For **Case b**, we set a security level $k_l{}^*$, during the SETUP phase, where all queried ciphertexts are encrypted with respect to $k_l{}^*$, under an access tree $\Gamma$. Unless there exists an authentication procedure for managing access to cloud resources, it is worthy noticeable that the adversary $\mathcal{A}$ cannot distinguish between $u_1$ and $u_2$, because $\Gamma(\mathcal{S}_{u_1}) = \Gamma(\mathcal{S}_{u_2}) = 1$. Indeed, our scheme inherits this privacy-preserving property from the attribute-based encryption mechanisms. Thus, our multi-level attribute base encryption scheme ensures the privacy of users, against honest but curious adversaries.

Referring to the e-health use case, presented in section 2.1, the privacy preserving property is highly recommended, especially against curious outsiders, where access patterns are important pieces of information that have to be protected. For example, the unlinkability property supported by our construction ensures that a curious adversary cannot deduce if a patient is followed either by doctor *A* or doctor *B*.

If a perfect privacy property is needed, our multi-level access control construction can be extended by considering an anonymous authentication system supporting the inspection feature, based on the use of attribute based signatures, as presented in Kaaniche and Laurent proposal (Kaaniche and Laurent, 2016a), (Kaaniche and Laurent, 2016b). Indeed, this enables a user to anonymously authenticate with the cloud provider, while providing only required information, with respect to its access policy, before accessing to cloud resources. Hence, this extension ensures unlinkability between the different sessions while preserving the anonymity of the requesting user. In addition, our scheme can easily support hidden access control structures, where access patterns are encrypted with Public Key Encryption with Keyword Search scheme (PEKS) (Asghar et al., 2014) (Boneh et al., 2004).

□

# 7 Theoretical Performances

In this section, we discuss the processing and communication overhead of our proposed multi-level attribute-based scheme $\mathtt{ML-ABE}$ compared to the

naive approach NA introduced in section 2.2. As such, we assess theoretical complexity where the encrypting entity has to create $k$ different access control policies for the naive approach.
To this purpose, we define the following costs:

- $\gamma_M$ : cost of two group elements' multiplication in a multiplicative group
- $\gamma_E$ : cost of an exponentiation in a multiplicative group
- $|MT|$ : size of an aggregate access tree, referred to as master tree
- $|AT|$ : size of an access tree for an access policy $k$
- $Y_{MT}$ : number of leaves of the master access tree
- $Y_{AT}$ : number of leaves of an access tree, with respect to an access policy $k$
- $|E|$ : size of a multiplicative group element

Table 1 presents detailed computation and communication overhead comparison between our proposed construction and the naive approach, based on the processing cost and the size of the ciphertext. Note that the communication and storage overhead are both referring to the size of the ciphertext.

Table 1: Theoretical Comparisons between $\mathtt{ML-ABE}$ and NA

| | $\mathtt{ML-ABE}$ | NA |
|---|---|---|
| Processing Cost | $k\gamma_M + 2(k + Y_{MT})\gamma_E$ | $k\gamma_M + 2k(1 + Y_{AT})\gamma_E$ |
| Size of Ciphertext | $\{MT, 2(k + Y_{MT})\|E\|\}$ | $\{kAT, 2k(1 + Y_{AT})\|E\|\}$ |

It is worthy noticeable that the size of the master access tree, proposed in our construction $\mathtt{ML-ABE}$, is lower than the size of the set of access trees related to $k$ access policies introduced by the naive approach NA. This is mainly due to the involved number of attributes (access tree leaves), that should be duplicated for different access tree in NA. Obviously, the number of leaves of the master tree $Y_{MT}$ would be lower than the sum of leaves of access trees related to $k$ access structures of NA, such that $Y_{MT} \leq \sum_k Y_{AT_k}$. Consequently, the communication and storage costs introduced by the $\mathtt{ML-ABE}$ approach are considerably optimized, compared to NA.

In addition, for the NA approach, the enciphering entity has to create an access tree $AT$ to each different security level. Thus, he has to assign different polynomials to each node of each access tree.
Consequently, our approach presents competitive processing and communication costs, where the number of polynomials, that have to be assigned to each node of an access tree, is reduced compared to NA, thanks to the use of an aggregate access structure.

Nonetheless, our approach is not convenient when defining different independent access policies under the same master access tree (i.e; there is no duplicated attributes for each defined security level $k$), as well as for dynamic environments requiring the modification of the encryption policies. Hence, in such use-cases, the NA approach and our multi-level access control approach introduce similar processing and communication costs.

Finally, the $\mathtt{ML-ABE}$ approach presents interesting computation, communication and storage overhead in collaborative use cases, thanks to the definition of multiple access structures. However, it is still inappropriate for hierarchical scenarios that require restrictive privileges, such as in redaction use-cases in military services. That is, these use cases often rely on encapsulated access structures, defined by hierarchical levels of security, such that each higher level of security $k+1$ introduces additional attributes, compared to the security level $k$, that have to be satisfied with respect to the related access policy $AT_{k+1}$.

# 8 Conclusion

In this paper, we presented a new scheme to design multi-level access control policies for e-health applications based on an original usage of attribute based encryption schemes.
Indeed, our proposal ensures flexible fine-grained access control, supporting multi-security levels with respect to different granted access rights for each outsourced data file.
Additionally, our multi-level access attribute-based scheme is deliberately designed to ensure the confidentiality and privacy preserving properties against both malicious and honest but curious adversaries. As such, our construction is proven secure against selective, non-adaptive chosen ciphertext attacks in the generic group model. Finally, a quantitative comparison of our proposal with the naive approach shows the interesting processing and communication cost of our multi-level access control scheme, due to the application of aggregate access policies.

# REFERENCES


Asghar, M. R., Gehani, A., Crispo, B., and Russello, G. (2014). Pidgin: Privacy-preserving interest and content sharing in opportunistic networks. In *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security*, ASIA CCS '14, pages 135–146. ACM.

Ateniese, G., Chou, D. H., de Medeiros, B., and Tsudik, G. (2005). *Sanitizable Signatures*. Springer Berlin Heidelberg.

Beimel, A. (2011). Secret-sharing schemes: A survey. IWCC'11.

Bethencourt, J., Sahai, A., and Waters, B. (2007). Ciphertext-policy attribute-based encryption. In *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, SP '07, Washington, DC, USA. IEEE Computer Society.

Boneh, D., Boyen, X., and Goh, E.-J. (2005). *Hierarchical Identity Based Encryption with Constant Size Ciphertext*. Springer Berlin Heidelberg.

Boneh, D., Crescenzo, G. D., Ostrovsky, R., and Persiano, G. (2004). Public key encryption with keyword search.

Di Vimercati, S. D. C., Foresti, S., Jajodia, S., Paraboschi, S., Pelosi, G., and Samarati, P. (2010). Encryption-based policy enforcement for cloud storage. In *Distributed Computing Systems Workshops (ICDCSW), 2010 IEEE 30th International Conference on*, pages 42–51. IEEE.

di Vimercati, S. D. C., Foresti, S., Jajodia, S., Paraboschi, S., and Samarati, P. (2013). On information leakage by indexes over data fragments. In *Data Engineering Workshops (ICDEW), 2013 IEEE 29th International Conference on*, pages 94–98. IEEE.

Horváth, M. (2015). Attribute-based encryption optimized for cloud computing. In *SOFSEM 2015: Theory and Practice of Computer Science*, pages 566–577. Springer.

Huang, Q., Yang, Y., and Shen, M. (2016). Secure and efficient data collaboration with hierarchical attribute-based encryption in cloud computing. *Future Generation Computer Systems*.

Hur, J. and Noh, D. K. (2011). Attribute-based access control with efficient revocation in data outsourcing systems. *IEEE Transactions on Parallel and Distributed Systems*, 22(7):1214–1221.

Jahid, S., Mittal, P., and Borisov, N. (2011). Easier: Encryption-based access control in social networks with efficient revocation. In *The 6th ACM Symposium on Information, Computer and Communications Security*, pages 411–415. ACM.

Johnson, R., Molnar, D., Song, D., and Wagner, D. (2002). *Homomorphic Signature Schemes*, pages 244–262. Springer Berlin Heidelberg.

Kaaniche, N. and Laurent, M. (2016a). Attribute-based signatures for supporting anonymous certification. In *European Symposium on Research in Computer Security*, pages 279–300. Springer.

Kaaniche, N. and Laurent, M. (2016b). Security analysis of habs. In *Unpublished Note*, pages 1 –7.

Miyazaki, K., Hanaoka, G., and Imai, H. (2006). Digitally signed document sanitizing scheme based on bilinear maps. In *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security*, ASIACCS '06, pages 343–354. ACM.

Raykova, M., Zhao, H., and Bellovin, S. M. (2012). Privacy enhanced access control for outsourced data sharing. In *International Conference on Financial Cryptography and Data Security*, pages 223–238. Springer.

Ruj, S. (2014). Attribute based access control in clouds: A survey. In *IEEE International Conference on Signal Processing and Communications (SPCOM)*, pages 1–6.

Sahai, A. and Waters, B. (2005). Fuzzy identity-based encryption. In *EUROCRYPT 2005*, pages 457–473. Springer.

Steinfeld, R., Bull, L., and Zheng, Y. (2002). *Content Extraction Signatures*. Springer Berlin Heidelberg.

Yu, S., Wang, C., Ren, K., and Lou, W. (2010). Attribute based data sharing with attribute revocation. In *The 5th ACM Symposium on Information, Computer and Communications Security*, pages 261–270.