SABE: A Selective Attribute-based Encryption for an Efficient Threshold Multi-level Access Control

Nesrine Kaaniche and Maryline Laurent
SAMOVAR, CNRS, Telecom SudParis, University Paris-Saclay,
Member of the Chair Values and Policies of Personal Information, Paris, France

Keywords: Multi-level Access Control, Ciphertext-policy Attribute-based Encryption, Flexible Access Policies, Data Secrecy.

Abstract: With the emergence of decentralized systems and distributed infrastructures, access control to outsourced data becomes more complex, as it should be flexible and distinguishable among users with different access rights. In this paper, we present SABE, a Selective Attribute-based Encryption scheme, as a new threshold multi-level access control mechanism based on an original use of attribute based encryption schemes. Our proposal is multi-fold. First, it ensures fine-grained access control, supporting multi-security levels with respect to different granted access privileges for each outsourced data file. Second, SABE is proven secure against selective non-adaptive chosen ciphertext attacks in the generic group model. Third, our construction is proven to provide efficient processing and communication complexities, compared to most closely related schemes.

1 INTRODUCTION

The emergence of decentralized systems and distributed infrastructures has increased the complexity of access control to outsourced data and has given rise to encrypted access schemes. These mechanisms, referred to as Attribute based Encryption (ABE) mechanisms, are used to encrypt data files with respect to an access policy computed on a set of attributes.

Let us take a simple example for illustrating the need for a multi level access tree for ABE. During the reviewing process of scientific papers submitted to a conference, access to the papers can be provided to general chairs, program committee, publicity chairs or to the registration manager for the validation of registered papers. This can be formalized by an access policy based on users' attributes, which assigns different access rights to each part of the scientific paper to the actors. For instance, reviewers should not have access to identifying information of the authors (e.g; name, affiliation, $\cdots$), registration staff should not have access to the paper contents, while general chairs have access to the whole research paper's content. Thus, to enable access to encrypted data, the encryptor is assumed to create an access structure for each part of the scientific paper which is then encrypted, w.r.t. each group of authorized entities. Depending on the attributes comprising each access structure, there might be a strong overlapping of attributes between access structures, thus leading to duplicated efforts when encrypting and decrypting each part of the data content. In addition, the management of access control policies becomes more complex and the burden of enciphering keys' management rises mainly with the dynamicity of users groups.

In this paper, we present SABE, Selective Attribute based Encryption, a novel and efficient threshold encryption scheme relying on attribute based mechanisms, and provided with the following features:
(1) SABE ensures a selective access to data based on users' granted privileges. A party willing to encrypt a data file only specifies one single access structure and certain security levels assigned to different data blocks of the enciphered file. Thus, a user is next able to decipher a sub-set of data blocks associated to a security level k if the secret keys of that user satisfy the related k_l sub-sets of attributes.
(2) SABE is a threshold access-control scheme associated to multi-security levels, i.e. each security level defines the threshold number k_l of related sub-sets of attributes that need to be satisfied.
(3) Relying on an attribute based encryption mechanism, users sharing the same access privileges are not required to collaborate to extract the secret encrypting key. Thus, the complexity of key management is

Kaaniche, N. and Laurent, M.
SABE: A Selective Attribute-based Encryption for an Efficient Threshold Multi-level Access Control.
DOI: 10.5220/0006855501550167
In *Proceedings of the 15th International Joint Conference on e-Business and Telecommunications (ICETE 2018) - Volume 2: SECRYPT*, pages 155-167
ISBN: 978-989-758-319-3

minimized, providing an efficient processing and communication overhead, and the definition of access control policies is flexible and distinguishable among users with different privileges to access data.
(4) SABE is proven secure against non-adaptive chosen ciphertext attacks.

Paper Organization – Section 2 discusses related works, presents the problem statement and highlights security and functional requirements. Section 3 introduces our definitions and gives background on access structures and Lagrange Interpolation. Section 4 details our system and threat models and section 5 presents the concrete construction.The security of our scheme is discussed in section 6. Finally, The scheme performances are evaluated in section 7 The potential of SABE technique to support security and privacy in concrete networking and computing applications in section 8 before concluding in section 9.

2 RELATED WORKS AND SECURITY REQUIREMENTS ANALYSIS

Sharing data contents between different involved actors is often an issue, due to the complexity of access control policies' management. This issue becomes more complex when involved actors do not share the same access privileges to each part of the data file. In the following, we first detail related works in subsection 2.1. Then, we present the problem statement based on a real-world use case, in subsection 2.2. Afterwards, we introduce the security and functional requirements for the design of the SABE mechanism in subsection 2.3.

2.1 Related Work

Fine-grained access control based on selective encryption is a novel approach that enables selective access to encrypted data while supporting a compelling key management process. 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 ban access to some parts of the data, some processes propose to cover out or remove these chunks. These mechanisms are known by *redaction* tools. They generally rely on malleable cryptographic primitives (e.g; chameleon hash functions instead of the conventional hash functions) to enable redactors, based on their own respective private keys to modify some portions of the originally encrypted file. Although these mechanisms allow selective access to some parts of the originally encrypted file, they are also still inefficient with multi-level access privileges.

In 2010, Di Vimercati et al. (Di Vimercati et al., 2010) introduce a selective authorization policy model based on graph theory in order to ensure *read* privilege. 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. That is, (Di Vimercati et al., 2010) proposal is based on the use of both a key agreement algorithm and a key derivation function that enable a key to be derived from another key and a public token. The combination of these two algorithms permits to correctly convert access policies defined by data owners into encryption policies. Later, in 2013, Di Vimercati et al. (di Vimercati et al., 2013) propose another approach to support modification of outsourced data files. The main idea of (di Vimercati et al., 2013) proposal relies on the association of each content with a write tag. The remote server permits a user to perform the write operation on a data 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 the 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) 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 selective 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 order to ensure fine grained access control to outsourced data, several constructions relying on ABE have been proposed (Kaaniche and Laurent, 2017b), (Hur and Noh, 2011),(Yu et al., 2010),(Jahid et al., 2011), (Horváth, 2015), (Huang et al., 2016), (Belguith et al., 2016). Indeed, Hur et al. proposed an access control scheme based on CP-ABE in data outsourcing systems such as cloud computing (Hur and Noh, 2011). Horvath proposed a fine-grained access control scheme for securely sharing data in cloud environments (Horváth, 2015). To guarantee an efficient revocation scheme, this construction relies on an identity based user revocation mechanism to manage access rights . The proposed extension is based on

multiple independent attribute authorities which makes the revocation of specific users (e.g. based on users' identities) from the system is possible without updates of public and private keys. However, these schemes only focus on data sharing, at one single security level and cannot support selective access privileges to outsourced data.

In (Huang et al., 2016), Huang et al. propose a data collaboration scheme, such that authorized users can share data in a collaborative manner. In fact, the data owner encrypts data with respect to a selected access policy based on CP-ABE, while the cooperative user re-encrypts the modified data and signs a collaboration request with his attributes. As such, only the users whose attributes satisfy the access policy can modify outsourced data. (Huang et al., 2016) employs a delegation mechanism based hierarchical ABE, which contains a central authority and a number of independent domains. Each domain holds a domain authority that requests a secret parameter from the higher level authority and generates attribute secret keys for its domain users. The (Huang et al., 2016) proposal introduces a partial decryption and signing construction, where users are able to outsource most of the decryption and signing computation overhead to the service provider.

Afterwards, Khan et al. presented a multi-level access control scheme, proving a single ciphertext over a global access policy (Khan et al., 2016). The data owner can define limited users' privileges to different chunks of the whole data D. However, the enciphering symmetric key sk is defined as a vector of different sub-keys $sk = [s_i]_{i\in[1,n+1]}$, where each s_i is the related encrypting key of m_i, such that $D = \{m_1, m_2, \cdots, m_n\}$ and s_{n+1} is the enciphering key of the whole data D. Thus, the (Khan et al., 2016) construction requires heavy computation and communication costs. Recently, in 2017, Kaaniche and Laurent proposed a multi-level access control scheme based on attribute based mechanisms for e-health applications (Kaaniche and Laurent, 2017a). Their construction permits the enciphering party to encrypt a data file, while specifying an access structure and a certain number of security levels. Thus, a user can decrypt a sub-set of data chunks associated to a security level k if that user's private keys satisfy the sub-set of attributes related to the k-security level. Nonetheless, the (Kaaniche and Laurent, 2017a) proposal does not offer a threshold multi-level access, such that the deciphering entity has to satisfy a precise set of attributes to be able to decrypt a given sub-sets of data blocks w.r.t. a given security level. In addition, the encrypting entity has to encipher each sub-set of data blocks associated with a given security level, using on a different secret.

Although these schemes proposed efficient solutions to protect data contents from unauthorized access, they are still inefficient with multi-level access policies, where users must share the same data content with different access rights to distinct data chunks. In addition, these schemes mainly provide interesting computation and communication costs at the receiving entity side, w.r.t to the decryption procedure, while it is important to focus on the processing overhead at the data owner side, responsible for defining different access policies to different parts of outsourced data files.

2.2 Problem Statement

In real-world data sharing scenarios, different organisations and actors can be involved. The shared data must be protected from unauthorized access while ensuring fine grained access control for different authorized actors. For preserving data confidentiality against malicious users, encryption should be applied while supporting flexible sharing of encrypted data among dynamic groups of users, with fine-grained access control policies.

Let us consider the case of the reviewing process of scientific papers: the website administrator manages access to the encrypted versions of submitted papers with respect to three different groups of users: *reviewers*, *sub-reviewers* and *PC chair*. Indeed, according to their related credentials, some entities can have access to the paper content as well as the name of authors and their personal information, while others may have access to only anonymized versions of the submitted papers, referred to as *blinded papers*. Thus, the aforementioned group of users define several access control policies as follows:

- access to the blinded paper – [((*reviewer* `or` *sub-reviewer*) `and` *research track*) `and` (*computer systems* `or` *network security*)] `or` [(*computer systems* `or` *network security*) `and` *PC chair*] `or` [((*reviewer* `or` *sub-reviewer*) `and` *research track*) `and` *PC chair*] ;
- access to the paper and identifying information – ((*reviewer* `or` *sub-reviewer*) `and` *research track*) `and` (*computer systems* `or` *network security*) `and` (*PC chair*);

To enable access to encrypted data, the available option related to the use of ABE mechanisms is based on the naive computing approach referred to as `NC` in the following. In `NC`, the data owner, i.e; the website administrator, is assumed to create an access structure for each part of the scientific paper which is

then encrypted, with respect to every group of authorized users. Concretely, the blinded version of the paper and the whole paper content including identifying information have to be enciphered with respect to different access structures. However, it is worthy noticeable that (i) these access structures share several leaf nodes, corresponding to redundant required attributes and (ii) each data content is encrypted several times under different access policies. In fact, this use-case points out that there might be overlapping between access structures for the same data share, thus leading to complex access structures. In addition, the management of access policies becomes more complex and the burden of enciphering keys' derivation for every data chunk rises mainly with groups of users having different access privileges. That is why we propose a multi-level access control mechanism for the same data content, based on an ABE aggregate access tree.

2.3 Security and Functional Requirements

The proposed SABE scheme has to fulfill the following security properties and functional features:

- ***data confidentiality*** – even in case of collusions, SABE has to ensure the secrecy of encrypted data contents against malicious entities, namely unauthorized and revoked users.
- ***multi-level fine grained access control*** – SABE should support flexible multi-level security policies among groups of users with different granted privileges.
- ***threshold support*** – SABE should ensure the threshold feature, such that each defined security level may encompass several sub-trees.
- ***low processing and communication costs*** – SABE encrypted data file should be short-sized as the transmission overhead is essential in the emerging infrastructure context. In addition, the encryption and decryption processes should have a low computation cost in order to reduce the impact of heavy cryptographic mechanisms on the efficiency of the intended algorithms.

3 PRELIMINARIES

3.1 Mathematical Background

In this section, we introduce our prerequisites, namely multi-leveled pairing functions, access structures and Lagrange interpolation, defined as follows:

Definition 1. ***(Leveled Multilinear Maps** (Hohenberger et al., 2013) (Garg et al., 2013))*
Let $\mathcal{G}$ be a group generator $\mathcal{G}(1^\kappa,k)$, where κ is a security parameter and k is the number of allowed pairing operations. $\mathcal{G}(1^\kappa,k)$ outputs a sequence of groups such that $\vec{G}=[\mathbb{G}_1,\cdots,\mathbb{G}_k]$, each of prime order $p>2^\kappa$. Let g_i be a canonical generator of $\mathbb{G}_i$. We assume the existence of a set of bilinear maps $\{e_{i,j}:\mathbb{G}_i\times\mathbb{G}_j\longrightarrow\mathbb{G}_{i+j}|i,j\geq 1,i+j\leq k\}$. The map $e_{i,j}$ satisfies the following relation:

$$e_{i,j}({g_i}^a,{g_j}^b)={g_{i+j}}^{ab},\qquad \forall a,b\in\mathbb{Z}_p$$

Definition 2. ***(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.

Definition 3. ***(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 $\delta_j(x)=\prod_{0\leq i\neq j\leq k}\frac{x-x_i}{x_j-x_i}$.

3.2 Cryptographic Assumptions

For our construction, we consider the following complexity assumptions:

- ***Discrete Logarithm Problem (DLP)*** – Let $\mathbb{G}$ be a multiplicative cyclic group of a prime order p, and g is a generator of $\mathbb{G}$. The DLP problem is, given the public element $y=g^x\in\mathbb{G}$, there is no efficient probabilistic algorithm $\mathcal{A}_{DLP}$ that can compute the integer x.
- ***Computational Diffie Hellman Assumption (CDH)*** – Let $\mathbb{G}$ be a group of a prime order p, and g is a generator of $\mathbb{G}$. The CDH problem is, given the tuple of elements (g,g^a,g^b), where $\{a,b\}\overset{R}{\leftarrow}\mathbb{Z}_p$, there is no efficient probabilistic algorithm $\mathcal{A}_{CDH}$ that computes g^{ab}.

4 OVERVIEW

Our *Selective Attribute based Encryption* (SABE) enables a group of users to access different parts of an encrypted data file in a threshold manner with respect to their different granted privileges. The main idea behind SABE relies on ABE such that users' keys

and decryption capabilities are related to the attributes they possess. Indeed, the SABE scheme considers that the plaintext is composed of a set of messages and users' credentials (i.e; certified attributes) settle *which subset* of data blocks may be deciphered. It provides the ability to strike a balance between security and processing demands.

Accurately, the encrypting entity defines an access structure with respect to n attributes, while specifying multi-threshold 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. Note that each security level k_l corresponds to n_l sub-trees that permit to reconstruct a secret key v_l required to decipher the subset of data blocks associated to the security level k_l.

4.1 System Model

A selective attribute-based encryption scheme (SABE) for a message space $\mathcal{M}$ and an access structure space $\mathcal{G}$ relies on four randomized algorithms, defined as follows:

- setup – it is executed the master entity (i.e; attribute authority) to set up the system. The setup algorithm takes as input the security parameter κ and outputs the public parameters pp and the master key mk.
- encrypt – it is performed by the encryptor. The encrypt algorithm takes as input the public parameters pp, an access structure Γ over the universe of attributes $\mathbb{S}$ and the set of security levels $\{k_l\}_{l\in[1,c]}$, where c is the number of security levels and a message $M = \{m_l\}_{l\in[1,c]}$. It encrypts the message M w.r.t. c security levels and outputs a ciphertext $CT = \{\Gamma, \forall k_l : CT_l\}$, such that $l \in [1,c]$ and each security level has to be satisfied by at least n_l root subtrees. This algorithm is performed such that only a user that holds a set of certified attributes w.r.t. a security level k_l that satisfies the access structure Γ can decipher the encrypted message CT_l.
- keygen – it is run by the attribute authority. This algorithm takes as input the public parameters pp, the master key mk and a set of attributes $\mathcal{S}$ and outputs the related secret key sk.
- decrypt – it is performed by the deciphering entity. The decrypt algorithm takes as input the public parameters pp, the ciphertext CT, which contains an access policy Γ, the security level k_l and the secret key sk associated to the set of attributes $\mathcal{S}$. Recall that $\mathcal{S}$ has to satisfy Γ, w.r.t. a security level k_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 κ, all universe descriptions $\mathbb{S}$, all $(\mathtt{pp},\mathtt{mk}) \in \mathsf{setup}(\kappa)$, all $\mathcal{S} \subseteq \mathbb{S}$, all $M \in \mathcal{M}$, all $\Gamma \in \mathcal{G}$, all $sk \in \mathsf{keygen}(\mathtt{pp},\mathtt{mk},\mathcal{S})$, all $k_l \in \mathcal{K}$ ($\mathcal{K}$ is the security level space) and all $CT \in \mathsf{encrypt}(\mathtt{pp},\Gamma,M,\{k_l\}_{l\in[1,c]})$, if $\mathcal{S}$ satisfies Γ w.r.t. a security level k_l, then decrypt $(\mathtt{pp},CT,k_l,sk)$ algorithm outputs m_l.

4.2 Selective CCA-1 Security Model

Let $\Pi = (\mathsf{setup},\mathsf{encrypt},\mathsf{keygen},\mathsf{decrypt})$ be a SABE scheme for a message space $\mathcal{M}$ and an access structure space $\mathcal{G}$. To prove the resistance of SABE against selective chosen ciphertexts attacks, we consider a security game, referred to as $G^{S-CCA}(1^\kappa)$, between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. That is, $\mathcal{A}$ defines a challenge access structure Γ^*, 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 Γ^*. $G^{S-CCA}(1^\kappa)$ is formally defined as follows:

INIT — $\mathcal{C}$ executes the setup algorithm, gives the public parameters pp to $\mathcal{A}$ and keeps secret mk.

QUERIES — $\mathcal{A}$ can repeatedly make any of the following queries for each session j:

- obtain : $\mathcal{A}$ queries for the secret jets $\{sk_j\}_{j\in[1,t]}$ w.r.t. a set of attributes $\{\mathcal{S}_j\}_{j\in[1,t]}$ related to the security level $\{k_{l,j}\}$.
- cordec : $\mathcal{A}$ sends the pair $(CT_j,\mathcal{S}_j)$ and asks for the decryption of a selected ciphertext CT_j, based on the private key associated to the set of attributes $\mathcal{S}_j$. Note that if $\mathcal{C}$ has not previously extracted sk_j related to $\mathcal{S}_j$, then $\mathcal{C}$ does the extraction based on the obtain algorithm and outputs the result of the decryption of CT_j w.r.t. $\{k_{l,j}\}$ selected security levels.

CHALLENGE — $\mathcal{A}$ submits two equal length messages M_0 and M_1, gives the access policy Γ^* and the set of security levels $\{k_l\}^*$, such that none of the previous quiered sets $\{\mathcal{S}_j\}_{j\in[1,t]}$ satisfies Γ^* for $\{k_l\}^*$. Consequently, $\mathcal{C}$ flips a coin b and encrypts M_b under Γ^*, w.r.t. $\{k_l\}^*$. The resulting ciphertext CT^* is then submitted to $\mathcal{A}$.

GUESS — $\mathcal{A}$ outputs a guess b' of b. The output of $G^{S-CCA}(1^\kappa)$ is 1 if and only if $b = b'$.

Definition 4. SABE security w.r.t. $G^{S-CCA}(1^\kappa)$
A SABE scheme Π *is selectively CCA secure (i.e; selectively secure against chosen-ciphertext attacks) for attribute universe* $\mathbb{S}$ *if for all probabilistic polynomial-time adversaries* $\mathcal{A}$*, there exists a negligible function* ε*, such that* $Pr[G^{S-CCA}(1^\kappa) = 1] = \frac{1}{2} \pm \varepsilon$.

5 SABE CONSTRUCTION

In this section, we first introduce the access tree model (subsection 5.1) before detailing our SABE concrete construction (subsection 5.2).

5.1 Access Tree Model

In SABE, we consider that each access structure, referred to as Γ is defined with respect to two levels:

- Level 1 – the first level encompasses the root node and its direct children. The root node is defined as k_l-*out-of-c* security levels. Each security level k_l requires at least p_l subsets of attributes and n_l subtrees of the root node for the reconstruction of the corresponding secret key v_l.
- Level 2 – it 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, as detailed in Bethencourt et al. construction (Bethencourt et al., 2007).

Note that the same notation as (Bethencourt et al., 2007), is used to describe the access tree. Each non-leaf node of Γ is expressed w.r.t. the number of its children num_x and a threshold value t_x, where $1 \leq t_x \leq num_x$. 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 Γ_x the subtree of Γ 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 Γ_x, it is referred to as $\Gamma_x(\mathcal{S}) = 1$.

Hence, Γ is rooted by the root node r. For instance, depending on the number of the attributes l and the number of subtrees n_l rooted by the root node, the user may decrypt CT_l, w.r.t. k_l, such that:

$$\text{if } \exists x_{i i\in[1,n_l]} \text{ such that } \Gamma_{x_i}(\mathcal{S}) = 1, \qquad \text{then } \Gamma_{k_l}(\mathcal{S}) = 1$$

where Γ_{k_l} is the access tree for the security level k_l.

5.2 Concrete Construction

SABE construction is based on the following algorithms:

$\texttt{setup}(\kappa)$ – this algorithm first selects leveled 4-linear maps. In fact, it defines two symmetric pairing functions $\hat{e}_1$ and $\hat{e}_2$, such that $\hat{e}_1 : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ and $\hat{e}_2 : \mathbb{G}_2 \times \mathbb{G}_2 \to \mathbb{G}_3$, where $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_3$ are three multiplicative groups of prime order p. It also selects three random generators g, h and f of $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_3$, respectively, such as $\hat{e}_1(g,g) = h$ and $\hat{e}_2(h,h) = f$. The $\texttt{setup}$ algorithm defines for each security level $k_i \in \mathcal{K}$ (i.e; $\mathcal{K}$ is the security level universe), an element X_{k_i} such as $\forall k_i, X_{k_i} = \hat{e}_2(h,h)^{\alpha_i}$, where $i \in [1,n]$ $\alpha_i \in \mathbb{Z}_p$ and n is the maximum number of security levels. The public parameters pp are defined as follows:

$$\texttt{pp} = \{\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \hat{e}_1, \hat{e}_2, g, h, f, p, n, \{X_i\}_{i\in[1,n]}\}$$

The master key $\texttt{mk}$ is the set of generators $\{h^{\alpha_i}\}_{i\in[1,n]}$.

$\texttt{encrypt}(\texttt{pp}, \Gamma, M, \{k_l\}_{l\in[1,c]})$ – it first selects a polynomial q_x for each node x and sets the degree d_x of each polynomial, 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).

Let q_r be the polynomial associated to the root node and defined as $q_r(x) = s + a_1 x + \cdots + a_{d_r} x^{d_r}$, where $d_r = t_r - 1$. In the following, we denote by a_0 the secret s.

Subsequently, for each security level k_l, the $\texttt{encrypt}$ algorithm calculates $t_r - n_l$ auxiliary constants such as:

$$\Sigma_{k_l} = \{{X_{k_l}}^{-a_1}, \cdots, {X_{k_l}}^{-a_{t_r - n_l}}\}$$

The ciphertext is presented as follows:

$$\begin{aligned} CT \quad = \{\Gamma, \quad & \forall k_l : \tilde{C}_{k_l} = m_l \cdot X_{k_l}^s, \Sigma_{k_l}, \\ & \forall y : C_y = g^{q_y(0)}, C'_y = \mathcal{H}(\texttt{att}(y))^{q_y(0)}\} \end{aligned}$$

where Y is the set of leaf nodes and $\mathcal{H}$ is a hash function, such that $\mathcal{H} : \{0,1\}^* \to \mathbb{G}_1$.

$\texttt{keygen}(\texttt{pp}, \texttt{mk}, \mathcal{S})$ – it chooses a random $r \in \mathbb{Z}_p$ and a set of random values $\{r_j\}$ (i.e; j is the number of attributes of $\mathcal{S}$). The private key sk associated to $\mathcal{S}$ is defined as:

$$\begin{aligned} sk \quad = \{ \quad & \forall k_i \in \mathcal{K} : D_{k_i} = h^{r^{-1}\alpha_i}, \\ & \forall a_j \in \mathcal{S} : D_j = g^r \cdot \mathcal{H}(j)^{r_j}, D'_j = g^{r_j}\} \end{aligned}$$

$\texttt{decrypt}(\texttt{pp}, CT, k_l, sk)$ – for the decryption algorithm, we assume that the decryptor satisfies the k_l-security level with n_l sub-trees of Γ being satisfied. This latter proceeds as follows:

- Level 2 – it works in a recursive manner, relying on the *DecryptNode* function as presented in (Bethencourt et al., 2007), resulting in $F_x = \hat{e}_1(g,g)^{rq_x(0)}$, for each node x that belongs to Level 2 of Γ.
- Level 1 – Let S_r be the set of a n_l-sized set of child nodes x of the root node. Referring to the generalization of Shamir's threshold scheme (Shamir,

) (Hassler et al., 1993), the decryption algorithm computes two vectors $\vec{y}$ and $\vec{a}$, such as

$$\vec{y} = \{y_i\}_{i\in[1,n_l]} = \{q_r(\texttt{index}(x_i))\}_{i\in[1,n_l]} = \{q_i(0)\}_{i\in[1,n_l]}$$

and $\vec{a} = \{a_j\}_{i\in[0,d_r]}$, where a_j are the coefficients of the root polynomial of degree d_r.

Then, the decrypting entity defines a matrix $U = \{U_{ij}\} = \{{u_i}^j\}$, such as $u_i = \texttt{index}(x_i)$ and $\vec{y} = U \cdot \vec{a}$.

Afterwards, the algorithm takes the first and the last $n_l - 1$ columns of U and creates a sub-matrix U_s such that $U_s = \{{u_i}^j\}$, where $i = 1, \cdots, n_l$ and $j = 0, t_r - n_l + 1, t_r - n_l + 2, \cdots, d_r$. Afterwards, the decrypt algorithm computes the inverse matrix of U_s, referred to as ${U_s}^{-1} = \{v_{ij}\}$. The inverse matrix is then used to form the following modified system of equations: ${U_s}^{-1} \cdot \vec{y} = {U_s}^{-1} \cdot U \cdot \vec{a}$. The first equation of this system is of the form $s_{k_l} = a_0 + a_1\lambda_1 + \cdots + a_{t_r-n_l}\lambda_{t_r-n_l}$, where $s_{k_l} = \sum_{j=1}^{n_l} y_j v_{1j}$.

To extract the deciphering key, the decrypt algorithm computes $F_{R_{k_l}}$ such as:

$$F_{R_{k_l}} = \prod_{y_k\in S_r} [\hat{e}_1(g,g)^{rq_k(0)}]^{v_{1k}} \quad (1)$$

$$= \hat{e}_1(g,g)^{\sum_{y_k\in S_r} rq_k(0)v_{1k}} \quad (2)$$

$$= \hat{e}_1(g,g)^{rs_{k_l}} \quad (3)$$

The decrypt algorithm can now decrypt the ciphertext with respect to the k_l-security level, such as:

$$\frac{\tilde{C}_{k_l}}{\hat{e}_2(D_{k_l}, F_{R_{k_l}}) \prod_{k\in[1,t_r-n_l], z_k\in\Sigma_{k_l}} {z_k}^{\lambda_k}} = \frac{m_l \cdot X_{k_l}^{s}}{{X_{k_l}}^{s}} = m_l \quad (4)$$

6 SECURITY ANALYSIS

In this section, we first prove the correctness of our SABE scheme, in section 6.1. Then, we discuss the security of our proposed scheme with respect to the security model detailed in section 4.2.

6.1 SABE Correctness

The correctness of our proposition relies on the correctness of Equation 5:

$$\frac{\tilde{C}_{k_l}}{\hat{e}_2(D_{k_l}, F_{R_{k_l}}) \prod_{k\in[1,t_r-n_l], z_k\in\Sigma_{k_l}} {z_k}^{\lambda_k}} \stackrel{?}{=} m_k, \quad (5)$$

where $F_{R_{k_l}} = \prod_{y_k\in S_r} [\hat{e}_1(g,g)^{rq_k(0)}]^{v_{1k}}$.

Upon receiving the ciphertext CT, the decrypting entity proceeds as follows based on two levels:

- Level 2 – the decrypt algorithm works in a recursive manner, relying on the algorithm *DecryptNode* as presented in (Bethencourt et al., 2007). For each non-leaf node x, having z child nodes, the *DecryptNode* algorithm outputs F_x such as :

$$F_x = \prod_{z\in S_x} {F_z}^{\delta_{i,S'_x(0)}} = \hat{e}_1(g,g)^{rq_x(0)}$$

  where: S_x is an arbitrary k_x sized set of child nodes z such that $F_z \neq \perp$, $i = \texttt{index}(z)$ and $S'_x = \{index(z) : z \in S_x\}$. We note that if no such S_x exists then the node is not satisfied and the function returns $\perp$.
- Level 1 – after executing the *DecryptNode* algorithm for all child-nodes of the root node, the decrypt algorithm proceeds as follows: for the first level, we suppose that the deciphering entity satisfies the k_l-security level. Recall that for each security level k_l, S_r is the n_l-sized set of child nodes x of the root node r. To extract the deciphering key, the decrypting entity first defines the vectors $\vec{a}$ and $\vec{y}$ and the matrices U, U_s and ${U_s}^{-1}$, with respect to k_l as explained in section 5. Then, it defines the equation $s_{k_l} = a_0 + a_1\lambda_1 + \cdots + a_{t_r-n_l}\lambda_{t_r-n_l}$, where $s_{k_l} = \sum_{j=1}^{n_l} y_j v_{1j}$ and computes $F_{R_{k_l}}$ as detailed in Equation 1.

For ease of presentation, we denote by ® the quantity $\frac{\tilde{C}_{k_l}}{\hat{e}_2(D_{k_l}, F_{R_{k_l}}) \prod_{k\in[1,t_r-n_l], z_k\in\Sigma_{k_l}} {z_k}^{\lambda_k}}$. Finally, the decrypt algorithm deciphers the ciphertext with respect to the k_l-security level as follows:

$$\begin{aligned}
® &= \frac{\tilde{C}_{k_l}}{\hat{e}_2(h^{r^{-1}\alpha_l}, \hat{e}_1(g,g)^{rs_{k_l}}) \prod_{k\in[1,t_r-n_l], z_k\in\Sigma_{k_l}} {z_k}^{\lambda_k}} \\
&= \frac{\tilde{C}_{k_l}}{\hat{e}_2(h,h)^{\alpha_l s_{k_l}} \hat{e}_2(h,h)^{-\alpha_l \sum_{k\in[1,t_r-n_l]} a_k\lambda_k}} \\
&= \frac{\tilde{C}_{k_l}}{\hat{e}_2(h,h)^{\alpha_l (s_{k_l} - \sum_{k\in[1,t_r-n_l]} a_k\lambda_k)}} \\
&= \frac{m_l \cdot X_{k_l}^{a_0}}{{X_{k_l}}^{a_0}}, \qquad (a_0 = s) \\
&= m_l
\end{aligned}$$

6.2 Selective CCA-1 Security

The resistance of the SABE scheme against is selective chosen ciphertexts attacks relies on Theorem 6.1.

Theorem 6.1. *Our SABE scheme is secure against selective non-adaptive chosen ciphertext attacks in the Generic Group Model (GGM), under the DLP and the CDH assumptions, with respect to the* $G^{S-CCA}(1^\kappa)$ *security game.*

Proof. One of the main challenges to design our SABE scheme was to prevent collusion attacks among users. Hence, as our scheme is based on the CP-ABE construction of Bethencourt et al. (Bethencourt et al., 2007), it randomizes, in the same way, users' private keys such that they cannot be combined. In fact, each secret element D_j, related to an attribute j, encompasses a random value r associated to the user, and r_j related to the attribute j, which prevents colluding users to override their rights. Subsequently, SABE is resistant to collusion attacks. In addition, to decrypt a ciphertext w.r.t. a security level k_l, $\mathcal{A}$ must recover $X_{k_l}{}^s = \hat{e_2}(h,h)^{\alpha_l \cdot s}$, where the secret sharing key s is embedded in the ciphertext. For this purpose, $\mathcal{A}$ has to retrieve the corresponding $\tilde{C}_{k_l}$ and the related private key element D_{k_l} from the user's private key.

To prove that our scheme is secure against selective non-adaptive chosen ciphertext attacks, we first consider that $\mathcal{A}$ is running the G^{S-CCA} experiment with an entity $\mathcal{B}$. This latter is running the $Exp_{\mathcal{B}}$ Bethencourt et al. security game (Bethencourt et al., 2007), with $\mathcal{C}$. The objective of this proof is to show that the advantage of $\mathcal{A}$ to win the $G^{S-CCA}(1^\kappa)$ security game is equivalent to the advantage of $\mathcal{B}$ to win the Bethencourt et al. security game (Bethencourt et al., 2007). Hereafter, $\mathcal{A}$ and $\mathcal{B}$ proceed as follows:

INIT — $\mathcal{C}$ executes `setup`, gives `pp` to $\mathcal{B}$ and keeps secret `mk`. Consequently, $\mathcal{B}$ sends `pp` to $\mathcal{A}$.

QUERIES — $\mathcal{B}$ sets an empty table T and repeatedly make the following queries, such that for each session j:

- `obtain` : $\mathcal{A}$ queries $\{sk_j\}_{j\in[1,t]}$ w.r.t. a set of attribute $\{\mathcal{S}_j\}_{j\in[1,t]}$ associated to $\{k_{l,j}\}$ security levels. That is, $\mathcal{B}$ uses $\mathcal{C}$ to derive and send the queried secret keys to $\mathcal{A}$. The private keys $\{sk_j, \mathcal{S}_j\}_{j\in[1,t]}$ are returned to $\mathcal{B}$. Subsequently, $\mathcal{B}$ sets a new entry with the pair $\{sk_j, \mathcal{S}_j\}_{j\in[1,t]}$ and returns $\{sk_{j,GID}\}_{j\in N}$ to $\mathcal{A}$.
- `cordec` : $\mathcal{A}$ sends $(CT_j, \mathcal{S}_j)$ and queries for the decryption result of the ciphertext CT_j, w.r.t. $\mathcal{S}_j$. Indeed, $\mathcal{B}$ checks if an entry sk_j for $\mathcal{S}_j$ does exist in T w.r.t. $\{\Gamma^*, k_{l,j}\}$ and retrieves sk_j. Then, $\mathcal{B}$ deciphers CT_j and sends the result to $\mathcal{A}$.

CHALLENGE — $\mathcal{A}$ submits two equal length messages M_0 and M_1 and gives the access policy Γ^* and the set of security levels $\{k_l\}^*$, such that none of the previous sets $\{\mathcal{S}_j\}_{j\in[1,t]}$ satisfies Γ^* w.r.t. $\{k_l\}^*$. Consequently, $\mathcal{B}$ selects $\Gamma_{\mathcal{B}}$ such that $\Gamma_{\mathcal{B}} \subseteq \Gamma^*$. We have to emphasize 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 $\mathcal{A}$. $\mathcal{C}$ flips a coin b, encrypts M_b under $\Gamma_{\mathcal{B}}$ and sends the resulting ciphertext $\{CT_b\}^*$ to $\mathcal{A}$.

We distinguish two different cases for the $G^{S-CCA}(1^\kappa)$ game, as follows:

Case 0: we set only one security level $k_l{}^*$, during the INIT phase such as the public parameter n defined by $\mathcal{C}$ is equal to 1. That is, all queried secret keys are associated to the set of attributes $\mathcal{S}_i$ that decrypt ciphertexts, encrypted w.r.t. $k_l{}^*$, for each session i. As such, we notice that the two first steps INIT and QUERIES of the $G^{S-CCA}(1^\kappa)$ are similar to the (Bethencourt et al., 2007) experiment. Additionally, the challenge access structure selected by $\mathcal{A}$ is equivalent to the access policy defined by $\mathcal{B}$ (i.e; $\Gamma_{\mathcal{B}} = \Gamma^*$, where all sub-trees of Γ^* have to be included in $\Gamma_{\mathcal{B}}$).

Case 1: during the INIT phase, $\mathcal{C}$ defines several security levels, where $n \neq 1$. Thus, we point out two sub-cases as follows:

- **Case 1-a :** during QUERIES, a single security level is selected, such that all queried `cordec` have to return response w.r.t. the pre-fixed $k_l{}^*$, whereas queried private keys are encoded under different security levels, for each session i. That is, $\mathcal{B}$ has to select $\Gamma_{\mathcal{B}}$ where subtrees ST_i required to satisfy each selected security level $\{k_l\}^*$ of Γ^* have to be included in $\Gamma_{\mathcal{B}}$.
- **Case 1-b:** during QUERIES, the attacker $\mathcal{A}$ sends `cordec` queries to the challenger $\mathcal{C}$ with respect to different security levels $\{k_{l,i}\}$ for each different session i and a ciphertext CT_i. We note that CT_i may be encoded under different security levels. During the challenge phase, $\mathcal{A}$ sends two different messages M_0 and M_1 and asks $\mathcal{C}$ to encipher the selected message under a security level $k_l{}^*$ that has never been queried during QUERIES. Hence, $\mathcal{B}$ chooses $\Gamma_{\mathcal{B}}$ such that identified subtrees ST_i required to satisfy the security level $\{k_l\}^*$ of Γ^* have to be included in $\Gamma_{\mathcal{B}}$.

In the $G^{S-CCA}(1^\kappa)$ security game, including **Case 0** and **Case 1**, the challenge ciphertext has a component $\tilde{C}_{k_l}$ which is either $M_0 \cdot X_{k_l}{}^s$ or $M_1 \cdot X_{k_l}{}^s$ (i.e; s is the enciphering secret key). Hence, we consider a modified game, defined in (Bethencourt et al., 2007), in which $\tilde{C}_{k_l}$ is either $\hat{e_2}(h,h)^{\alpha_l \cdot s}$ or $\hat{e_2}(h,h)^{\theta}$, where θ is selected uniformly at random. $\mathcal{A}$ has to guess which is the case. The adversary's advantage is obviously equal to ε in the original security game. In fact, no efficient adversary $\mathcal{A}$ can output $b' = b$,

in the security experiment $Exp_{\mathcal{A}}(1^{\kappa})$, better than a random guess. Recall that a random guess b' by $\mathcal{A}$ is equal to b, with a probability $1/2$. Thus, we call ε the advantage of $\mathcal{A}$ if $b' = b$ with the probability $1/2 \pm \varepsilon$. As such, in the modified game, the adversary advantage is at least $\varepsilon/2$, while considering two equivalent sub-cases: when $\mathcal{A}$ has to distinguish between $M_0 \cdot {X_{k_l}}^s$ and $\hat{e_2}(h,h)^{\theta}$ and when $\mathcal{A}$ has to distinguish between $M_1 \cdot {X_{k_l}}^s$ and $\hat{e_2}(h,h)^{\theta}$. Hereafter, we consider $\mathcal{A}$'s advantage in the modified game.

$G^{S-CCA}(1^{\kappa})$ **Game Analysis** – As introduced in (Bethencourt et al., 2007) (Boneh et al., 2005), each element of $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_3$ is encoded as a unique random. The encoding properties of elements in $\mathbb{G}_i$ is presented by $\xi_{0,i} : \mathbb{Z}_p \to \{0,1\}^*$ that maps all $a \in \mathbb{Z}_p$ to the representation $\xi_{0,i}(a)$ of $g^a \in \mathbb{G}_i$ and $\xi_{T,i} : \mathbb{Z}_p \to \{0,1\}^*$ that maps all $a \in \mathbb{Z}_p$ to the representation $\xi_{T,i,j}(a)$ of $\hat{e}_j(g,g)^a \in \mathbb{G}_i$ ($i \in \{1,2,3\}$ and $j \in \{1,2\}$). The adversary communicates with the oracles to perform actions in $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_3$, $\hat{e}_1$ and $\hat{e}_2$ based on $\xi_{0,i}$ and $\xi_{T,i,j}$ representations.

For **Case 0**, during the INIT phase, $\mathcal{C}$ sets $n = 1$, chooses $\alpha \in \mathbb{Z}_p$ and sends the public parameters $\xi_{0,1}(1) = g, \xi_{0,2}(1) = h$ and $\xi_{T,3,2}(\alpha)$ to the adversary. Subsequently, $\mathcal{B}$ initializes an empty table T. Then, during QUERIES, $\mathcal{A}$ queries several times `obtain` and `cordec` algorithms. For each `obtain` query, $\mathcal{C}$ 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. In the sequel, for a session j, `obtain` selects a random $r^{(j)}$, computes $D_k = h^{\alpha/r^{(j)}}$. Then, for each $i \in \mathcal{S}_j$, it provides $D_i = g^{r^{(j)} + t_i r_i^{(j)}}$ and $D'_i = g^{r_i^{(j)}}$. Then, $\mathcal{B}$ sets these computed values as new entries in T, and sends them to $\mathcal{A}$.

Then, for the `cordec` oracle, $\mathcal{A}$ submits $(\mathcal{S}_j, CT_j)$ and asks for the decryption of CT_j, w.r.t. the pre-defined security level ${k_l}^*$. Hence, $\mathcal{B}$ performs the decryption of $CT^{(j)}$ for each session j and provides a message M_j or an error message if the set of attributes does not satisfy the access policy w.r.t. the pre-defined security level. Clearly, the SABE construction is close to the CP-ABE construction proposed by Bethencourt et al. (Bethencourt et al., 2007). The main difference consists in the derivation of the embedded shared secret, obfuscated in the exponent of the related pairing function. Indeed, in addition to Lagrange interpolation proposed by the (Bethencourt et al., 2007) scheme, in our proposal, the processing of Level 1 of an access structure Γ, with respect to ${k_l}^*$ requires pointing out the correct public element $X_{{k_l}^*} = \hat{E}_2(h,h)^{\alpha_k}$, which is different for each level.

As such, to prove that **Case 0** is close to the (Bethencourt et al., 2007) construction, we consider an *absurdum* reasoning, where $\mathcal{A}$ can win $Exp_{\mathcal{A}}$ with non-negligible probability. To do so, we consider that the root polynomial in $Exp_{\mathcal{B}}$ is equal to $q_{r,Exp_{\mathcal{B}}}(x) = \sum_{i=0}^{p} a_i x^i$, where $a_0 = s$. In the sequel, we easily verify that 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 $\sum_{j=1}^{p} \sum_{i=0}^{p} a'_i {x_j}^i = s$.

To do, we consider $p-1$ random values a'_i, where $i \in [1, p-1]$. As such, we have the following equality:

$$\sum_{j=1}^{p} \sum_{i=0}^{p} a'_i {x_j}^i = s = \sum_{j=1}^{p} \sum_{i=0}^{p-1} a'_i {x_j}^i + \sum_{j=1}^{p} a'_p {x_j}^p \qquad (6)$$

Following Equation 6, we notice that:

$$a'_p = \frac{s - \sum_{j=1}^{p} \sum_{i=0}^{p-1} a'_i {x_j}^i}{\sum_{j=1}^{p} a'_p {x_j}^p} \qquad (7)$$

From Equation 6 and Equation 7, it is worth noticing that the polynomial $q_{r,Exp_{\mathcal{A}}}$ exists. Consequently, $\mathcal{A}$ receives the challenge ciphertext CT_b. If the adversary $\mathcal{A}$ can win the $Exp_{\mathcal{A}}$ experiment with a non-negligible probability, then $\mathcal{A}$ can guess b' which is therefore sent to $\mathcal{B}$. As such, $\mathcal{B}$ 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. Additionally, for **Case 0** of $G^{S-CCA}(1^{\kappa})$, the INIT, QUERIES and CHALLENGE phases relies on one single security level, such that M_b contains one single data block w.r.t. the k_l security level, this first case follows the selective CCA-security of the Bethencourt et al.'s CP-ABE scheme (Bethencourt et al., 2007). In the sequel, the advantage of $\mathcal{A}$ 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 $G^{S-CCA}(1^{\kappa})$ game.

For **Case 1**, during the INIT phase, $\mathcal{C}$ defines n and sends the public parameters $\xi_{0,1}(1) = g, \xi_{0,2}(1) = h$ and $\forall j \in [1,n], \xi_{T,3,2}(\alpha_j)$ to $\mathcal{B}$. Let us notice that **Case 1** can be modeled in *multi-user setting*, such that there are multiple public keys and multiple challenge ciphertexts that can be dependent. In our case, the public keys correspond to the security levels' public parameters X_{k_i} and challenge ciphertexts consist of the different chuncks of the challenge message $M^* = \{{m_l}^*\}_{l \in [1,c^*]}$. Hence, **Case 1** is a generalization of selective CCA security in the multi-user setting and the adversary $\mathcal{A}$, having access to n different public keys, can perform multiple *Left-or-Right* queries. These challenge ciphertexts must 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.

During the CHALLENGE phase, we distinguish **Case 1-a**, where $\mathcal{A}$ sends two messages M_0 and M_1, the access structure Γ^* and a security level ${k_l}^*$; and **Case 1-b** where $\mathcal{A}$ sends two messages M_0 and M_1 and the access structure Γ^*, which has to be encrypted with respect to a set of security levels $\{{k_l}^*\}$.

First, for **Case 1-a**, as the encryption is performed w.r.t. a single security level ${k_l}^*$, then the challenge message has to be composed of one single data block and the cardinal of the set of auxiliary constants is equal to 0 (i.e; $|\Sigma_{k_l}| = 0$). As the sequel, the progress of the CHALLENGE phase of **Case 1-a** is similar to **Case 0**, leading us to an adversary advantage equal at most to $O(\frac{q^2}{p})$.

Second, for **Case 1-b**, when $\mathcal{A}$ asks for the encryption of the challenge message, $\mathcal{C}$ does the following. $\mathcal{C}$ first chooses a random $s \in \mathbb{F}_p$ and uses the linear secret sharing scheme associated with the access structure Γ^* to construct the shares σ_k and λ_i of s for all relevant sub-trees k and attributes i, respectively. As detailed in (Bethencourt et al., 2007), both λ_i and σ_k shares have to be chosen uniformly and independently at random values from $\mathbb{F}_p$. Subsequently, the simulation chooses μ randoms $\theta_l \in \mathbb{F}_p$, where $l \in [1,\mu]$ and μ is the cardinal of the set of security levels $\{{k_l}^*\}_{l\in[1,\mu]}$. Finally, $\mathcal{C}$ outputs the encryption of the challenge message such that: for each security level k_l, we have $\tilde{C}_{k_l} = \hat{e}_2(h,h)^{\theta_l}$ and $\Sigma_{k_l} = \{\xi_{T,3,2}(\alpha_l)^{-\sigma_t}\}$, where $t \in [1, t_r - n_l]$ and $l \in [1,\mu]$ (cf; section 5.2). 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 satisfy Γ^* w.r.t. any security level, then $\mathcal{C}$ does not issue the key. Similarly, if $\mathcal{A}$ asks for Γ^*, w.r.t. 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 $O(\mu\frac{q^2}{p})$, due to the randomness of the choice of variable values in the simulation.

In fact, knowing that this randomization is required for generating the ciphertext, $\mathcal{A}$ is led to break the CDH assumption. The $G^{S-CCA}(1^\kappa)$ security is then considered with respect to the CDH-assumption. Intuitively, here, $\mathcal{B}$ relies on the capabilities of $\mathcal{A}$ to forge a ciphertext C_i obtained from interactions with $\mathcal{C}$ in $Exp_{\mathcal{A}}$. Since $\mathcal{A}$ and $\mathcal{B}$ algorithms are based on coin tosses, the first condition for $\mathcal{B}$ to succeed is that it does not abort the game before $\mathcal{A}$. In (Ahn et al., 2012), this probability has been shown to be $\frac{1}{e}$ if the probability for the coin flipping to be 0 is $\frac{1}{\xi_c+1}$, where ξ_c is the number of ciphertexts' queries. The other condition of the attacker is to be able to identify the value of λ_i for each C_i or to guess the value θ_l related to a security level k_l. After a time t', this probability is equal to $\frac{1}{\xi_c+1}$. This shows that $\mathcal{B}$ can violate the CDH-assumption with a probability equal to $\frac{\varepsilon}{e(\xi_c+1)}$ which conflicts the fact that $\mathbb{G}_1$ is a (t,ε)-CDH group. Indeed, the adversary' view in this simulation is identically distributed for all security levels. In fact, despite the multi-user setting environment, the encryptions of data blocks of the challenge message M_b are completely independent, thanks to the use of the encoding function $\xi_{T,3,2}$. As such, **Case 1-b** can be considered as μ random repetitions of **Case 0** simulation, with respect to μ security levels.

As such, we prove that our SABE construction is secure against selective non-adaptive chosen ciphertexts attacks in the Generic Group Model (GGM), under the DLP and the CDH assumptions, with respect to $Exp_{\mathcal{A}}(1^\kappa)$ experiment. □

7 SABE PERFORMANCES DISCUSSION

In this section, we discuss the functional properties as well as the processing, communication and storage cost of our SABE construction compared to the naive computing approach NC introduced in section 2.2 and two of the most closely related schemes (Kaaniche and Laurent, 2017a), (Khan et al., 2016). That is, we first present a theoretical performance analysis, based on mathematical operations' complexities, in subsection 7.1. Then, we discuss measurement results of different mathematical operations' computation and present an estimation of the different SABE algorithms calculation times, relying on the cpabe toolkit [1]. Finally, we discuss the support of the threshold feature when designing multi-level access control schemes in subsection 7.3.

7.1 Theoretical Performance Analysis

For our theoretical performance analysis, we assess the theoretical complexity where the encrypting entity has to create k different access control policies, associated to k different security levels, for the naive approach. To this purpose, we define, in Table 1 the following costs:

Table 2 presents detailed computation, communication and storage overhead comparison, based on the processing cost of the encryption and decryption algorithms and the size of the ciphertext. Note that the

[1] http://acsc.cs.utexas.edu/cpabe/index.html

Table 1: Notations.

Notations	Description
γ_M	cost of two group elements' multiplication in a multiplicative group
γ_E	cost of an exponentiation in a multiplicative group
γ_e	cost of a symmetric pairing function computation
γ_ξ	cost of a leveled multi-linear map computation
$\|MT\|$	size of an aggregate access tree, referred to as master tree
$\|AT\|$	size of an access tree for an access policy k
η_k	number of auxiliary elements associated to a security level k
Y_{MT}	number of leaves of the master access tree
$\eta_{x_{MT}}$	number of interior nodes of the master access tree
Y_{AT}	number of leaves of an access tree, with respect to an access policy k
$\eta_{x_{AT}}$	number of interior nodes of an access policy associated to a security level k
$\|E\|$	size of a multiplicative group element

communication and storage overhead are both referring to the size of the ciphertext.

It is worth noticing that the size of the master access tree, proposed in our SABE construction, is lower than the size of the set of access trees related to k access policies introduced by the naive approach NC. This is mainly due to the involved number of attributes (access tree leaves), that should be duplicated for different access trees in NC. Obviously, the number of leaves of the master tree Y_{MT} is lower than the sum of leaves of access trees related to k access structures of NC, such that $Y_{MT} \leq \sum_k Y_{AT_k}$. As such, our SABE approach presents competitive communication and storage costs, compared to the NC approach. Notice that the ciphertext size of our SABE scheme is comparable to the (Khan et al., 2016) proposal. Otherwise, our SABE construction presents larger ciphertext-sizes compared to (Kaaniche and Laurent, 2017a), due to additional auxiliary elements associated with security levels. Indeed, these group elements permit our SABE scheme to provide the threshold feature, which is not supported by either (Kaaniche and Laurent, 2017a) nor (Khan et al., 2016) schemes.

In addition, based on the NC 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, during the encryption phase. Consequently, the processing and communication costs introduced by the SABE approach are considerably optimized, where the number of polynomials, that have to be assigned to each node of an access tree, is reduced compared to NC, thanks to the use of an aggregate access structure. For the decrypt algorithm, SABE introduces merely the same computation cost generated by the decryption procedure of (Kaaniche and Laurent, 2017a) and (Khan et al., 2016), except with only one extra multi-linear map calculation. This processing overhead remains interesting and attractive thanks to the support of threshold feature offering more flexibility for access policies' definition, as discussed in subsection 7.3.

7.2 Numerical Performance Analysis

Referring to the cpabe toolkit [2] proposed in (Bethencourt et al., 2007), the computation costs of the key generation, encryption and decryption algorithms are mainly depending on the number of attributes. The cpabe toolkit provides a set of programs implementing CP-ABE schemes (Bethencourt et al., 2007), using the PBC library [3] for the algebraic operations. The code is split into two packages, libbswabe (i.e; a library implementing the core cryptographic operations) and cpabe (i.e; higher level functions and user interface), proving four main algorithms, namely cpabe − setup, cpabe − keygen, cpabe − enc and cpabe − dec. To give some information on the performance achieved by our scheme, some experiments are conducted, for several mathematical operations (i.e; exponentiation, multiplication and pairing functions) on an Intel *E5-1650-v3* 6 cores, where each core relies on 1200 MHz. We set the security parameter to $\lambda = 112$, based on the super-singular curve $y^2 = x^3 + x$ over a finite field and we run 1000 samples for getting an average duration.

Our measurements show that the computation of a symmetric pairing function requires approximately 6 ms, exponentiations and multiplications take about 1.2 ms and 0.5 ms, respectively. In addition, as stated above, based on the cpabe toolkit, the calculation of the cpabe − keygen algorithm, similar to the keygen algorithm of our proposed algorithm, is perfectly linear to the number of attributes. It takes about 1 second for generating a private key containing around 30 attributes (Bethencourt et al., 2007). Moreover, it is worth noticing that our encrypt algorithm follows the (Bethencourt et al., 2007) construction, except for the generation of auxiliary elements, depending on the execution of exponentiations and multiplications in a multiplicative group. In the sequel, the running time of the decrypt algorithm is also almost perfectly linear to the number of attributes involved in the access policy, where the execution time of cpabe − enc algorithm takes 1.5 seconds for 60 gates (i.e; leaf nodes).

7.3 Threshold Support

Unlike (Kaaniche and Laurent, 2017a), (Khan et al., 2016) schemes and the NC approach, our SABE construction provides the threshold feature, as shown in Table 2. This property aims at offering more flexibility while defining multiple access policies.

[2] http://acsc.cs.utexas.edu/cpabe/index.html

[3] https://crypto.stanford.edu/pbc/

Table 2: Theoretical Performance Comparisons.

Scheme	*Processing cost*		*Ciphertext size*	*Threshold support*
	encrypt	decrypt		
(Kaaniche and Laurent, 2017a)	$k\gamma_M + 2k(1+Y_{MT})\gamma_E$	$(\eta_{x_{MT}} + Y_{AT})[2\gamma_e + \gamma_E + \gamma_M] + (z\eta_{x_{MT}} + 2)\gamma_M + \gamma_e$	$\{\|MT\|, 2(k+Y_{MT})\|E\|\}$	✗
(Khan et al., 2016)	$2(k+1)\gamma_M + 2(k+1)(1+Y_{MT})\gamma_E$	$(\eta_{x_{MT}} + Y_{AT})[2\gamma_e + \gamma_E + \gamma_M] + (z\eta_{x_{MT}} + 2)\gamma_M + \gamma_e$	$\{\|MT\|, 2(k+1)\|E\| + 2Y_{MT}\|E\|\}$	✗
NC	$k\gamma_M + 2k(1+Y_{AT})\gamma_E$	$(\eta_{x_{AT}} + Y_{AT})[2\gamma_e + \gamma_E + \gamma_M] + (z\eta_{x_{AT}} + 2)\gamma_M + \gamma_e$	$\{k\|AT\|, 2k(1+Y_{AT})\|E\|\}$	✗
SABE	$k\gamma_M + 2[k(2+\eta_k) + Y_{MT}]\gamma_E$	$(\eta_{x_{MT}} + Y_{AT})[2\gamma_e + \gamma_E + \gamma_M] + (z\eta_{x_{MT}} + 2)\gamma_M + \gamma_e$	$\{\|MT\|, [k(2+\eta_k) + 2Y_{MT}]\|E\|\}$	✓

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). Hence, in such use-cases, the NC approach is much more appropriate in terms of processing (i.e; encryption process) and communication costs, mainly due to $\{\eta_k\}_c$, the set of auxiliary elements that has to be associated to each message m_k enciphered with respect to a security level k, where $k \in [1, c]$.

Furthermore, it is still inappropriate for hierarchical scenarios that require restrictive privileges, such as for 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}.

Finally, thanks to the use of a threshold approach for access policies' definition, SABE presents interesting computation, communication and storage overhead in collaborative use cases, where each security level requires the definition of several combinations of sub-access policies.

8 POSSIBLE APPLICATIONS

Our SABE scheme comes as an alternative that aims at providing sufficient security with an important gain in processing and communication overhead. In the following, we discuss a set of potential applications for selective attribute based encryption mechanisms, namely monitoring encrypted content, database search as well as mobile communications.

Monitoring Encrypted Content – this case highlights situations when encrypted contents are usable for monitoring. For example, several applications such as military images, media audience or video surveillance where some faces have to be scrambled, require identifying partially encrypted data files with no need to decrypt the whole contents. Indeed, our SABE mechanism permits to decrypt parts of the enciphered contents with respect to assigned credentials under different security levels.

Database Search – nowadays, databases hold a critical concentration of sensitive information and their volume is increasing very quickly. In such cases, database outsourcing is becoming increasingly popular. Clients' databases are stored at an external service provider that should provide mechanisms for securing access to these contents, mainly by encrypting the outsourced data. However, the problem consists in ensuring a selective retrieval over encrypted data. Several existing access control mechanisms, designed for distributed applications, operate on client-server architectures with respect to the basic assumption that the remote server is in charge of defining and enforcing access control policies. As such, our SABE scheme addresses the problem of enforcing access control by following up data encryption. The idea is concretely to use different encryption keys for different security levels as proposed, for example, for XML documents. To access such encrypted data, users have to decrypt them by using the appropriate deciphering key. If different users know different keys, with respect to their assigned credentials, they have different access rights.

Mobile Communication – mobile phones, PDAs and various mobile terminals are more and more often used for multimedia communication that require efficient access control mechanisms. Resource consumption is the main limiting factor for the development and deployment of such security mechanisms. This is mainly due to the nature of the smart things, which are resource-impoverished nodes where the implementation of heavy cryptographic primitives is unfeasible. The resource consumption in mobile communications is tied up to the amount of data being processed, stored, and transmitted. As such, reducing the amount of processed and transmitted data can effectively save energy. Our SABE mechanism can be considered as a promoting solution that permits to save computing and storage capabilities of mobile terminals by removing redundant processed information within the network flow. For example, our selective attribute based encryption technique should be a candidate for protecting content in a home multimedia network where some of the receiving devices are expected to be mobile (i.e., resource-constrained) or meant to be very inexpensive. So that, saving computational complexity is very important.

9 CONCLUSION

In this paper, we propose a novel cryptographic mechanism to ensure multi-level access control, based on the use of an attribute based encryption scheme. Our selective attribute based encryption mechanism SABE, enables the enciphering user to encrypt the same data content, based on an ABE aggregate access tree, and the deciphering entity to decrypt the subsets of data blocks with respect to a security level k_l. Indeed, SABE supports a fine grained access control mechanism with low processing costs, which is directly inherited from the expressiveness of ciphertext-policy attribute based encryption for defining access policies. Additionally, our proposal is proven secure against selective, non-adaptive chosen ciphertext attacks in the generic group model. Besides, a quantitative comparison of SABE with the naive computing approach shows the gain of our construction with respect to the processing and communication costs, especially due to the use of an aggregate access structure. Finally, we present the potential of SABE technique to support security and privacy in concrete networking and computing applications.

REFERENCES

Ahn, J. H., Boneh, D., Camenisch, J., Hohenberger, S., Shelat, A., and Waters, B. (2012). Computing on authenticated data. In *Proc. of TCC*, LNCS.

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

Belguith, S., Kaaniche, N., Jemai, A., Laurent, M., and Attia, R. (2016). Pabac: a privacy preserving attribute based framework for fine grained access control in clouds. In *SECRYPT 2016: 13th International Conference on Security and Cryptography*, volume 4, pages 133–146. Scitepress.

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.

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.

Garg, S., Gentry, C., Halevi, S., Sahai, A., and Waters, B. (2013). Attribute-based encryption for circuits from multilinear maps. In *Advances in Cryptology–CRYPTO 2013*, pages 479–499. Springer.

Hassler, H., Posch, R., and Ristić, V. (1993). *Unique Keys Enabling Multithreshold Schemes*. IIG-report-series / Institutes for Information Processing Graz / Institute fur Informationsverarbeitung Graz: IIG-report-series. Institutes for Information Processing Graz.

Hohenberger, S., Sahai, A., and Waters, B. (2013). Full domain hash from (leveled) multilinear maps and identity-based aggregate signatures. In *Advances in Cryptology–CRYPTO 2013*, pages 494–512. Springer.

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.

Kaaniche, N. and Laurent, M. (2017a). Attribute based encryption for multi-level access control policies. In *SECRYPT 2017: 14th International Conference on Security and Cryptography*, volume 6, pages 67–78. Scitepress.

Kaaniche, N. and Laurent, M. (2017b). Data security and privacy preservation in cloud storage environments based on cryptographic mechanisms. *Computer Communications*, 111:120–141.

Khan, F., Li, H., and Zhang, L. (2016). Owner specified excessive access control for attribute based encryption. *IEEE Access*, 4:8967–8976.

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

Shamir, A. How to share a secret. *Commun. ACM*, 22(11).

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.