Key-Policy ABE with Delegation of Rights

Cécile Delerablée[1], Lenaïck Gouriou[1,2,3], and David Pointcheval[2,3]

[1] Leanear, France
[2] DIENS, École normale supérieure, CNRS, PSL University, Paris, France
[3] INRIA, Paris, France

Abstract. This paper revisits Key-Policy Attribute-Based Encryption (KP-ABE), allowing the delegation of keys, traceability of compromised keys, and key anonymity, as additional properties. Whereas delegation of rights has been addressed in the seminal paper by Goyal *et al.* in 2006, introducing KP-ABE, this feature has almost been neglected in all subsequent works in favor of better security levels. However, in multi-device scenarios, this is quite important to allow users to independently authorize their own devices, and thus to delegate their rights. But then, one may require some tracing capabilities and some anonymity properties.
To this aim, we define a new variant of KP-ABE including delegation, with *switchable* attributes, in both the ciphertexts and the keys, and new indistinguishability properties. We then provide a concrete and efficient instantiation with adaptive security under the sole SXDH assumption in the standard model. We eventually show how this can be used for delegatable KP-ABE with black-box traceability.

1 Introduction

Multi-device scenarios have become prevalent in recent years, as it is now quite usual for people to own multiple phones and computers for personal and professional purposes. Users manage multiple applications across different devices, which brings forth new kinds of requirements. One must be able to granularly control what each of his devices can do for numerous applications, with a cost that is minimal for the user and the overall system. In particular, it is expected that one can control what each of its devices can access, for example restricting the rights to read sensitive documents from a professional laptop or phone during travel. Furthermore, if one suspects a device to be compromised, it should be possible to trace and revoke it without impacting the service or the other devices of the user. At the same time, these operations must happen transparently between different devices from the perspective of the user. This means each device should be autonomously configurable with regards to interactions with a central authority or to other devices. Eventually, one may also expect the delegated keys to be unlinkable, for some kind of anonymity for the users.

Usual current authentication means define a unique account for the user, providing the same access-rights to all the devices. This is equivalent to a key-cloning approach, where the user clones his key in every device. In this case, all the devices of the same user are easily linked together, from their keys. This definitely does not offer a satisfactory level of granularity and does not allow to differentiate devices, which prevents tracing and revocation of specific devices.

Key-Policy Attribute-Based Encryption (KP-ABE), in the seminal paper of Goyal *et al.* [GPSW06], offers interesting solutions to these issues. Indeed, in their solution, a policy is embedded inside each user's private key, any user can finely-tune the policy for each of his devices when delegating his keys, for any more restrictive policy. Besides, since keys become different in each device, one could expect to trace and revoke devices independently. However, this is not immediately possible, and we also suggest complementing these features with a certain level of unlinkability between the different keys of a single user in order to better protect the privacy of users.

1.1 Related Work

Attribute-Based Encryption (ABE) has first been proposed in the paper by Goyal *et al.* [GPSW06]. In an ABE system, on the one hand, there is a policy $\mathcal{P}$ and, on the other hand, there are some attributes $(A_i)_i$, and one can decrypt a ciphertext with a key if the policy $\mathcal{P}$ is satisfied on the attributes $(A_i)_i$. They formally defined two approaches: Key-Policy Attribute-Based Encryption (KP-ABE), where the policy is specified in the decryption key and the attributes are associated to the ciphertext; Ciphertext-Policy Attribute-Based Encryption (CP-ABE), where the policy is specified in the ciphertext and the attributes are associated to the decryption key.

In their paper, they proposed a concrete construction of KP-ABE, for any monotonous access structure defined by a policy expressed as an access-tree with threshold internal gates and leaves associated to attributes. Attributes in the ciphertext are among a large universe $\mathcal{U}$ (not polynomially bounded). Given an access-tree $\mathcal{T}$ embedded in a private key, and a set of attributes $\Gamma \subset \mathcal{U}$ associated to a ciphertext, one can decrypt if and only if $\mathcal{T}(\Gamma) = 1$. To evaluate $\mathcal{T}(\Gamma)$, one starts from the leaves of the tree, setting the ones with a matching attribute in Γ to True, and other leaves to False. Then, each node is evaluated recursively from its children, until the root is reached. We say that $\mathcal{T}(\Gamma) = 1$ if the root is satisfied (True). If none of the subsets of Γ can lead to a True for the root, then $\mathcal{T}(\Gamma) = 0$.

Furthermore, they also laid down the bases for delegation of users' private keys. Given an access-tree $\mathcal{T}$ associated to a private key, one can delegate a new key, associated with a more restrictive access-tree $\mathcal{T}'$.

This first paper on KP-ABE allows fine-grained access-control for multiple devices, dealing with delegation of keys for more restrictive policies. However, their choice of parameters for delegation is conflictual with the ones that would be necessary for tracing compromised keys. Indeed, on the one hand, for delegation to work properly, users must be given enough information in the public key to be able to produce valid delegated keys. On the other hand, for the tracing process to be effective in a black-box way, users must not be able to detect it. Information required for delegation may help to detect tracing. From our knowledge, this natural tension between the two features is in all the existing literature. This is the reason why delegation and traceability for KP-ABE has not been dealt yet.

Black-Box Traitor-Tracing. In a black-box traitor-tracing system, a tracing authority can interact with a Pirate Decoder (PD) that non-legitimately decrypts ciphertexts, using one or more decryption keys of legitimate users (the traitors). The keys used by the PD, or the aggregated key, are unknown to the tracing authority when we are dealing with black-box tracing, the most reasonable scenario. The goal of traitor-tracing is to determine which user's private keys are used by the PD, only interacting with the PD in a black-box way, in turn allowing to identify the traitors or the compromised devices. An approach is to embed codewords (also called "fingerprints") with specific properties in the decryption keys. These codewords can then be recovered, under some marking assumptions that address collusion of traitors, after only a few interactions with the PD. Boneh and Shaw [BS95] proposed a tracing technique by embedding codewords in each ciphertext. With this approach, the ciphertext size has to be linear in the length of the codeword, and this length quickly increases with the size of the possible collusion. Boneh and Naor [BN08] improved this approach with a shorter ciphertext: only some bits of the codeword are involved in each ciphertext, but in this case tracing requires additional assumptions on the decryption capabilities of the PD.

Boneh *et al.* [BSW06], followed by [BW06], proposed traceability (and revocation) whatever the size of the collusion, but with ciphertexts of size $\sqrt{N}$, where N is the maximal number of users. Wong *et al.* [LW15,LLLW17] combined this technique into a CP-ABE, with policy encoded in a Linear Secret Sharing Scheme (LSSS). Those techniques nevertheless seem incompatible with delegation properties. Intuitively, their approach assigns each single user to a different cell in a table, and then methodically tests each cell of the table for a traitor, with linear tracing. This is quite exclusive with delegation for the users, as one cannot add more cells in the table.

Lai and Tang [LT18] proposed a framework for traitor-tracing in ABE. Their technique is a generic transformation to make any ABE into a traceable ABE, following above Boneh and Shaw [BS95] methodology. By representing bits in fingerprinting codewords as attributes, they successfully embed the words into any ABE key. However, their construction is a generic one, and the additional layer excludes delegation for the usual ABE scheme. Nevertheless, our approach will be in this vein, but for a very specific construction.

Predicate Encryption/Inner-Product Encryption (IPE) were used by Okamoto and Takashima [OT10,OT12a,OT12b], together with LSSS: the receiver can read the message if a predicate is satisfied on some information in the decryption key and in the ciphertext. Inner-product encryption (where the predicate checks whether the vectors embedded in the key and in the ciphertext are orthogonal) is the major tool. Their technique of Dual Pairing Vector Space (DPVS) provided two major advantages in KP-ABE applications: whereas previous constructions were only secure against selective attacks (the attributes in the challenge ciphertext were known before the publication of the keys), this technique allowed full security (where the attributes in the challenge ciphertext are chosen at the challenge-query time). In addition, it allows the notion of attribute-hiding (from [KSW08]) where no information leaks about the attributes associated to the ciphertext, except for the fact that they are accepted or not by the policies in the keys. It gets closer to our goals, as tracing might become undetectable. However, these approaches do not seem compatible with delegation, as the security proofs require all the key generation material to remain secret to the key issuer, excluding delegation capabilities to users.

As follow-up works, Chen *et al.* [CGW15,CGW18] designed multiple systems for IPE, with adaptive security, and explored full attribute-hiding with weaker assumptions and shorter ciphertexts and secret keys than in the previous work of Okamoto-Takashima. However, it does not fit our expectations on delegation, for the same reasons.

Recent works by Attrapadung have been proposed for ABE by introducing Pair Encoding Systems, which allow for all possible predicates and large universes [AT20], but they deal neither with delegation nor with any kind of attribute-hiding which could be compatible for our use.

1.2 Contributions

Since the approach of [OT12a] is close to our goal, we extend its original construction to make it compatible with delegation. As an illustration, we start by a simple variant that handles delegation with adaptive security, in the standard model and under the SXDH assumption, in the appendix B. But since it does not allow undetectable traceability, we enhance the definition of KP-ABE to provide indistinguishability properties for traceability.

To this aim, of delegatable KP-ABE with traceability, which is our main contribution, we first detail one of the main limitation we have to overcome: with the original approach of [GPSW06], attributes associated to the ciphertext are explicitly stated as elements in the ciphertext. Removing some attributes can thus allow revocation, but this is public, and thus incompatible with any tracing procedure. To prevent that, we introduce a new primitive: Switchable-Attribute Key-Policy Attribute-Based Encryption (SA-KP-ABE), where one can invalidate some attributes in the ciphertext, without removing them. This will bring new properties to the attributes in ciphertexts (for the traceability) but also symmetrically to the leaves in keys (for anonymity).

In a SA-KP-ABE scheme, attributes in a ciphertext and leaves in a key can be switched in two different states: valid or invalid for attributes in a ciphertext, with the disjoint sets $\Gamma = \Gamma_v \cup \Gamma_i$, and passive or active for leaves in a key, with the disjoint sets $\mathcal{L} = \mathcal{L}_p \cup \mathcal{L}_a$. With this additional property, a set of valid/invalid attributes $\Gamma = \Gamma_v \cup \Gamma_i$ is accepted by an access-tree with active/passive leaves $\mathcal{L} = \mathcal{L}_a \cup \mathcal{L}_p$, if the tree is accepting when all the leaves in $\mathcal{L}$ associated to an attribute in Γ are set to True, except if the leaf is active (in $\mathcal{L}_a$) and the associated attribute invalid (in Γ_i). Concretely, active leaves are chosen during the Key

Generation procedure by the authority, and then the keys are given to the users, with these switchable policies. During the Encryption procedure, one might specify some attributes to be invalid, which virtually and secretly switches some active leaves to False. Note that users with corresponding passive leaves are not impacted.

A second contribution is a concrete and efficient instantiation of such an SA-KP-ABE, with security proofs in the standard model and under the SXDH assumption. We eventually explain how one can deal with delegatable and traceable KP-ABE from such a primitive.

1.3 Discussion

Combining active leaves and invalid ciphertexts, one may revoke users following a specific strategy, as in [BS95] or [BN08] (more details will be provided in Section 6). However, in order for tracing to work, one needs the adversary not to be able do distinguish valid from invalid attributes in the ciphertexts: to answer this, we will define the notion of Attribute-Indistinguishability. By symmetry, we also introduce another indistinguishability notion for keys, called Key-Indistinguishability, where active and passive leaves are indistinguishable. In case of delegation, the latter indistinguishability will help us to achieve some kind of anonymity, by making delegated keys issued from the same key unlinkable.

We note that this setting bears common characteristics with some recent KP-ABE approaches. However there are key major specificities to consider. First, Waters [Wat09] introduced the technique called Dual System Encryption (DSE), to improve the security level of KP-ABE, from selective security in [GPSW06] to adaptive security: the set Γ of attributes in the challenge ciphertext can be chosen by the adversary at the challenge-query time. In DSE, keys and ciphertexts can be set *semi-functional*, which is in the same vein as our active leaves in keys and invalid attributes in ciphertexts. However, DSE solely uses semi-functional keys and ciphertexts during the simulation, in the security proof, while our construction exploits them in the real-life construction. The security proof thus needs another layer of tricks. Second, the attribute-hiding notions are strong properties that have been well studied in different IPE works. However, one does not need to achieve such a strong result for tracing: Our proposed Attribute-Indistinguishability is well-suited to KP-ABE, and properly tailored to enable tracing.

To conclude, in the standard model, and under the sole SXDH assumption, we have built an adaptively secure KP-ABE that is well-suited for fine-grained delegation, and have developed a new primitive, SA-KP-ABE, with an adaptively-secure construction, allowing traceability and providing some privacy guarantees.

As a final remark, while our techniques have been developed with a vocabulary dedicated to access-tree policies, they seem fully compatible with a policy enforced through more general LSSS. We found access-tree useful to develop intuition for delegation and the construction of the SA-KP-ABE. However, we are confident that techniques converting access-trees to LSSS and reciprocally, as in [CPP17], can apply to our construction: the embedding of the policy is done through Inner-Product, which is well-suited for LSSS just as much as for access-trees, and our developments essentially rely on a specific embedding perspective.

1.4 Organization

In the next section, we detail the tools that we will need in our constructions and proofs, such as the Dual Pairing Vector Spaces, along with useful illustrations of its use for our proofs. Section 3 explains KP-ABE and access-tree structures. Sections 4 and 5 develop the main contributions, with the definition of SA-KP-ABE and the first construction, with the security analysis. In Section 6, we explain how our new primitive answers our initial goal of delegatable KP-ABE with traceability.

Because of lack of space, but for the sake of clarity, several details and additional information are deferred to the appendix.

2 Preliminaries

2.1 Computational Assumptions

We will make use of a pairing-friendly setting $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, e, G_1, G_2, q)$, with a bilinear map e from $\mathbb{G}_1 \times \mathbb{G}_2$ into $\mathbb{G}_t$, and G_1 (respectively G_2) is a generator of $\mathbb{G}_1$ (respectively $\mathbb{G}_2$). We will use additive notations for $\mathbb{G}_1$ and $\mathbb{G}_2$, and multiplicative notations in $\mathbb{G}_t$.

As usual, we will admit the Decisional Diffie-Hellman (DDH) Assumption in $\mathbb{G}_1$ or $\mathbb{G}_2$. More generally, in any group $\mathbb{G}$, this assumption is defined as follows:

Definition 1 (Decisional Diffie-Hellman Assumption). *The DDH assumption in $\mathbb{G}$, of prime order q with generator G, states that no algorithm can efficiently distinguish the two distributions*

$$\mathcal{D}_0 = \{(a \cdot G, b \cdot G, ab \cdot G), a, b \xleftarrow{\$} \mathbb{Z}_q\} \qquad \mathcal{D}_1 = \{(a \cdot G, b \cdot G, c \cdot G), a, b, c \xleftarrow{\$} \mathbb{Z}_q\}$$

And we will denote by $\mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}}(T)$ the best advantage an algorithm can get in distinguishing the two distributions within time bounded by T. For the sake of clarity, in the sequence of games, we will sometimes use the following DSDH assumption:

Definition 2 (Decisional Separation Diffie-Hellman Assumption). *The DSDH assumption in $\mathbb{G}$, of prime order q with generator G, between two constant values x, y, states that no algorithm can efficiently distinguish the two distributions, where $a, b \xleftarrow{\$} \mathbb{Z}_q$,*

$$\mathcal{D}_x = \{(a \cdot G, b \cdot G, (ab + x) \cdot G)\} \qquad \mathcal{D}_y = \{(a \cdot G, b \cdot G, (ab + y) \cdot G)\}$$

As $c + x$ and $c + y$ are perfectly indistinguishable for a random c, then the best advantage an algorithm can get in distinguishing the two distributions within time T is upper-bounded by $2 \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}}(T)$. Eventually, we will make the following more general assumption:

Definition 3 (Symmetric eXternal Diffie-Hellman Assumption). *The SXDH assumption in $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, e, G_1, G_2, q)$ makes the DDH assumptions in both $\mathbb{G}_1$ and $\mathbb{G}_2$.*

Then, we define $\mathsf{Adv}^{\mathsf{sxdh}}(T) = \max\{\mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(T), \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(T)\}$.

2.2 Dual Pairing Vector Spaces

More details are provided in the appendix A, but the main points are reviewed in this section to help following the constructions and the security proofs. Dual Pairing Vector Spaces (DPVS) have been proposed for efficient schemes with adaptive security [OT08,LOS+10,OT10,OT12b], in the same vein as Dual System Encryption (DSE) [Wat09], in prime-order groups under the DLIN assumption. In [LW10], DSE was using pairings on composite-order elliptic curves. Then, prime-order groups have been used with the SXDH assumption [CLL+13]. In all theses situations, one exploited indistinguishability of sub-groups or sub-spaces. In this paper, we use the SXDH assumption in a pairing-friendly setting $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, e, G_1, G_2, q)$ of primer order q, with the additional law between an element $X \in \mathbb{G}_1^n$ and $Y \in \mathbb{G}_2^n$: $X \times Y \stackrel{\mathsf{def}}{=} \prod_i e(X_i, Y_i)$, where $\mathbb{G}_t$ is usually denoted multiplicatively, while we will use additive notation for $\mathbb{G}_1$ and $\mathbb{G}_2$. Hence, if $X = (X_1, \ldots, X_n) = \vec{x} \cdot G_1 \in \mathbb{G}_1^n$ and $Y = (Y_1, \ldots, Y_n) = \vec{y} \cdot G_2 \in \mathbb{G}_2^n$:

$$(\vec{x} \cdot G_1) \times (\vec{y} \cdot G_2) = X \times Y = \prod\nolimits_i e(X_i, Y_i) = g_t^{\langle \vec{x}, \vec{y} \rangle}$$

where $g_t = e(G_1, G_2)$ and $\langle \vec{x}, \vec{y} \rangle$ is the inner product between vectors $\vec{x}$ and $\vec{y}$.

From any basis $\mathcal{B} = (\vec{b}_i)_i$ of $\mathbb{Z}_q^n$, we can define the basis $\mathbb{B} = (\mathbf{b}_i)_i$ of $\mathbb{G}_1^n$, where $\mathbf{b}_i = \vec{b}_i \cdot G_1$. Such a basis $\mathcal{B}$ is equivalent to a random invertible matrix $B \xleftarrow{\$} \mathrm{GL}_n(\mathbb{Z}_q)$, the matrix with

$\vec{b}_i$ as its i-th row. If we additionally use $\mathbb{B}^* = (\mathbf{b}_i^*)_i$, the basis of $\mathbb{G}_2^n$ associated to the matrix $B' = (B^{-1})^\top$, as $B \cdot B'^\top = I_n$,

$$\mathbf{b}_i \times \mathbf{b}_j^* = (\vec{b}_i \cdot G_1) \times (\vec{b}'_j \cdot G_2) = g_t^{\langle \vec{b}_i, \vec{b}'_j \rangle} = g_t^{\delta_{i,j}},$$

where $\delta_{i,j} = 1$ if $i = j$ and $\delta_{i,j} = 0$ otherwise, for $i, j \in \{1, \ldots, n\}$: $\mathbb{B}$ and $\mathbb{B}^*$ are called *Dual Orthogonal Bases.* A pairing-friendly setting $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, e, G_1, G_2, q)$, with such dual orthogonal bases $\mathbb{B}$ and $\mathbb{B}^*$ of size n, is called a *Dual Pairing Vector Space.*

2.3 Change of Basis

Let us consider the basis $\mathbb{U} = (\mathbf{u}_i)_i$ of $\mathbb{G}^n$ associated to a random matrix $U \in \mathrm{GL}_n(\mathbb{Z}_q)$, and the basis $\mathbb{B} = (\mathbf{b}_i)_i$ of $\mathbb{G}^n$ associated to the product matrix BU, for any $B \in \mathrm{GL}_n(\mathbb{Z}_q)$, a vector $\vec{x}$ in $\mathbb{B}$, denoted $(\vec{x})_\mathbb{B}$ means

$$\begin{aligned}(\vec{x})_\mathbb{B} &= \sum\nolimits_i x_i \cdot \mathbf{b}_i = \sum\nolimits_i x_i \cdot \vec{b}_i \cdot G = \vec{x} \cdot BU \cdot G = (\vec{x} \cdot B) \cdot U \cdot G = \vec{y} \cdot U \cdot G \\ &= \sum\nolimits_i y_i \cdot \vec{u}_i \cdot G = \sum\nolimits_i y_i \cdot \mathbf{u}_i = (\vec{y})_\mathbb{U} \text{ where } \vec{y} = \vec{x} \cdot B.\end{aligned}$$

Hence, $(\vec{x})_\mathbb{B} = (\vec{x} \cdot B)_\mathbb{U}$ and $(\vec{x} \cdot B^{-1})_\mathbb{B} = (\vec{x})_\mathbb{U}$ where we denote $\mathbb{B} \stackrel{\mathsf{def}}{=} B \cdot \mathbb{U}$. For any invertible matrix B, if $\mathbb{U}$ is a random basis, then $\mathbb{B} = B \cdot \mathbb{U}$ is also a random basis. Then, with $B^{-1} = (\vec{b'}_1^\top, \ldots, \vec{b'}_n^\top)$, $\vec{x} = \vec{y} \cdot (\vec{b'}_1^\top, \ldots, \vec{b'}_n^\top)$:

$$\mathbb{B} = B \cdot \mathbb{U}, B' = \begin{pmatrix} \vec{b'}_1 \\ \vdots \\ \vec{b'}_n \end{pmatrix}, \text{ and } (\vec{x})_\mathbb{B} = (\vec{y})_\mathbb{U} \Longrightarrow \vec{x} = (\langle \vec{y}, \vec{b'}_1 \rangle, \ldots, \langle \vec{y}, \vec{b'}_n \rangle).$$

Let us consider the random dual orthogonal bases $\mathbb{U} = (\mathbf{u}_i)_i$ and $\mathbb{U}^* = (\mathbf{u}_i^*)_i$ of $\mathbb{G}_1^n$ and $\mathbb{G}_2^n$ respectively associated to a matrix U (which means that $\mathbb{U}$ is associated to the matrix U and $\mathbb{U}^*$ is associated to the matrix $U' = (U^{-1})^\top$): the bases $\mathbb{B} = B \cdot \mathbb{U}$ and $\mathbb{B}^* = B' \cdot \mathbb{U}^*$, where $B' = (B^{-1})^\top$, are also dual orthogonal bases:

$$\mathbf{b}_i \times \mathbf{b}_j^* = g_t^{\vec{b}_i \cdot \vec{b'}_j^\top} = g_t^{\vec{u}_i \cdot B \cdot (B^{-1})^\top \cdot \vec{u'}_j^\top} = g_t^{\vec{u}_i \cdot \vec{u'}_j^\top} = g_t^{\delta_{i,j}}.$$

All the security proofs will exploit changes of bases, of essentially two kinds:

Formal Change of Basis, where we start from two dual orthogonal bases $\mathbb{U}$ and $\mathbb{U}^*$ of dimension 2, and set

$$B = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \qquad B' = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix} \qquad \mathbb{B} = B \cdot \mathbb{U} \qquad \mathbb{B}^* = B' \cdot \mathbb{U}^*$$

then,

$$\begin{aligned}(x_1, x_2)_\mathbb{U} &= (x_1, x_2 - x_1)_\mathbb{B} & \qquad (y_1, y_2)_{\mathbb{U}^*} &= (y_1 + y_2, y_2)_{\mathbb{B}^*} \\ (0, x_2)_\mathbb{U} &= (0, x_2)_\mathbb{B} & (0, y_2)_{\mathbb{U}^*} &= (y_2, y_2)_{\mathbb{B}^*}\end{aligned}$$

In practice, this change of basis makes $\mathbf{b}_1 = \mathbf{u}_1 + \mathbf{u}_2$, $\mathbf{b}_2 = \mathbf{u}_2$, $\mathbf{b}_1^* = \mathbf{u}_1^*$, $\mathbf{b}_2^* = -\mathbf{u}_1^* + \mathbf{u}_2^*$. If $\mathbf{b}_1$ and $\mathbf{b}_2^*$ are kept private, the adversary cannot know whether we are using $(\mathbb{U}, \mathbb{U}^*)$ or $(\mathbb{B}, \mathbb{B}^*)$. This exact change of basis will be used to duplicate some component, as shown in the second example where y_2 in the second component is duplicated in the first component in $\mathbb{B}^*$, without impacting vectors in $\mathbb{U}$, if all the first components are 0.

SubSpace-Ind: with $\mathbf{b}_2^*$ hidden

$$\mathbf{c} = (\ x_1 \quad x_2 \quad x_3\ )_{\mathbb{B}} \approx (\ x_1 \quad x'_2 \quad x_3\ )_{\mathbb{B}}$$

$$\mathbf{k} = (\ y_1 \quad y_2 \quad y_3\ )_{\mathbb{B}^*} = (\ y_1 \quad y_2 \quad y_3\ )_{\mathbb{B}^*}$$

Swap-Ind: with $\mathbf{b}_1^*, \mathbf{b}_2^*$ hidden

$$\mathbf{c} = (\ x_1 \quad 0 \quad x_3\ )_{\mathbb{B}} \approx (\ 0 \quad x_1 \quad x_3\ )_{\mathbb{B}}$$

$$\mathbf{k} = (\ y_1 \quad y_1 \quad y_3\ )_{\mathbb{B}^*} = (\ y_1 \quad y_1 \quad y_3\ )_{\mathbb{B}^*}$$

Index-Ind: with $\mathbf{b}_3^*$ hidden, if $p \neq t$

$$\mathbf{c} = (\ \sigma \cdot (1, p) \quad x_3\ )_{\mathbb{B}} \approx (\ \sigma \cdot (1, p) \quad x'_3\ )_{\mathbb{B}}$$

$$\mathbf{k} = (\ \pi \cdot (t, -1) \quad y_3\ )_{\mathbb{B}^*} = (\ \pi \cdot (t, -1) \quad y_3\ )_{\mathbb{B}^*}$$

Colored cells x are random values, while gray cells x correspond to chosen modified values.

Fig. 1: Computationally indistinguishable Changes of Basis

Computational Change of Basis, where we start from two dual orthogonal bases $\mathbb{U}$ and $\mathbb{U}^*$ of dimension 2. From a triple $(a \cdot G_1, b \cdot G_1, c \cdot G_1)$, where $c = ab + \tau \bmod q$ with either $\tau = 0$ or $\tau \xleftarrow{\$} \mathbb{Z}_q^*$, one can set

$$B = \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix} \qquad B' = \begin{pmatrix} 1 & 0 \\ -a & 1 \end{pmatrix} \qquad \mathbb{B} = B \cdot \mathbb{U} \qquad \mathbb{B}^* = B' \cdot \mathbb{U}^*$$

then,
$$(b, c)_{\mathbb{U}} + (x_1, x_2)_{\mathbb{B}} = (b, c - ab)_{\mathbb{B}} + (x_1, x_2)_{\mathbb{B}} = (x_1 + b, x_2 + \tau)_{\mathbb{B}}$$
$$(y_1, y_2)_{\mathbb{U}^*} = (y_1 + ay_2, y_2)_{\mathbb{B}^*}$$

where τ can be either 0 or random. We should however note that in this case, $\mathbf{b}_2^*$ cannot be computed, as $a \cdot G_2$ is not known. This will not be a problem as this element is not provided as a public element, but just kept by the simulator.

Partial Change of Basis: in the constructions, bases will be of higher dimension, but we will often only change a few basis vectors. We will then specify the vectors as indices to the change of basis matrix, as explained in the appendix A.

We briefly recap, see Figure 1, all the indistinguishable modifications (under the DSDH assumption in $\mathbb{G}_1$, but it can also be applied in $\mathbb{G}_2$), on random dual orthogonal bases $\mathbb{B}$ and $\mathbb{B}^*$. They can all be proven using a change of basis similar to the above *Computational Change of Basis*:

SubSpace-Ind Property, on $(\mathbb{B}, \mathbb{B}^*)_{1,2}$: from the view of $\mathbb{B}$ and $\mathbb{B}^* \backslash \{\mathbf{b}_2^*\}$, and any vector $(y_1, y_2, \ldots, y_n)_{\mathbb{B}^*}$, for chosen $y_2, \ldots, y_n \in \mathbb{Z}_q$, but unknown random $y_1 \xleftarrow{\$} \mathbb{Z}_q$, one cannot distinguish the vectors $(x_1, x'_2, x_3, \ldots, x_n)_{\mathbb{B}}$ and $(x_1, x_2, x_3, \ldots, x_n)_{\mathbb{B}}$, for chosen $x'_2, x_2, \ldots, x_n \in \mathbb{Z}_q$, but unknown random $x_1 \xleftarrow{\$} \mathbb{Z}_q$ (see Theorem 19).

Swap-Ind Property, on $(\mathbb{B}, \mathbb{B}^*)_{1,2,3}$: from the view of $\mathbb{B}$ and $\mathbb{B}^* \backslash \{\mathbf{b}_1^*, \mathbf{b}_2^*\}$, and any vector $(y_1, y_1, y_3, \ldots, y_n)_{\mathbb{B}^*}$, for chosen $y_1, y_3, \ldots, y_n \in \mathbb{Z}_q$, one cannot distinguish the vectors $(x_1, 0, x_3, x_4, \ldots, x_n)_{\mathbb{B}}$ and $(0, x_1, x_3, x_4, \ldots, x_n)_{\mathbb{B}}$, for chosen $x_1, x_4, \ldots, x_n \in \mathbb{Z}_q$, but unknown random $x_3 \xleftarrow{\$} \mathbb{Z}_q$ (see Theorem 21).

Index-Ind Property, on $(\mathbb{B}, \mathbb{B}^*)_{1,2,3}$: from the view of $\mathbb{B}$ and $\mathbb{B}^* \backslash \{\mathbf{b}_3^*\}$, and any vector $(\pi \cdot (t, -1), y_3, \ldots, y_n)_{\mathbb{B}^*}$, for chosen $y_3, \ldots, y_n \in \mathbb{Z}_q$, but unknown random $\pi \xleftarrow{\$} \mathbb{Z}_q$, and for any chosen $t \neq p \in \mathbb{Z}_q$, one cannot distinguish the vectors $(\sigma \cdot (1, p), x_3, x_4, \ldots, x_n)_{\mathbb{B}}$ and $(\sigma \cdot (1, p), x'_3, x_4, \ldots, x_n)_{\mathbb{B}}$, for chosen $x'_3, x_3, x_4, \ldots, x_n \in \mathbb{Z}_q$, but unknown random $\sigma \xleftarrow{\$} \mathbb{Z}_q$ (see Theorem 23).

3 Key-Policy Attribute-Based Encryption

We recall a few definitions for Key-Policy Attribute-Based Encryption (KP-ABE), following [GPSW06], with explicit delegation.

3.1 Policy Definition

ABE limits decryption according to policies on attributes. The policy is either specified in the decryption key (KP-ABE) or in the ciphertext (CP-ABE), on attributes that are then specified in the ciphertext of the decryption key, respectively. In the following, we focus on the Key-Policy case.

Access Tree. As in this seminal paper [GPSW06], we will consider an access-tree $\mathcal{T}$ to model the policy on attributes in an unbounded universe $\mathcal{U}$, but with only AND and OR gates instead of more general threshold gates: an AND-gate being an n-out-of-n gate, whereas an OR-gate is a 1-out-of-n gate. This is also a particular case of the more general LSSS technique. Nevertheless, such an access-tree with only AND and OR gates is as expressive as with any threshold gates or LSSS. For any monotonic policy, we define our access-tree in the following way: $\mathcal{T}$ is a rooted labeled tree from the root ρ, with internal nodes associated to AND and OR gates and leaves associated to attributes. More precisely, for each leaf $\lambda \in \mathcal{L}$, $A(\lambda) \in \mathcal{U}$ is an attribute, and any internal node $\nu \in \mathcal{N}$ is labeled with a gate $G(\nu) \in \{\mathsf{AND}, \mathsf{OR}\}$ as an AND or an OR gate to be satisfied among the children in $\mathsf{children}(\nu)$. We will implicitly consider that any access-tree $\mathcal{T}$ is associated to the attribute-labeling A of the leaves and the gate-labeling G of the nodes. For any leaf $\lambda \in \mathcal{L}$ of $\mathcal{T}$ or internal node $\nu \in \mathcal{N}\backslash\{\rho\}$, the function parent links to the parent node: $\nu \in \mathsf{children}(\mathsf{parent}(\nu))$ and $\lambda \in \mathsf{children}(\mathsf{parent}(\lambda))$.

On a given list $\Gamma \subseteq \mathcal{U}$ of attributes, each leaf $\lambda \in \mathcal{L}$ is either satisfied (considered or set to True), if $A(\lambda) \in \Gamma$, or not (ignored or set to False) otherwise: we will denote $\mathcal{L}_\Gamma$ the restriction of $\mathcal{L}$ to the satisfied leaves in the tree $\mathcal{T}$ (corresponding to an attribute in Γ). Then, for each internal node ν, one checks whether all children (AND-gate) or at least one of the children (OR-gate) are satisfied, from the attributes associated to the leaves, and then ν is itself satisfied or not. More precisely, for each internal node ν, we denote $\mathcal{T}_\nu$ the subtree rooted at the node ν. Hence, $\mathcal{T} = \mathcal{T}_\rho$. A leaf $\lambda \in \mathcal{L}$ is satisfied if $\lambda \in \mathcal{L}_\Gamma$ then, recursively, $\mathcal{T}_\nu$ is satisfied if the AND/OR-gate associated to ν via $G(\nu)$ is satisfied with respect to status of the children in $\mathsf{children}(\nu)$: we denote $\mathcal{T}_\nu(\Gamma) = 1$ when the subtree is satisfied, and 0 otherwise:

$$\begin{array}{lll}
\mathcal{T}_\lambda(\Gamma) = 1 & \text{iff } \lambda \in \mathcal{L}_\Gamma & \text{for any leaf } \lambda \in \mathcal{L} \\
\mathcal{T}_\nu(\Gamma) = 1 & \text{iff } \forall \kappa \in \mathsf{children}(\nu), \mathcal{T}_\kappa(\Gamma) = 1 & \text{when } G(\nu) = \mathsf{AND} \\
\mathcal{T}_\nu(\Gamma) = 1 & \text{iff } \exists \kappa \in \mathsf{children}(\nu), \mathcal{T}_\kappa(\Gamma) = 1 & \text{when } G(\nu) = \mathsf{OR}
\end{array}$$

Illustration of Access-Trees with NNL. We illustrate how one can convert the complete subtree approach of NNL, into a policy for a KP-ABE, expressed as an access-tree (see Figure 2) to get a broadcast encryption with revocation : one first considers all the NNL subtrees as attributes (all internal nodes ν_i and all leaves j), a user's policy should accept any attribute corresponding to a subtree he belongs to. Hence, the policy is an OR-gate with all the acceptable attributes as leaves. With the complete subtree approach, it has been shown that the ciphertext contains at most $r \log N$ attributes, where N is the total number of users and r the number of revoked users.

Evaluation Pruned Tree. In the above definition, we considered an access-tree $\mathcal{T}$ on leaves $\mathcal{L}$ and a set Γ of attributes, with the satisfiability $\mathcal{T}(\Gamma) = 1$ where the predicate defined by $\mathcal{T}$ is true when all the leaves $\lambda \in \mathcal{L}_\Gamma$ are set to True. A Γ-evaluation tree $\mathcal{T}' \subset \mathcal{T}$ is a pruned version of $\mathcal{T}$, where one children only is kept to OR-gate nodes, down to the leaves, so that $\mathcal{T}'(\Gamma) = 1$. Basically, we keep a skeleton with only necessary True leaves to evaluate the internal nodes up to the root. We will denote $\mathsf{EPT}(\mathcal{T}, \Gamma)$ the set of all the evaluation pruned trees of $\mathcal{T}$ with respect to Γ. This set $\mathsf{EPT}(\mathcal{T}, \Gamma)$ is non-empty if and only if $\mathcal{T}(\Gamma) = 1$.

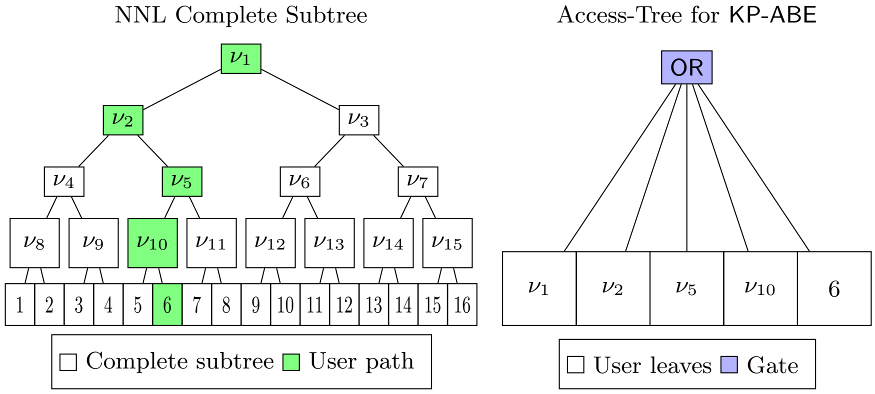

Fig. 2: NNL complete subtree method (for $N = 16$), which defines the universe of attributes $\mathcal{U} = \{\nu_1, \ldots, \nu_{15}, 1, \ldots, 16\}$, with the path to user 6 (left-side), from which we derive the access-tree of user 6 for $\mathsf{KP\text{-}ABE}$ on leaves associated to the attributes $\{\nu_1, \nu_2, \nu_5, \nu_{10}, 6\}$ (right-side)

Figure 3 gives an illustration of such an access-tree for a policy: when the colored leaves $\{\lambda_1, \lambda_3, \lambda_5, \lambda_8, \lambda_9, \lambda_{10}\}$ are True, the tree is satisfied. There are two possible evaluation pruned trees: down to the leaves $\{\lambda_1, \lambda_3, \lambda_5, \lambda_8\}$ or $\{\lambda_1, \lambda_3, \lambda_5, \lambda_9, \lambda_{10}\}$. As already noted, any monotonic access structure can be encoded into such an access-tree.

3.2 Definitions

We now recall the definition of $\mathsf{KP\text{-}ABE}$ from [GPSW06], with access-trees to define policies in the keys:

$\mathsf{Setup}(1^\kappa)$**.** From the security parameter κ, the algorithm defines all the global parameters PK and the master secret key MK;

$\mathsf{KeyGen}(\mathsf{MK}, \mathcal{T})$**.** For a master secret key MK and an access-tree $\mathcal{T}$, the algorithm outputs a private key $\mathsf{dk}_\mathcal{T}$;

$\mathsf{Encaps}(\mathsf{PK}, \Gamma)$**.** For a list Γ of attributes and global parameters PK, the algorithm generates the ciphertext C and an encapsulated key K;

$\mathsf{Decaps}(\mathsf{dk}_\mathcal{T}, C)$**.** Given the private key $\mathsf{dk}_\mathcal{T}$ and the ciphertext C, the algorithm outputs the encapsulated key K.

For correctness, the Decaps algorithm should output the encapsulated key K if and only if C has been generated for a set Γ that satisfies the policy $\mathcal{T}$ of the decryption key $\mathsf{dk}_\mathcal{T}$: $\mathcal{T}(\Gamma) = 1$.

Delegation. A major feature in [GPSW06] is delegation of decryption keys: a user with a decryption key dk corresponding to an access-tree $\mathcal{T}$ can compute a new decryption key corresponding to any more restrictive access-tree, or a less accessible tree $\mathcal{T}'$, than $\mathcal{T}$ with the following partial order: $\mathcal{T}' \leq \mathcal{T}$, if and only if for any subset Γ of attributes, $\mathcal{T}'(\Gamma) = 1 \Longrightarrow \mathcal{T}(\Gamma) = 1$. More concretely, in our case of access-trees, a more restrictive access-tree is, for each node ν,

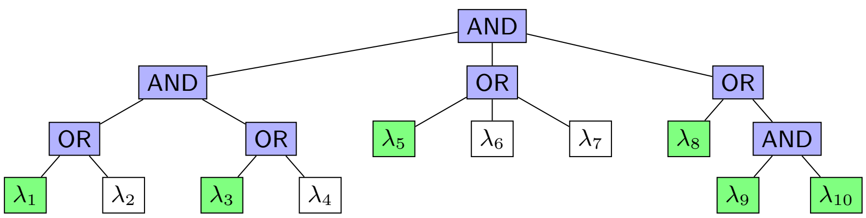

Fig. 3: Example of an access-tree with two different evaluation pruned trees for the leaves colored in green: $\{\lambda_1, \lambda_3, \lambda_5, \lambda_8\}$ or $\{\lambda_1, \lambda_3, \lambda_5, \lambda_9, \lambda_{10}\}$

1. if $G(\nu) = \mathsf{AND}$, one or more children are added (*i.e.*, more constraints);
2. if $G(\nu) = \mathsf{OR}$, one or more children are removed (*i.e.*, less flexibility);
3. the node ν is moved one level below as a child of an AND-gate at node ν', with additional sub-trees as children to this AND-gate (*i.e.*, more constraints).

We illustrate the last rule, with a simple example in Figure 4. There is thus the additional algorithm:

$\mathsf{Delegate}(\mathsf{dk}_{\mathcal{T}}, \mathcal{T}')$**.** Given a key $\mathsf{dk}_{\mathcal{T}}$ and a more restrictive $\mathcal{T}' \leq \mathcal{T}$, the algorithm outputs a decryption key $\mathsf{dk}_{\mathcal{T}'}$.

Security Notions. Whereas we could recall the classical indistinguishability, with only KeyGen-queries, we extend it to handle delegation queries: if one can ask several more restrictive delegations from an access-tree $\mathcal{T}$, one should not be able to distinguish an encapsulated key in a ciphertext under a non-trivial list of attributes, according to the obtained delegated keys only:

Definition 4 (Delegation-Indistinguishability). *Del-IND security for KP-ABE is defined by the following game between the adversary and a challenger:*

Initialize: *The challenger runs the* Setup *algorithm of KP-ABE and gives the public parameters* PK *to the adversary;*

Oracles: *The following oracles can be called in any order and any number of times, except for* RoREncaps *which can be called only once.*

- $\mathsf{OKeyGen}(\mathcal{T})$**:** *to model* KeyGen*-queries for any access-tree $\mathcal{T}$ of its choice, without getting back the decryption key, but for future delegation;*
- $\mathsf{ODelegate}(\mathcal{T}, \mathcal{T}')$**:** *to model* Delegate*-queries for any more restrictive access-tree $\mathcal{T}' \leq \mathcal{T}$ of its choice, for an already generated decryption key under $\mathcal{T}$, and gets back the decryption key $\mathsf{dk}_{\mathcal{T}'}$;*
- $\mathsf{RoREncaps}(\Gamma)$**:** *the challenge real-or-random encapsulation query on a set of attributes Γ is asked once only. The challenger asks for an encapsulation query on Γ and receives (K_0, C). It also generates a random key K_1. It eventually flips a random coin b, and outputs (K_b, C) to the adversary;*

$\mathbf{Finalize}(b')$**:** *The adversary outputs a guess b' for b. If for some access-tree $\mathcal{T}'$ asked to the* ODelegate*-oracle, $\mathcal{T}'(\Gamma) = 1$, on the challenge set Γ, $\beta \stackrel{\$}{\leftarrow} \{0, 1\}$, otherwise one sets $\beta = b'$. One outputs β.*

The advantage of an adversary $\mathcal{A}$ in this game is defined as

$$\mathsf{Adv}^{\mathsf{del\text{-}ind}}(\mathcal{A}) = \Pr[\beta = 1 | b = 1] - \Pr[\beta = 1 | b = 0].$$

One could of course consider **Chosen-Ciphertext security**, where the adversary could have access to some decryption oracles, without the decryption key itself. On the more limited side, one can consider **Selective-Set security**, where the adversary declares Γ at the initialization step, as in [GPSW06]. This Delegation-Indistinguishability is definitely stronger than basic Indistinguishability as the adversary can ask for a $\mathsf{OKeyGen}(\mathcal{T})$-query and thereafter the $\mathsf{ODelegate}(\mathcal{T}, \mathcal{T})$-query to immediately get the decryption key for $\mathcal{T}$.

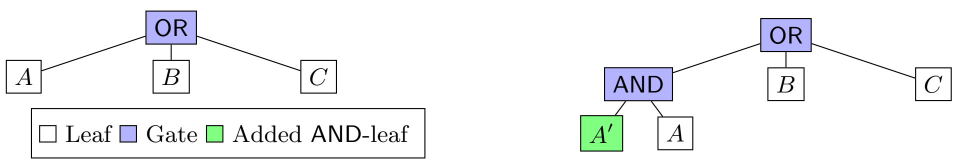

Fig. 4: Access-tree (left-side) and delegated-tree (right-side) where the leaf associated with attribute A is changed into an AND-gate with a new child leaf associated with attribute A'

3.3 More Properties for Access-Trees

For our construction, as in [GPSW06], for the policy, we have to define the notion of labeled access-trees as a secret sharing.

Labeled Access-Trees. We will label such trees with integers so that some labels on the leaves will be enough to recover the labels above, up to the root, as illustrated on Figure 5.

Definition 5 (Random y-Labeling). *Let us define a random y-labeling Λ_y of an access-tree $\mathcal{T}$, for any $y \in \mathbb{Z}_p$: the probabilistic algorithm $\Lambda_y(\mathcal{T})$ sets $a_\rho \leftarrow y$ for the root, and then in a top-down manner, for each internal node ν, starting from the root,*

- *if $G(\nu) = \mathsf{AND}$, with n children, a random n-out-of-n sharing of a_ν is associated to each children* i.e., *random values are associated to a_κ for all $\kappa \in \mathsf{children}(\nu)$, such that the sum is equal to a_ν in $\mathbb{Z}_p$;*
- *if $G(\nu) = \mathsf{OR}$, with n children, each children is associated to the value a_ν.*

$$\begin{aligned} G(\nu) = \mathsf{AND}: &\quad \forall\kappa \in \mathsf{children}(\nu), a_\kappa \stackrel{\$}{\leftarrow} \mathbb{Z}_p, \text{ such that } a_\nu = \sum_{\kappa\in\mathsf{children}(\nu)} a_\kappa \bmod p \\ G(\nu) = \mathsf{OR}: &\quad \forall\kappa \in \mathsf{children}(\nu), a_\kappa = a_\nu \end{aligned}$$

Algorithm $\Lambda_y(\mathcal{T})$ outputs $\Lambda_y = (a_\lambda)_{\lambda\in\mathcal{L}}$, for all the leaves $\lambda \in \mathcal{L}$ of the tree $\mathcal{T}$. Because of the linearity, one can remark that from any y-labeling $(a_\lambda)_\lambda$ of the tree $\mathcal{T}$, and a random z-labeling $(b_\lambda)_\lambda$ of $\mathcal{T}$, the sum $(b_\lambda + c_\lambda)_\lambda$ is a random $(y + z)$-labeling of $\mathcal{T}$. In particular, from any y-labeling $(a_\lambda)_\lambda$ of $\mathcal{T}$, and a random zero-labeling $(b_\lambda)_\lambda$ of $\mathcal{T}$, the values $c_\lambda \leftarrow a_\lambda + b_\lambda$ provide a random y-labeling of $\mathcal{T}$. Similarly, multiplying all the labels of a y-labeling by a constant c leads to a cy-labeling.

Evaluation of a Tree. For an acceptable set Γ for $\mathcal{T}$ and a labeling Λ_y of $\mathcal{T}$ for a random y, given only $(a_\lambda)_{\lambda\in\mathcal{L}_\Gamma}$, one can reconstruct $y = a_\rho$. Indeed, as $\mathcal{T}(\Gamma) = 1$, we use an evaluation pruned tree $\mathcal{T}' \in \mathsf{EPT}(\mathcal{T}, \Gamma)$. Then, in a bottom-up way, starting from the leaves, one can compute the labels of all the internal nodes, up to the root.

Unpredictability of the Root. On the other hand, this is also important to note that, when $\mathcal{T}(\Gamma) = 0$, with a random labeling Λ_y of $\mathcal{T}$ for a random y, given only $(a_\lambda)_{\lambda\in\mathcal{L}_\Gamma}$, y is unpredictable: for any $y, y' \in \mathbb{Z}_p$, $\mathcal{D}_y$ and $\mathcal{D}_{y'}$ are perfectly indistinguishable, where $\mathcal{D}_y = \{(a_\lambda)_{\lambda\in\mathcal{L}_\Gamma}, (a_\lambda)_\lambda \stackrel{\$}{\leftarrow} \Lambda_y(\mathcal{T})\}$. Intuitively, given $(a_\lambda)_{\lambda\in\mathcal{L}_\Gamma}$, as $\mathcal{T}(\Gamma) = 0$, one can complete the labeling so that the label of the root is any y.

In preparation of our proof for Attribute-Indistinguishability, we need to identify a specific property called *independent leaves*, which describes leaves for which the secret share leaks no information in any of the other leaves in the access-tree.

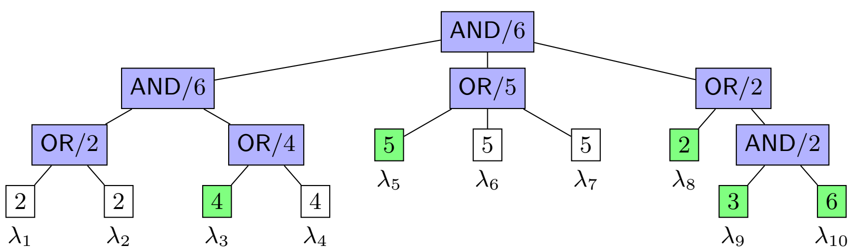

Fig. 5: Example of tree-labeling, with a non-satisfying set of attributes: leaves λ_8, λ_9 and λ_{10} are not independent

Definition 6 (Independent Leaves). *Given an access-tree $\mathcal{T}$ and a set Γ so that $\mathcal{T}(\Gamma) = 0$, we call independent leaves, in $\mathcal{L}_\Gamma$ with respect to $\mathcal{T}$, the leaves μ such that, given only $(a_\lambda)_{\lambda \in \mathcal{L}_\Gamma \setminus \{\mu\}}$, a_μ is unpredictable: for any y, $\mathcal{D}_\$$ and $\mathcal{D}'_y(\mu)$ are perfectly indistinguishable, where*

$$\mathcal{D}_\$ = \{(a_\lambda)_{\lambda \in \mathcal{L}_\Gamma}, y \xleftarrow{\$} \mathbb{Z}_p, (a_\lambda)_\lambda \xleftarrow{\$} \Lambda_y(\mathcal{T})\}$$
$$\mathcal{D}_y(\mu) = \{(a_\mu) \cup (a_\lambda)_{\lambda \in \mathcal{L}_\Gamma \setminus \{\mu\}}, (a_\lambda)_\lambda \xleftarrow{\$} \Lambda_y(\mathcal{T}), a_\mu \xleftarrow{\$} \mathbb{Z}_p\}.$$

With the illustration on Figure 5, with non-satisfied tree, when only colored leaves are set to True, leaves λ_3 and λ_5 are independent among the True leaves $\{\lambda_3, \lambda_5, \lambda_8, \lambda_9, \lambda_{10}\}$. But leaves λ_8, λ_9 and λ_{10} are not independent as $a_{\lambda_8} = a_{\lambda_9} + a_{\lambda_{10}} \bmod 7$ for any random labeling. Intuitively, given $(a_\lambda)_{\lambda \in \mathcal{L}_\Gamma \setminus \{\mu\}}$ and any a_μ, one can complete it into a valid labeling (with any random root label y as $\mathcal{T}(\Gamma) = 0$), for $\mu \in \{3, 5\}$, but not for $\mu \in \{8, 9, 10\}$.

In the appendix B, we provide a variant of Okamoto-Takashima scheme [OT12b], that initially provides some kind of attribute-hiding property, but to handle delegation. This is then in the same vein as [GPSW06], and we prove the Del-IND security under the SXDH assumption. But this is not enough for dealing with traceability, hence our new primitive.

4 Key-Policy ABE with Switchable Attributes

In addition to delegation, our main goal is to allow to arbitrarily introduce switchable attributes in the policies and in the sets of attributes, to give more flexibility to the KP-ABE scheme.

4.1 Definitions

For a Key-Policy ABE with Switchable Attributes (SA-KP-ABE), leaves in the access-tree can be made active or passive, and attributes in the ciphertext can be made valid or invalid. We thus enhance the access-tree $\mathcal{T}$ into $\tilde{\mathcal{T}} = (\mathcal{T}, \mathcal{L}_a, \mathcal{L}_p)$, where the implicit set of leaves $\mathcal{L} = \mathcal{L}_a \uplus \mathcal{L}_p$ is now the explicit disjoint union of the active-leaf and passive-leaf sets. Similarly, a ciphertext will be associated to the pair (Γ_v, Γ_i), also referred as a disjoint union $\Gamma = \Gamma_v \uplus \Gamma_i$, of the valid-attribute and invalid-attribute sets.

We note $\tilde{\mathcal{T}}(\Gamma_v, \Gamma_i) = 1$ if there is an evaluation pruned tree $\mathcal{T}'$ of $\mathcal{T}$ that is satisfied by $\Gamma = \Gamma_v \uplus \Gamma_i$ (*i.e.*, $\mathcal{T}' \in \mathsf{EPT}(\mathcal{T}, \Gamma)$), with the additional condition that all the active leaves in $\mathcal{T}'$ correspond only to valid attributes in Γ_v: $\exists \mathcal{T}' \in \mathsf{EPT}(\mathcal{T}, \Gamma), \forall \lambda \in \mathcal{T}' \cap \mathcal{L}_a$, $A(\lambda) \in \Gamma_v$. In other words, this means that an invalid attribute in the ciphertext should be considered as inexistent for active leaves, but only for those leaves.

We also have to enhance the partial order on $\mathcal{T}$ to $\tilde{\mathcal{T}}$, so that we can deal with delegation: $\tilde{\mathcal{T}}' = (\mathcal{T}', \mathcal{L}'_a, \mathcal{L}'_p) \leq \tilde{\mathcal{T}} = (\mathcal{T}, \mathcal{L}_a, \mathcal{L}_p)$ if and only if $\mathcal{T}' \leq \mathcal{T}$, $\mathcal{L}'_a \cap \mathcal{L}_p = \mathcal{L}'_p \cap \mathcal{L}_a = \emptyset$ and $\mathcal{L}'_a \subseteq \mathcal{L}_a$. More concretely, $\mathcal{T}'$ must be more restrictive, existing leaves cannot change their passive or active status, and new leaves can only be passive.

This allows a more flexible Key-Policy Attribute-Based Encapsulation with Switchable Attributes:

$\mathsf{Setup}(1^\kappa)$. From the security parameter κ, the algorithm defines all the global parameters PK, the secret key SK and the master secret key MK;

$\mathsf{KeyGen}(\mathsf{MK}, \tilde{\mathcal{T}})$. The algorithm outputs a key $\mathsf{dk}_{\tilde{\mathcal{T}}}$ which enables the user to decapsulate keys encrypted under a set of attributes $\Gamma = \Gamma_v \uplus \Gamma_i$ if and only if $\tilde{\mathcal{T}}(\Gamma_v, \Gamma_i) = 1$;

$\mathsf{Delegate}(\mathsf{dk}_{\tilde{\mathcal{T}}}, \tilde{\mathcal{T}}')$. Given a key $\mathsf{dk}_{\tilde{\mathcal{T}}}$ and a more restrictive access-tree $\tilde{\mathcal{T}}' \leq \tilde{\mathcal{T}}$, the algorithm outputs a decryption key $\mathsf{dk}_{\tilde{\mathcal{T}}'}$;

$\mathsf{Encaps}(\mathsf{PK}, \Gamma)$. For a set Γ of attributes, the algorithm generates the ciphertext C and an encapsulated key K;

$\mathsf{Encaps}^*(\mathsf{SK}, \Gamma_v, \Gamma_i)$. For a pair (Γ_v, Γ_i) of disjoint sets of attributes, the algorithm generates the ciphertext C and an encapsulated key K;

$\mathsf{Decaps}(\mathsf{dk}_{\tilde{\mathcal{T}}}, C)$**.** Given the key $\mathsf{dk}_{\tilde{\mathcal{T}}}$ and the ciphertext C, the algorithm outputs the encapsulated key K.

One can note the difference between Encaps with PK and Encaps* with SK, where the former runs the latter on the pair $(\Gamma, \emptyset)$. And as $\Gamma_i = \emptyset$, the public key is enough. This is thus still a public-key encryption scheme when only valid attributes are in the ciphertext, but the invalidation of some attributes require a secret information. For correctness, the Decaps algorithm should output the encapsulated key K if and only if C has been generated for a pair (Γ_v, Γ_i) that satisfies the policy $\tilde{\mathcal{T}}$ of the decryption key $\mathsf{dk}_{\tilde{\mathcal{T}}}$: $\tilde{\mathcal{T}}(\Gamma_v, \Gamma_i) = 1$. The following security notion enforces this property. But some other indistinguishability notions need to be defined in order to be able to exploit these switchable attributes in more complex protocols. For instance, this could be used to trace traitors in a multi-device setting or provide some anonymity properties.

4.2 Security Notions

Definition 7 (Delegation-Indistinguishability for SA-KP-ABE). *Del-IND security for SA-KP-ABE is defined by the following game:*

Initialize: *The challenger runs the* Setup *algorithm of SA-KP-ABE and gives the public parameters* PK *to the adversary;*

Oracles: *The following oracles can be called in any order and any number of times, except for* RoREncaps *which can be called only once.*

- $\mathsf{OKeyGen}(\tilde{\mathcal{T}})$**:** *The adversary is allowed to issue* KeyGen*-queries for any access-tree $\tilde{\mathcal{T}} = (\mathcal{T}, \mathcal{L}_a, \mathcal{L}_p)$ of its choice, without getting back the decryption key, but for future delegation;*
- $\mathsf{ODelegate}(\tilde{\mathcal{T}}, \tilde{\mathcal{T}}')$**:** *The adversary is allowed to issue several* Delegate*-queries for any more restrictive access-tree $\tilde{\mathcal{T}}' \leq \tilde{\mathcal{T}}$ of its choice, for an already generated decryption key under $\tilde{\mathcal{T}}$, and gets back the decryption key $\mathsf{dk}_{\tilde{\mathcal{T}}'}$;*
- $\mathsf{OEncaps}(\Gamma_v, \Gamma_i)$**:** *The adversary may be allowed to issue* Encaps**-queries on sets of attributes (Γ_v, Γ_i), and it gets back the encapsulation $C = \mathsf{Encaps}^*(\Gamma_v, \Gamma_i)$;*
- $\mathsf{RoREncaps}(\Gamma_v, \Gamma_i)$**:** *The adversary submits a unique real-or-random encapsulation query on a set of attributes $\Gamma = \Gamma_v \cup \Gamma_i$. The challenger asks for an encapsulation query on (Γ_v, Γ_i) and receives (K_0, C). It also generates a random key K_1. It eventually flips a random coin b, and outputs (K_b, C) to the adversary;*

$\mathbf{Finalize}(b')$**:** *The adversary outputs a guess b' for b. If for some access-tree $\tilde{\mathcal{T}}'$ asked to the* ODelegate*-oracle, $\tilde{\mathcal{T}}'(\Gamma_v, \Gamma_i) = 1$, on the challenge set (Γ_v, Γ_i), $\beta \stackrel{\$}{\leftarrow} \{0,1\}$, otherwise one sets $\beta = b'$. One outputs β.*

The advantage of an adversary $\mathcal{A}$ in this game is defined as

$$\mathsf{Adv}^{\mathsf{del\text{-}ind}}(\mathcal{A}) = \Pr[\beta = 1 | b = 1] - \Pr[\beta = 1 | b = 0].$$

In the basic form of Del-IND-security, where Encaps* encapsulations are not available, the RoREncaps-oracle only allows $\Gamma_i = \emptyset$, and no OEncaps-oracle is available. But as Encaps (with $\Gamma_i = \emptyset$) is a public-key algorithm, the adversary can generate valid ciphertexts by himself. We will call it "Del-IND-security for Encaps". For the more advanced security level, RoREncaps-query will be allowed on any pair (Γ_v, Γ_i), with the additional OEncaps-oracle. We will call it "Del-IND-security for Encaps*".

With these disjoint unions of $\mathcal{L} = \mathcal{L}_a \cup \mathcal{L}_p$ and $\Gamma = \Gamma_v \cup \Gamma_i$, we will also consider some indistinguishability notions on $(\mathcal{L}_a, \mathcal{L}_p)$ and (Γ_v, Γ_i), about which leaves are active or passive in $\mathcal{L} = \mathcal{L}_a \cup \mathcal{L}_p$ for a given key, and which attributes are valid or invalid in $\Gamma = \Gamma_v \cup \Gamma_i$ for a given ciphertext. The former will be the key-indistinguishability, whereas the latter will be attribute-indistinguishability. Again, as the encapsulation Encaps is public-key, the adversary can generate valid encapsulations by himself. However, we may provide access to an OEncaps-oracle to allow Encaps^* queries, but with constraints in the final step, to exclude trivial attacks against key-indistinguishability. Similarly there will be constraints in the final step on the OKeyGen/ODelegate-oracle queries for the attribute-indistinguishability.

Definition 8 (Key-Indistinguishability). *Key-IND security for SA-KP-ABE is defined by the following game:*

Initialize: *The challenger runs the* Setup *algorithm of SA-KP-ABE and gives the public parameters* PK *to the adversary;*

Oracles: $\mathsf{OKeyGen}(\tilde{\mathcal{T}})$, $\mathsf{ODelegate}(\tilde{\mathcal{T}}, \tilde{\mathcal{T}}')$, $\mathsf{OEncaps}(\Gamma_v, \Gamma_i)$, *as above;*

$\mathsf{RoAPKeyGen}(\tilde{\mathcal{T}})$**:** *The adversary submits one Real or All-Passive* KeyGen*-query for any access structure* $\tilde{\mathcal{T}}$ *of its choice, with a list* $\mathcal{L} = \mathcal{L}_a \cup \mathcal{L}_p$ *of active and passive attributes, and gets back* $\mathsf{dk}_0 = \mathsf{KeyGen}(\mathsf{MK}, (\mathcal{T}, \mathcal{L}_a, \mathcal{L}_p))$ *or* $\mathsf{dk}_1 = \mathsf{KeyGen}(\mathsf{MK}, (\mathcal{T}, \emptyset, \mathcal{L}))$*. It eventually flips a random coin* b*, and outputs* dk_b *to the adversary;*

Finalize(b'): *The adversary outputs a guess* b' *for* b*. If for some* (Γ_v, Γ_i) *asked to the* OEncaps*-oracle,* $\mathcal{T}(\Gamma_v \cup \Gamma_i) = 1$*, for the challenge access-tree* $\mathcal{T}$ *where* $\mathcal{L} = \mathcal{L}_a \cup \mathcal{L}_p$*,* $\beta \stackrel{\$}{\leftarrow} \{0,1\}$*, otherwise one sets* $\beta = b'$*. One outputs* β*.*

The advantage of an adversary $\mathcal{A}$ *in this game is defined as*

$$\mathsf{Adv}^{\mathsf{key\text{-}ind}}(\mathcal{A}) = \Pr[\beta = 1 | b = 1] - \Pr[\beta = 1 | b = 0].$$

For the constraints in the final step, we require the adversary not to ask for an encapsulation on attributes that would be accepted by the policy with all-passive keys.

An alternative version can be defined where the **Finalize(b')** checks whether some active leaf $\lambda \in \mathcal{L}_a$ from the challenge key corresponds to some invalid attribute $t \in \Gamma_i$ in an OEncaps-query, and then sets $\beta \stackrel{\$}{\leftarrow} \{0,1\}$, otherwise one sets $\beta = b'$. This is the **Distinct Key-Indistinguishability** (dKey-IND), as we expect active leaves and invalid attributes to be distinct in the challenge key and the obtained ciphertexts.

Definition 9 (Attribute-Indistinguishability). *Att-IND security for SA-KP-ABE is defined by the following game:*

Initialize: *The challenger runs the* Setup *algorithm of SA-KP-ABE and gives the public parameters* PK *to the adversary;*

Oracles: $\mathsf{OKeyGen}(\tilde{\mathcal{T}})$, $\mathsf{ODelegate}(\tilde{\mathcal{T}}, \tilde{\mathcal{T}}')$, $\mathsf{OEncaps}(\Gamma_v, \Gamma_i)$, *as above;*

$\mathsf{RoAVEncaps}(\Gamma_v, \Gamma_i)$**:** *The adversary submits one Real-or-All-Valid encapsulation query on distinct sets of attributes* (Γ_v, Γ_i)*. The challenger generates a ciphertext* $C = \mathsf{Encaps}^*(\mathsf{SK}, \Gamma_v, \Gamma_i)$ *as the real case, if* $b = 0$*, or* $C = \mathsf{Encaps}(\mathsf{PK}, \Gamma_v \cup \Gamma_i)$ *as the all-valid case, if* $b = 1$*, and outputs* C *to the adversary;*

Finalize(b'): *The adversary outputs a guess* b' *for* b*. If for some access-tree* $\tilde{\mathcal{T}}'$ *asked to the* ODelegate*-oracle,* $\tilde{\mathcal{T}}'(\Gamma_v \cup \Gamma_i, \emptyset) = 1$*, on the challenge set* (Γ_v, Γ_i)*,* $\beta \stackrel{\$}{\leftarrow} \{0,1\}$*, otherwise one sets* $\beta = b'$*. One outputs* β*.*

The advantage of an adversary $\mathcal{A}$ *in this game is defined as*

$$\mathsf{Adv}^{\mathsf{att\text{-}ind}}(\mathcal{A}) = \Pr[\beta = 1 | b = 1] - \Pr[\beta = 1 | b = 0].$$

This definition is a kind of attribute-hiding, where a user with keys for access-trees that are not satisfied by $\Gamma = \Gamma_v \cup \Gamma_i$ cannot distinguish valid from invalid attributes in the ciphertext.

An alternative version can be defined where the **Finalize(b')** checks whether some attribute $t \in \Gamma_i$ from the challenge query corresponds to some active leaf $\lambda \in \mathcal{L}'_a$ in a ODelegate-query, and then sets $\beta \stackrel{\$}{\leftarrow} \{0,1\}$, otherwise one sets $\beta = b'$. This is the **Distinct Attribute-Indistinguishability** (dAtt-IND), as we expect active leaves and invalid attributes to be distinct in the obtained keys and the challenge ciphertext.

5 A SA-KP-ABE Scheme

5.1 Description of our KP-ABE with Switchable Attributes

We extend the basic KP-ABE scheme proven in the appendix B, with leaves that can be made active or passive in a decryption key, and some attributes can be made valid or invalid in a ciphertext, and prove that it still achieves the Del-IND-security. For our construction, we will use two DPVS, of dimensions 3 and 7 respectively, in a pairing-friendly setting $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, e, G_1, G_2, q)$, using the notations introduced in Section 2.3. Essentially, we introduce a 7-th component (to the simpler scheme from the appendix B) to deal with switchable attributes. The two new basis-vectors $\mathbf{d}_7$ and $\mathbf{d}_7^*$ are in the secret key SK and the master secret key MK respectively:

$\mathsf{Setup}(1^\kappa)$**.** The algorithm chooses two random dual orthogonal bases

$$\mathbb{B} = (\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3) \qquad \mathbb{B}^* = (\mathbf{b}_1^*, \mathbf{b}_2^*, \mathbf{b}_3^*)$$
$$\mathbb{D} = (\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6, \mathbf{d}_7) \qquad \mathbb{D}^* = (\mathbf{d}_1^*, \mathbf{d}_2^*, \mathbf{d}_3^*, \mathbf{d}_4^*, \mathbf{d}_5^*, \mathbf{d}_6^*, \mathbf{d}_7^*).$$

It sets the public parameters $\mathsf{PK} = \{(\mathbf{b}_1, \mathbf{b}_3, \mathbf{b}_1^*), (\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_1^*, \mathbf{d}_2^*, \mathbf{d}_3^*)\}$, whereas the master secret key is $\mathsf{MK} = \{\mathbf{b}_3^*, \mathbf{d}_7^*\}$ and the secret key is $\mathsf{SK} = \{\mathbf{d}_7\}$. Other basis vectors are kept hidden

$\mathsf{KeyGen}(\mathsf{MK}, \tilde{\mathcal{T}})$**.** For an extended access-tree $\tilde{\mathcal{T}} = (\mathcal{T}, \mathcal{L}_a, \mathcal{L}_p)$, the algorithm first chooses a random $a_0 \xleftarrow{\$} \mathbb{Z}_q$, and a random a_0-labeling $(a_\lambda)_\lambda$ of the access-tree $\mathcal{T}$, and builds the key:

$$\mathbf{k}_0^* = (a_0, 0, 1)_{\mathbb{B}^*} \qquad \mathbf{k}_\lambda^* = (\pi_\lambda(1, t_\lambda), a_\lambda, 0, 0, 0, r_\lambda)_{\mathbb{D}^*}$$

for all the leaves λ, where $t_\lambda = A(\lambda)$, $\pi_\lambda \xleftarrow{\$} \mathbb{Z}_q$, and $r_\lambda \xleftarrow{\$} \mathbb{Z}_q^*$ if λ is an active leaf in the key ($\lambda \in \mathcal{L}_a$) or else $r_\lambda = 0$ for a passive leaf ($\lambda \in \mathcal{L}_p$). The decryption key $\mathsf{dk}_{\tilde{\mathcal{T}}}$ is then $(\mathbf{k}_0^*, (\mathbf{k}_\lambda^*)_\lambda)$.

$\mathsf{Delegate}(\mathsf{dk}_{\tilde{\mathcal{T}}}, \tilde{\mathcal{T}}')$**.** Given a private key for a tree $\tilde{\mathcal{T}}$ and a more restrictive subtree $\tilde{\mathcal{T}}' \leq \tilde{\mathcal{T}}$, the algorithm creates a delegated key $\mathsf{dk}_{\tilde{\mathcal{T}}'}$. It chooses a random $a'_0 \xleftarrow{\$} \mathbb{Z}_q$ and a random a'_0-labeling $(a'_\lambda)_\lambda$ of $\mathcal{T}'$; Then, it updates $\mathbf{k}_0^* \leftarrow \mathbf{k}_0^* + (a'_0, 0, 0)_{\mathbb{B}^*}$; It sets $\mathbf{k}_\lambda^* \leftarrow (\pi'_\lambda \cdot (1, t_\lambda), a'_\lambda, 0, 0, 0)_{\mathbb{B}^*}$ for a new leaf, or updates $\mathbf{k}_\lambda^* \leftarrow \mathbf{k}_\lambda^* + (\pi'_\lambda \cdot (1, t_\lambda), a'_\lambda, 0, 0, 0)_{\mathbb{B}^*}$ for an old leaf, with $\pi'_\lambda \xleftarrow{\$} \mathbb{Z}_q$.

$\mathsf{Encaps}(\mathsf{PK}, \Gamma)$**.** For a set Γ of attributes, the algorithm first chooses random scalars $\omega, \xi \xleftarrow{\$} \mathbb{Z}_q$. It then sets $K = g_t^\xi$ and generates the ciphertext $C = (\mathbf{c}_0, (\mathbf{c}_t)_{t\in\Gamma})$ where

$$\mathbf{c}_0 = (\omega, 0, \xi)_{\mathbb{B}} \qquad \mathbf{c}_t = (\sigma_t(t, -1), \omega, 0, 0, 0, 0)_{\mathbb{D}}$$

for all the attributes $t \in \Gamma$, with $\sigma_t \xleftarrow{\$} \mathbb{Z}_q$.

$\mathsf{Encaps}^*(\mathsf{SK}, (\Gamma_v, \Gamma_i))$**.** For a disjoint union $\Gamma = \Gamma_v \uplus \Gamma_i$ of sets of attributes (Γ_v is the set of valid attributes and Γ_i is the set of invalid attributes), the algorithm first chooses random scalars $\omega, \xi \xleftarrow{\$} \mathbb{Z}_q$. It then sets $K = g_t^\xi$ and generates the ciphertext $C = (\mathbf{c}_0, (\mathbf{c}_t)_{t\in(\Gamma_v \uplus \Gamma_i)})$ where

$$\mathbf{c}_0 = (\omega, 0, \xi)_{\mathbb{B}} \qquad \mathbf{c}_t = (\sigma_t(t, -1), \omega, 0, 0, 0, u_t)_{\mathbb{D}}$$

for all the attributes $t \in \Gamma_v \uplus \Gamma_i$, $\sigma_t \xleftarrow{\$} \mathbb{Z}_q$ and $u_t \xleftarrow{\$} \mathbb{Z}_q^*$ if $t \in \Gamma_i$ or $u_t = 0$ if $t \in \Gamma_v$.

$\mathsf{Decaps}(\mathsf{dk}_{\tilde{\mathcal{T}}}, C)$**.** The algorithm first selects an evaluation pruned tree $\mathcal{T}'$ of $\mathcal{T}$ that is satisfied by $\Gamma = \Gamma_v \cup \Gamma_i$, such that any leaf λ in $\mathcal{T}'$ is either passive in the key ($\lambda \in \mathcal{L}_p$) or associated to a valid attribute in the ciphertext ($t_\lambda \in \Gamma_v$). This means that the labels a_λ for all the leaves λ in $\mathcal{T}'$ allow to reconstruct a_0 by simple additions, where $t = t_\lambda$:

$$\mathbf{c}_t \times \mathbf{k}_\lambda^* = g_t^{\sigma_t \cdot \pi_\lambda \cdot \langle (t,-1),(1,t_\lambda)\rangle + \omega \cdot a_\lambda + u_t \cdot r_\lambda} = g_t^{\omega \cdot a_\lambda},$$

as $u_t = 0$ or $r_\lambda = 0$. Hence, the algorithm can derive $g_t^{\omega \cdot a_0}$. From $\mathbf{c}_0$ and $\mathbf{k}_0^*$, it can also compute $\mathbf{c}_0 \times \mathbf{k}_0^* = g_t^{\omega \cdot a_0 + \xi}$, which then easily leads to $K = g_t^\xi$.

First, note that the delegation works as $\mathbf{b}_1^*, \mathbf{d}_1^*, \mathbf{d}_2^*, \mathbf{d}_3^*$ are public. This allows to create a new key for $\tilde{\mathcal{T}}' \le \tilde{\mathcal{T}}$. But as $\mathbf{d}_7^*$ is not known, any new leaf is necessarily passive, and an active existing leaf in the original key cannot be converted to passive, and vice-versa. Indeed, all the randomnesses are fresh, excepted the last components r_λ that remain unchanged: this is perfectly consistent with the definition of $\tilde{\mathcal{T}}' \le \tilde{\mathcal{T}}$.

Second, in encapsulation, for invalidating a contribution $\mathbf{c}_t$ in the ciphertext with a non-zero u_t, for $t \in \Gamma_i$, one needs to know $\mathbf{d}_7$, hence the Encaps^* that requires SK, whereas Encaps with $\Gamma_i = \emptyset$ just needs PK.

Eventually, we stress that in the above decryption, one can recover $g_t^{\omega \cdot a_0}$ if and only if there is an evaluation pruned tree $\mathcal{T}'$ of $\mathcal{T}$ that is satisfied by Γ and the active leaves in $\tilde{\mathcal{T}}'$ correspond to valid attributes in Γ_v (used during the encapsulation). And this holds if and only if $\tilde{\mathcal{T}}(\Gamma_v, \Gamma_i) = 1$.

5.2 Del-IND-Security of our SA-KP-ABE for Encaps

For this security notion, we first consider only valid contributions in the challenge ciphertext, with indistinguishability of the Encaps algorithm. Which means that $\Gamma_i = \emptyset$ in the challenge pair. And the security result holds even if the vector $\mathbf{d}_7$ is made public:

Theorem 10. *Our SA-KP-ABE scheme is Del-IND for* Encaps *(with only valid attributes in the challenge ciphertext), even if* $\mathbf{d}_7$ *is public.*

The proof can be found in the appendix C.1. It essentially reduces to the IND-security result of the KP-ABE scheme in the appendix B.3.

We now detail an overview of the proof, as the structure of the first games is common among most of our proofs. The global sequence of games is described on Figure 6, with another sequence of sub-games on Figure 7.

$\mathbf{G}_0$ Real Del-IND-Security game

$$\begin{array}{lllll|llll}
\mathbf{c}_0 = (\ \omega & 0 & \xi\ ) & \mathbf{c}_t = (\ \sigma_t(1,t) & \omega & 0 & 0 & 0 & u_t\ ) \\
\mathbf{k}^*_{\ell,0} = (\ a_{\ell,0} & 0 & 1\ ) & \mathbf{k}^*_{\ell,\lambda} = (\ \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) & a_{\ell,\lambda} & 0 & 0 & 0 & r_{\ell,\lambda}\ )
\end{array}$$

$\mathbf{G}_1$ $\mathsf{SubSpace\text{-}Ind}$ Property, on $(\mathbb{B},\mathbb{B}^*)_{1,2}$ and $(\mathbb{D},\mathbb{D}^*)_{3,4}$, between 0 and $\tau \xleftarrow{\$} \mathbb{Z}_q$

$$\begin{array}{lllll|llll}
\mathbf{c}_0 = (\ \omega & \colorbox{lightgray}{τ} & \xi\ ) & \mathbf{c}_t = (\ \sigma_t(1,t) & \omega & \colorbox{lightgray}{τ} & 0 & 0 & u_t\ ) \\
\mathbf{k}^*_{\ell,0} = (\ a_{\ell,0} & 0 & 1\ ) & \mathbf{k}^*_{\ell,\lambda} = (\ \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) & a_{\ell,\lambda} & 0 & 0 & 0 & r_{\ell,\lambda}\ )
\end{array}$$

$\mathbf{G}_2$ $\mathsf{SubSpace\text{-}Ind}$ Property, on $(\mathbb{D},\mathbb{D}^*)_{1,2,6}$, between 0 and τz_t

$$\begin{array}{lllll|llll}
\mathbf{c}_0 = (\ \omega & \tau & \xi\ ) & \mathbf{c}_t = (\ \sigma_t(1,t) & \omega & \tau & 0 & \colorbox{lightgray}{τz_t} & u_t\ ) \\
\mathbf{k}^*_{\ell,0} = (\ a_{\ell,0} & 0 & 1\ ) & \mathbf{k}^*_{\ell,\lambda} = (\ \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) & a_{\ell,\lambda} & 0 & 0 & 0 & r_{\ell,\lambda}\ )
\end{array}$$

$\mathbf{G}_3$ Introduction of an additional random-labeling. See Figure 7

$$\begin{array}{lllll|llll}
\mathbf{c}_0 = (\ \omega & \tau & \xi\ ) & \mathbf{c}_t = (\ \sigma_t(1,t) & \omega & \tau & 0 & \tau z_t & u_t\ ) \\
\mathbf{k}^*_{\ell,0} = (\ a_{\ell,0} & \colorbox{lightgray}{$r_{\ell,0}$} & 1\ ) & \mathbf{k}^*_{\ell,\lambda} = (\ \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) & a_{\ell,\lambda} & 0 & 0 & \colorbox{lightgray}{$s_{\ell,\lambda}/z_{t_{\ell,\lambda}}$} & r_{\ell,\lambda}\ )
\end{array}$$

$\mathbf{G}_4$ Formal basis change, on $(\mathbb{B},\mathbb{B}^*)_{2,3}$, to randomize ξ

$$\begin{array}{lllll|llll}
\mathbf{c}_0 = (\ \omega & \tau & \colorbox{lightgray}{ξ''}\ ) & \mathbf{c}_t = (\ \sigma_t(1,t) & \omega & \tau & 0 & \tau z_t & u_t\ ) \\
\mathbf{k}^*_{\ell,0} = (\ a_{\ell,0} & r_{\ell,0} & 1\ ) & \mathbf{k}^*_{\ell,\lambda} = (\ \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) & a_{\ell,\lambda} & 0 & 0 & s_{\ell,\lambda}/z_{t_{\ell,\lambda}} & r_{\ell,\lambda}\ )
\end{array}$$

Gray cells $\colorbox{lightgray}{$x$}$ mean they have been changed in this game.

Fig. 6: Global sequence of games for the Del-IND-security proof of our SA-KP-ABE

In the two first games $\mathbf{G}_1$ and $\mathbf{G}_2$, one is preparing the floor with a random τ and random masks z_t in the ciphertexts $\mathbf{c}_t$ (actually, the challenge ciphertext corresponding to the attribute t). Note that until the challenge query is asked, one does not exactly know the attributes in Γ (as we are in the adaptive-set setting), but we prepare all the possible $\mathbf{c}_t$, and only the ones corresponding to attributes in Γ will be provided to the adversary. The main step is to get to

$\mathbf{G}_{2.k.0}$ Hybrid game for $\mathbf{G}_2$, with $1 \leq k \leq K+1$ (from Figure 6)

$$\begin{array}{lllll}
 & \mathbf{c}_0 = (\ \omega\ \ \tau\ \ \xi\ ) & \mathbf{c}_t = (\ \sigma_t(1,t) & \omega & \mid\ \tau\ \ 0\ \ \tau z_t\ \ u_t\ ) \\
\ell < k & \mathbf{k}^*_{\ell,0} = (\ a_{\ell,0}\ \ r_{\ell,0}\ \ 1\ ) & \mathbf{k}^*_{\ell,\lambda} = (\ \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) & a_{\ell,\lambda} & \mid\ 0\ \ 0\ \ s_{\ell,\lambda}/z_{t_{\ell,\lambda}}\ \ r_{\ell,\lambda}\ ) \\
\ell \geq k & \mathbf{k}^*_{\ell,0} = (\ a_{\ell,0}\ \ 0\ \ 1\ ) & \mathbf{k}^*_{\ell,\lambda} = (\ \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) & a_{\ell,\lambda} & \mid\ 0\ \ 0\ \ 0\ \ r_{\ell,\lambda}\ )
\end{array}$$

$\mathbf{G}_{2.k.1}$ SubSpace-Ind Property, on $(\mathbb{B}^*,\mathbb{B})_{1,2}$ and $(\mathbb{D}^*,\mathbb{D})_{3,4}$, between 0 and $s_{k,*}$

$$\mathbf{k}^*_{k,0} = (\ a_{k,0}\ \ \boxed{s_{k,0}}\ \ 1\ ) \quad \mathbf{k}^*_{k,\lambda} = (\ \pi_{k,\lambda}(t_{k,\lambda},-1)\ \ a_{k,\lambda}\ \mid\ \boxed{s_{k,\lambda}}\ \ 0\ \ 0\ \ r_{k,\lambda}\ )$$

$\mathbf{G}_{2.k.2}$ Masking of the labeling. See 12 for the Encaps proof, or 8 for the Encaps* one

$$\mathbf{k}^*_{k,0} = (\ a_{k,0}\ \ s_{k,0}\ \ 1\ ) \quad \mathbf{k}^*_{k,\lambda} = (\ \pi_{k,\lambda}(t_{k,\lambda},-1)\ \ a_{k,\lambda}\ \mid\ \boxed{0}\ \ 0\ \ \boxed{s_{k,\lambda}/z_{t_{k,\lambda}}}\ \ r_{k,\lambda}\ )$$

$\mathbf{G}_{2.k.3}$ Limitations on KeyGen-queries: $s_{k,0}$ unpredictable, replaced by a random $r_{k,0}$

$$\mathbf{k}^*_{k,0} = (\ a_{k,0}\ \ \boxed{r_{k,0}}\ \ 1\ ) \quad \mathbf{k}^*_{k,\lambda} = (\ \pi_{k,\lambda}(t_{k,\lambda},-1)\ \ a_{k,\lambda}\ \mid\ 0\ \ 0\ \ s_{k,\lambda}/z_{t_{k,\lambda}}\ \ r_{k,\lambda}\ )$$

Fig. 7: Sequence of games on the K keys for the Del-IND-security proof of our SA-KP-ABE

Game $\mathbf{G}_3$, with an additional labeling $(s_{\ell,0},(s_{\ell,\lambda})_\lambda)$, using hybrid games starting from Game $\mathbf{G}_2$. The sequence on Figure 7 gives more details: the new labelling is added in each ℓ-th key (in $\mathbf{G}_{2.k.1}$), then each label is masked by the random z_t for each attribute t (in $\mathbf{G}_{2.k.2}$). In order to go to game $\mathbf{G}_{2.k.3}$ one exploits the limitations one expects from the adversary in the security game: the adversary cannot ask keys on access-trees $\mathcal{T}$ such that $\mathcal{T}(\Gamma)=1$, for the challenge set Γ. To formally make this final step to work, we need another level of hybrid games, in the same vein as the one presented on Figure 12, in the appendix B.3

Simulation of delegation can just be done by using the key generation algorithm, making sure we use the same randomness for all the keys delegated from the same one. As the vector $\mathbf{d}^*_7$ is known to the simulator, this is easy. As $\mathbf{d}_7$ is public, the adversary can run by himself both Encaps and Encaps*.

We stress that our construction makes more basis vectors public, than in the schemes from [OT12b], as only $\mathbf{b}^*_3$ is for the key issuer. This makes the proof more tricky, but this is the reason why we can deal with delegation for any user.

5.3 Del-IND-Security of our SA-KP-ABE for Encaps*

We now study the full indistinguishability of the ciphertext generated by an Encaps* challenge, with delegated keys. The intuition is that when $u_t \cdot r_{\ell,\lambda} \neq 0$, the share $a_{\ell,\lambda}$ in $g_t^{\omega \cdot a_{\ell,\lambda}+u_t \cdot r_{\ell,\lambda}}$ cannot be recovered, but we have to formally prove it. As above, the idea of the proof is to introduce an additional labeling in every ℓ-th key, in the hidden components, that will only impact the challenge ciphertext. When the corresponding ciphertext $\mathbf{c}_t$ will be missing in the challenge ciphertext, the label $s_{\ell,\lambda}$ will be unknown; when the ciphertext $\mathbf{c}_t$ is present but invalid, we will show that the label can be randomly altered if $u_{t_{\ell,\lambda}} \cdot r_{\ell,\lambda} \neq 0$, making it unusable. Then, the root of the labeling is indistinguishable from random.

As shown in the previous proof, we can simulate all the delegated keys as if they were original keys, except that we have to make sure we use the same r_λ for the leaves that originate from the same initial key.

In addition, in the security proof, we will need to anticipate whether $u_t \cdot r_{\ell,\lambda} = 0$ or not when simulating the keys, and the challenge ciphertext as well (even before knowing the exact query (Γ_v,Γ_i)). Without being in the selective-set setting where both Γ_v, and Γ_i would have to be specified before generating the public parameters PK, we ask to know disjoint super-sets $A_v, A_i \subseteq \mathcal{U}$ of attributes. Then, in the challenge ciphertext query, we will ask that $\Gamma_v \subseteq A_v$ and $\Gamma_i \subseteq A_i$. We will call this setting the *semi-adaptive super-set* setting, where the super-sets have to be specified before the first decryption keys are issued. Furthermore, the set of attributes $\Gamma = \Gamma_v \uplus \Gamma_i$ used in the real challenge query is only specified at the moment of the challenge, as in the adaptive setting.

For this proof, $\mathbf{d}_7$ must be kept secret (cannot be provided to the adversary). We will thus give access to an Encaps* oracle. We also have to show how to simulate it.

Theorem 11. *Our SA-KP-ABE scheme is Del-IND for Encaps*, in the semi-adaptive super-set setting (where $A_v, A_i \subseteq \mathcal{U}$ so that $\Gamma_v \subseteq A_v$ and $\Gamma_i \subseteq A_i$ are specified before asking for keys).*

We stress that the semi-adaptive super-set setting is much stronger than the selective-set setting where the adversary would have to specify both Γ_v and Γ_i before the setup. Here only super-sets have to be specified, and just before the first key-query. The adversary is thus given much more power.

The full proof can be found in the appendix C.2, but as this is a major result of this paper, we provide some hints here. The idea of the sequence is as above to introduce an additional labeling $(s_{\ell,0}, (s_{\ell,\lambda})_\lambda)$ in hidden components in each ℓ-th key (in $\mathbf{G}_{2.k.1}$, from Figure 7), where each label is masked by a random z_t for each attribute t (in $\mathbf{G}_{2.k.2}$). In our setting, we only consider keys that are really provided to the adversary, and thus delegated keys. They can be generated as fresh keys excepted the r_λ's that have to be the same for leaves in keys delegated from the same initial key. However, in order to go to game $\mathbf{G}_{2.k.3}$, one cannot directly conclude that $s_{k,0}$ is independent from the view of the adversary: we only know $\tilde{\mathcal{T}}_k(\Gamma_v, \Gamma_i) = 0$, but not necessarily $\mathcal{T}_k(\Gamma_v \uplus \Gamma_i) = 0$, as in the previous proof.

To this aim, we revisit this gap with an additional sequence presented in the Figure 8 where we focus on the k-th key and the challenge ciphertext. In that sequence, we first prepare with additional random values $y_{\ell,\lambda}$, with the same repetition properties as the $r_{\ell,\lambda}$. Thereafter, in another sub-sequence of games on the attributes, presented in details in Figure 14, after some additional preparation, we can use the Swap-Ind property in Games $\mathbf{G}_{2.k.2.3.p.5}$ and $\mathbf{G}_{2.k.2.3.p.7}$, to completely randomize $s_{k,\lambda}$ when $u_{t_{k,\lambda}} \cdot r_{k,\lambda} \neq 0$. Hence, the $s_{k,\lambda}$ are unknown either when $z_{t_{k,\lambda}}$ is not known (the corresponding element is not provided in the challenge ciphertext) or this is a random $s'_{k,\lambda}$ when $u_{t_{k,\lambda}} \cdot r_{k,\lambda} \neq 0$. The property of the access-tree then makes $s_{k,0}$ perfectly unpredictable, which can be replaced by a random independent $r_{k,0}$.

5.4 Distinct Indistinguishability Properties

We first claim easy results, for which the proofs are symmetrical:

Theorem 12. *Our SA-KP-ABE scheme is dKey-IND, even if $\mathbf{d}_7^*$ is public.*

Theorem 13. *Our SA-KP-ABE scheme is dAtt-IND, even if $\mathbf{d}_7$ is public.*

Both proofs can be found in the appendix C.3 and the appendix C.4. In these alternative variants, all the invalid attributes in all the queried ciphertexts do not correspond to any active leaf in the challenge keys (for dKey-IND) or all active leaves in all the queried keys do not correspond to any invalid attribute in the challenge ciphertext (for dAtt-IND). Then, we can gradually replace all the real keys by all-passive in the former proof or all the real ciphertexts by all-valid in the latter proof. In both cases, the advantage is bounded by $(4+2P)P \cdot \mathsf{Adv}^{\mathsf{sxdh}}(t)$, where P is the number of attributes involved during the game, in a key or in a ciphertext.

5.5 Attribute-Indistinguishability

Theorem 14. *Our SA-KP-ABE scheme is Att-IND, even if $\mathbf{d}_7$ is public, if all the active keys correspond to independent leaves with respect to the set of attributes $\Gamma = \Gamma_v \uplus \Gamma_i$ in the challenge ciphertext.*

The proof can be found in the appendix C.5. This is also an important result with respect to our target application of tracing, combined with possible revocation. Indeed, with such a result, if a user is revoked independently of the tracing procedure (the policy would reject him even if all his passive leaves match valid attributes in the ciphertext), he will not be able to detect whether there are invalid attributes in the ciphertext and thus that the ciphertext is from a tracing procedure. This gives us a strong resistance to collusion. The sequence of

$\mathbf{G}_{2.k.2.0}$ Intermediate sequence from $\mathbf{G}_{2.k.2}$ (from Figure 11)

$$
\begin{array}{lrcl}
 & \mathbf{c}_t = (& \sigma_t(1,t) \quad \omega & | \quad \tau \quad 0 \quad \tau z_t \quad u_t \;) \\
\ell < k & \mathbf{k}^*_{\ell,0} = (& a_{\ell,0} \quad r_{\ell,0} \quad 1\;) & \\
 & \mathbf{k}^*_{\ell,\lambda} = (& \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) \;\; a_{\ell,\lambda} & | \quad 0 \quad 0 \quad s'_{\ell,\lambda}/z_{t_{\ell,\lambda}} \quad r_{\ell,\lambda} \;) \\
\ell = k & \mathbf{k}^*_{k,0} = (& a_{k,0} \quad s_{k,0} \quad 1\;) & \\
 & \mathbf{k}^*_{k,\lambda} = (& \pi_{k,\lambda}(t_{k,\lambda},-1) \;\; a_{k,\lambda} & | \quad 0 \quad 0 \quad s_{k,\lambda}/z_{t_{k,\lambda}} \quad r_{k,\lambda} \;) \\
\ell > k & \mathbf{k}^*_{\ell,0} = (& a_{\ell,0} \quad 0 \quad 1\;) & \\
 & \mathbf{k}^*_{\ell,\lambda} = (& \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) \;\; a_{\ell,\lambda} & | \quad 0 \quad 0 \quad 0 \quad r_{\ell,\lambda} \;)
\end{array}
$$

$s'_{\ell,\lambda}$ is either the label $s_{\ell,\lambda}$ when $r_{\ell,\lambda} \cdot u_{t_{\ell,\lambda}} = 0$, or a random scalar in $\mathbb{Z}_q$ otherwise

$\mathbf{G}_{2.k.2.1}$ SubSpace-Ind Property, on $(\mathbb{D},\mathbb{D}^*)_{4,5}$, between τ and 0

$$\mathbf{c}_t = (\;\sigma_t(1,t) \quad \omega \quad | \quad \boxed{0} \quad 0 \quad \tau z_t \quad u_t \;)$$

$\mathbf{G}_{2.k.2.2}$ SubSpace-Ind Property, on $(\mathbb{D}^*,\mathbb{D})_{2,4}$, between 0 and $y_{\ell,\lambda}$

$$
\begin{array}{rl}
\mathbf{k}^*_{k,0} = (& a_{k,0} \quad s_{k,0} \quad 1\;) \\
\mathbf{k}^*_{k,\lambda} = (& \pi_{k,\lambda}(t_{k,\lambda},-1) \;\; a_{k,\lambda} \quad | \quad \boxed{y_{k,\lambda}} \quad 0 \quad s_{k,\lambda}/z_{t_{k,\lambda}} \quad r_{k,\lambda} \;)
\end{array}
$$

$\mathbf{G}_{2.k.2.3}$ Formal basis change, on $(\mathbb{D},\mathbb{D}^*)_{5,7}$, to duplicate $r_{\ell,\lambda}$

$$
\begin{array}{lll}
\ell < k & \mathbf{k}^*_{\ell,0} = (\; a_{\ell,0} \;\; r_{\ell,0} \;\; 1 \;) & \mathbf{k}^*_{\ell,\lambda} = (\ldots \,|\; y_{\ell,\lambda} \;\; \boxed{r_{\ell,\lambda}} \;\; s'_{\ell,\lambda}/z_{t_{\ell,\lambda}} \;\; r_{\ell,\lambda} \;) \\
\ell = k & \mathbf{k}^*_{k,0} = (\; a_{k,0} \;\; s_{k,0} \;\; 1 \;) & \mathbf{k}^*_{k,\lambda} = (\ldots \,|\; y_{k,\lambda} \;\; \boxed{r_{k,\lambda}} \;\; s_{k,\lambda}/z_{t_{k,\lambda}} \;\; r_{k,\lambda} \;) \\
\ell > k & \mathbf{k}^*_{\ell,0} = (\; a_{\ell,0} \;\; 0 \;\; 1 \;) & \mathbf{k}^*_{\ell,\lambda} = (\ldots \,|\; y_{\ell,\lambda} \;\; \boxed{r_{\ell,\lambda}} \;\; 0 \;\; r_{\ell,\lambda} \;)
\end{array}
$$

$\mathbf{G}_{2.k.2.4}$ Alteration of the labeling. See Figure 14

$$
\begin{array}{lll}
\ell < k & \mathbf{k}^*_{\ell,0} = (\; a_{\ell,0} \;\; r_{\ell,0} \;\; 1 \;) & \mathbf{k}^*_{\ell,\lambda} = (\ldots \,|\; y_{\ell,\lambda} \;\; \boxed{0} \;\; s'_{\ell,\lambda}/z_{t_{\ell,\lambda}} \;\; r_{l,\lambda} \;) \\
\ell = k & \mathbf{k}^*_{k,0} = (\; a_{k,0} \;\; s_{k,0} \;\; 1 \;) & \mathbf{k}^*_{k,\lambda} = (\ldots \,|\; y_{k,\lambda} \;\; \boxed{0} \;\; \boxed{s'_{k,\lambda}/z_{t_{k,\lambda}}} \;\; r_{k,\lambda} \;) \\
\ell > k & \mathbf{k}^*_{\ell,0} = (\; a_{\ell,0} \;\; 0 \;\; 1 \;) & \mathbf{k}^*_{\ell,\lambda} = (\ldots \,|\; y_{\ell,\lambda} \;\; \boxed{0} \;\; 0 \;\; r_{\ell,\lambda} \;)
\end{array}
$$

$\mathbf{G}_{2.k.2.5}$ Limitations on KeyGen-queries: $s_{k,0}$ unpredictable, replaced by a random $r_{k,0}$

$$
\begin{array}{lll}
\ell = k & \mathbf{k}^*_{k,0} = (\; a_{k,0} \;\; \boxed{r_{k,0}} \;\; 1 \;) & \mathbf{k}^*_{k,\lambda} = (\ldots \,|\; y_{k,\lambda} \;\; 0 \;\; s'_{k,\lambda}/z_{t_{k,\lambda}} \;\; r_{k,\lambda} \;)
\end{array}
$$

$\mathbf{G}_{2.k.2.6}$ SubSpace-Ind Property, on $(\mathbb{D}^*,\mathbb{D})_{2,4}$, between $y_{\ell,\lambda}$ and 0

$$
\begin{array}{rl}
\mathbf{k}^*_{k,0} = (& a_{k,0} \quad r_{k,0} \quad 1\;) \\
\mathbf{k}^*_{k,\lambda} = (& \pi_{k,\lambda}(t_{k,\lambda},-1) \;\; a_{k,\lambda} \quad | \quad \boxed{0} \quad 0 \quad s'_{k,\lambda}/z_{t_{k,\lambda}} \quad r_{k,\lambda} \;)
\end{array}
$$

$\mathbf{G}_{2.k.2.7}$ SubSpace-Ind Property, on $(\mathbb{D},\mathbb{D}^*)_{4,5}$, between 0 and τ

$$\mathbf{c}_t = (\;\sigma_t(1,t) \quad \omega \quad | \quad \boxed{\tau} \quad 0 \quad \tau z_t \quad u_t \;)$$

Fig. 8: Sequence of games on the keys for the Del-IND-security proof of our SA-KP-ABE

games is in the same vein as for the previous proofs, except the $\mathsf{RoREncaps}$-challenge is instead a $\mathsf{RoAVEncaps}$-challenge, where we require all the policies in the queried keys to reject the challenge ciphertext even if all the ciphertexts are valid: $\tilde{\mathcal{T}}_\ell(\Gamma_v \cup \Gamma_i, 0) = 0$. In the sequence, we again introduce an additional labeling $(s_{\ell,0}, (s_{\ell,\lambda})_\lambda)$ in the hidden components of each ℓ-th key, where each label is masked by a random z_t for each attribute t. Then, in the real case, $b = 0$, we will cancel the u_t values in the ciphertext, but this will move the $r_{\ell,\lambda}$ values in the label-column of the keys and merge them with $s_{\ell,\lambda}$. As a consequence, when $r_{\ell,\lambda} \neq 0$, the labels will be altered. This could make a difference between the real and all-valid cases: in the latter, labels all correspond to a correct labeling. But under the additional assumption that active keys correspond to independent leaves, there is no actual difference for the adversary: the real and all-valid ciphertexts are perfectly the same.

6 Application to Traitor-Tracing

Our initial motivation was to adapt $\mathsf{KP\text{-}ABE}$ with delegation to support tracing, which should not be detectable by the pirate decoder. We now explain how our $\mathsf{SA\text{-}KP\text{-}ABE}$ primitive allows that. We first recall the definitions of tracing, and then we illustrate with a possible family of policies with switchable leaves and attributes.

6.1 Definition

We thus add a Tracing algorithm to initial definition of delegatable $\mathsf{KP\text{-}ABE}$ from Section 3.2:

$\mathsf{Setup}(1^\kappa, n, t)$**.** From the security parameter κ, the total number n of users in the system, and the maximal size t on the collusion, the algorithm defines all the global parameters PK, the master secret key MK, and the tracing key TK;

$\mathsf{KeyGen}(\mathsf{MK}, \mathcal{T}, \mathsf{id})$**.** From a master key MK and an access tree $\mathcal{T}$, the algorithm outputs a key $\mathsf{dk}_{\mathsf{id},\mathcal{T}}$, specific to the user id;

$\mathsf{Delegate}(\mathsf{dk}_{\mathsf{id},\mathcal{T}}, \mathcal{T}', \mathsf{id}')$**.** Given a key $\mathsf{dk}_{\mathsf{id},\mathcal{T}}$ and a more restrictive access-tree $\mathcal{T}' \leq \mathcal{T}$, the algorithm outputs a decryption key $\mathsf{dk}_{\mathsf{id}',\mathcal{T}'}$;

$\mathsf{Encaps}(\mathsf{PK}, \Gamma)$**.** For a set Γ of attributes, the algorithm generates the ciphertext C and an encapsulated key K;

$\mathsf{Decaps}(\mathsf{dk}_{\mathsf{id},\mathcal{T}}, C)$**.** Given the key $\mathsf{dk}_{\mathsf{id},\mathcal{T}}$ and the ciphertext C, the algorithm outputs the encapsulated key K;

$\mathsf{Trace}^D(\mathsf{SK}, \Gamma)$**.** Given the secret key SK, and a black-box access to a Pirate Decoder D, the tracing algorithm outputs an index set I which identifies a set of malicious users, among the users id and id' compatible with Γ.

In the above definition, id' might be for a specific device of user id. Then the authority generates keys for users, and users delegate for devices, with any more restrictive policy $\mathcal{T}' \leq \mathcal{T}$: one can consider that $\mathsf{id}' = \mathsf{id} \| d$, for device d. One can then trace users and devices.

We expect two properties from the Trace algorithm on a perfect Pirate Decoder for a set Γ (that always decrypts the encapsulated key), when the number of traitors compatible with Γ is at most t: it always outputs a non-empty set of traitors, but does never wrongly accuse anybody.

Definition 15 (Traceability).

Initialize: *The challenger runs the* Setup *algorithm and gives the public parameters* PK *to the adversary;*

- $\mathsf{OKeyGen}(\mathsf{id}, \mathcal{T})$**:** *The adversary is allowed to issue* KeyGen*-queries for any access-tree* $\mathcal{T}$ *of its choice and gets back the secret key* $\mathsf{dk}_{\mathsf{id},\mathcal{T}}$ *;*

$\mathsf{ODelegate}(\mathsf{id}, \mathcal{T}, \mathsf{id}', \mathcal{T}')$**:** *The adversary is allowed to issue several* Delegate*-queries for any more restrictive access-tree $\mathcal{T}' \leq \mathcal{T}$ of its choice, for an already generated decryption key under $\mathcal{T}$, and gets back the decryption key $\mathsf{dk}_{\mathsf{id}',\mathcal{T}'}$;*

Finalize: *The adversary generates a set of attributes Γ and a perfect Decoder D on Γ, the challenger runs $\mathsf{Trace}^D(\mathsf{SK}, \Gamma)$ to get back I. Let us denote U_c (corrupted users) the set of id' for which $\mathcal{T}'$ has been asked such that $\mathcal{T}'(\Gamma) = 1$. If the size of U_c is at most t, but $I \not\subset U_c$ or I is empty, one outputs 1, otherwise one outputs 0.*

The success $\mathsf{Adv}^{\mathsf{trace}}(\mathcal{A})$ of an adversary $\mathcal{A}$ in this game is the probability to have 1 as output.

We stress that the above definition requires a perfect Pirate Decoder. This could be relaxed, but this is enough for our illustration.

6.2 Fingerprinting Code

Our technique will exploit traceable codes as in [CFN94] that allow to trace back codewords from words that have been derived from legitimate codewords. It uses the definition of feasible set, the list the words that can be derived from a set of words:

Definition 16 (Feasible Set). *Let $W = \{w^{(1)}, \dots, w^{(t)}\}$ be a set of t words in $\{0,1\}^\ell$. We say a word $w \in \{0,1\}^\ell$ is **feasible** for W if for all $i = 1, \dots, \ell$, there is a $j \in \{1, \dots, t\}$ such that $w_i = w_i^{(j)}$. The set of words feasible for W is the **feasible set** of W, denoted $F(W) = \{w \in \{0,1\}^\ell, \forall i, \exists w' \in W, w_i = w'_i\}$.*

A fingerprinting code is a particular traceable code. It defines a set of codewords that allows correct and efficient tracing to recover the traitor codewords from a word derived from them (in the feasible set). For the sake of clarity, we focus on binary codes:

Definition 17 (Fingerprinting Code). *A **fingerprinting code** is a pair of algorithms (G, T) defined as follows:*

Code generator *G is a probabilistic algorithm that takes a tuple (n, t) as input, where n is the number of codewords to output, and t is the maximal collusion size. The algorithm outputs a **code** Π of n codewords of bit-length ℓ.*

Tracing algorithm *T is a deterministic algorithm that takes as input a word $w^* \in \{0,1\}^\ell$ to trace. The algorithm T outputs a subset $S \subseteq \Pi$ of possible traitors.*

*Such a fingerprinting code is said t-**secure** if for all $n > t$ and all subsets $C \subseteq \{1, \dots, n\}$ of size at most t, when we set $\Pi = \{w^{(1)}, \dots, w^{(n)}\} \leftarrow G(n, t)$ and $W_C = \{w^{(i)}\}_{i \in C}$, for any word $w^* \in F(W_C)$, then $\emptyset \neq T(w^*) \subseteq C$.*

Again, we could relax the definition with error probabilities in identifying a traitor and in framing an honest user. Tardos codes [Tar03] are examples of short codes with probabilistic tracing capabilities and low error rates.

6.3 Traceable and Delegatable KP-ABE

We now explain how our SA-KP-ABE primitive is enough for tracing. For the sake of simplicity, in the following, we will keep $\mathsf{id}' = \mathsf{id}$, without specifying the device, still with any $\mathcal{T}' \leq \mathcal{T}$, but then devices of the same user cannot be traced. Only users can be traced, but various devices might have different policies:

$\mathsf{Setup}^{\mathsf{Tr}}(1^\kappa, n, t)$**.** The algorithm calls $\mathsf{Setup}(1^\kappa)$ and gets back $\mathsf{PK}, \mathsf{MK}, \mathsf{SK}$. It also calls code generator algorithm $G(n, t)$ to get the code Π. It sets the parameters as $\mathsf{PK}^{\mathsf{Tr}} = \mathsf{PK}$, $\mathsf{MK}^{\mathsf{Tr}} = (\mathsf{MK}, \Pi)$ and $\mathsf{TK}^{\mathsf{Tr}} = (\mathsf{SK}, T)$.

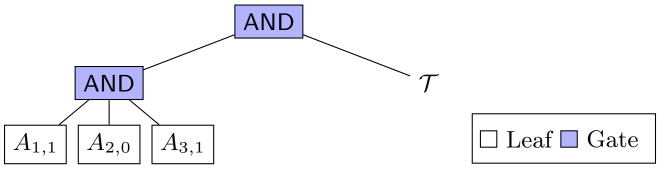

Fig. 9: BS95 codeword-based tree for the word $w = (1, 0, 1)$

$\mathsf{KeyGen}^{\mathsf{Tr}}(\mathsf{MK}^{\mathsf{Tr}}, \mathsf{id}, \mathcal{T})$. For an access-tree $\mathcal{T}$, the algorithm defines $\mathcal{T}^{\mathsf{Tr}}$, where $\mathcal{T}^{\mathsf{Tr}} = \mathcal{T} \wedge \mathcal{T}_{\mathsf{Tr}}$ are linked by an AND-gate at their root. The access-tree $\mathcal{T}_{\mathsf{Tr}}$ is constructed as follows (see Figure 9) :
- Choose a word $w_{\mathsf{id}} = w_{\mathsf{id},1} \dots w_{\mathsf{id},\ell}$ from Π, for any new id;
- Set $\mathcal{T}_{\mathsf{Tr}}$ as the AND of active leaves λ_i associated to the attributes $A_{i,w_{\mathsf{id},i}}$, for $i = 1, \dots, \ell$.

The algorithm then calls $\mathsf{KeyGen}(\mathsf{MK}, \tilde{\mathcal{T}}^{\mathsf{Tr}})$, where all leaves are passive in $\mathcal{T}$ and all leaves are active in $\mathcal{T}_{\mathsf{Tr}}$, and gets back $\mathsf{dk}_{\tilde{\mathcal{T}}^{\mathsf{Tr}}}$, and finally sets $\mathsf{dk}^{\mathsf{Tr}}_{\mathsf{id},\mathcal{T}} \leftarrow \mathsf{dk}_{\tilde{\mathcal{T}}^{\mathsf{Tr}}}$.

$\mathsf{Delegate}^{\mathsf{Tr}}(\mathsf{dk}^{\mathsf{Tr}}_{\mathsf{id},\mathcal{T}}, \mathcal{T}')$. Given a private key for an access-tree $\mathcal{T}$ and a more restrictive subtree $\mathcal{T}' \leq \mathcal{T}$, but for the same identity (as we focus on $\mathsf{id}' = \mathsf{id}$), the algorithm calls $\mathsf{Delegate}(\mathsf{dk}_{\tilde{\mathcal{T}}}, \tilde{\mathcal{T}}')$, where $\tilde{\mathcal{T}}$ and $\tilde{\mathcal{T}}'$ are $\mathcal{T}$ and $\mathcal{T}'$ combined with $\mathcal{T}_{\mathsf{Tr}}$ as above, to get a new delegated key $\mathsf{dk}_{\tilde{\mathcal{T}}'}$, and sets $\mathsf{dk}^{\mathsf{Tr}}_{\mathsf{id},\mathcal{T}'} = \mathsf{dk}_{\tilde{\mathcal{T}}'}$.

$\mathsf{Encaps}^{\mathsf{Tr}}(\mathsf{PK}^{\mathsf{Tr}}, \Gamma)$. For a set Γ of attributes, the algorithm defines $\Gamma^{\mathsf{Tr}} = \{A_{1,0}, A_{1,1}, \dots, A_{\ell,0}, A_{\ell,1}\}$. It then calls $\mathsf{Encaps}(\mathsf{PK}, \Gamma \cup \Gamma^{\mathsf{Tr}})$ and gets the output K and C. It then sets $K^{\mathsf{Tr}} = K$ and $C^{\mathsf{Tr}} = C$.

$\mathsf{Decaps}^{\mathsf{Tr}}(\mathsf{dk}^{\mathsf{Tr}}_{\mathsf{id},\mathcal{T}}, C)$. The algorithm calls $\mathsf{Decaps}^{\mathsf{Tr}}(\mathsf{dk}_{\tilde{\mathcal{T}}}, C)$, for $\mathsf{dk}_{\tilde{\mathcal{T}}} = \mathsf{dk}^{\mathsf{Tr}}_{\mathsf{id},\mathcal{T}}$, to get K, and outputs $K^{\mathsf{Tr}} = K$.

$\mathsf{Trace}^{\mathsf{Tr}}(\mathsf{TK}^{\mathsf{Tr}}, \Gamma)$. On input the tracing key $\mathsf{TK}^{\mathsf{Tr}} = (\mathsf{SK}, T)$, and access to a perfect Pirate Decoder D, the algorithm repeats the following experiment, for $j = 1, \dots, \ell$, to build the word w^*:
1. Set $\Gamma_v^{(0)} = \Gamma \cup \{A_{k,b}, k \neq j, b \in \{0, 1\}\} \cup \{A_{j,0}\}$ and $\Gamma_i^{(0)} = \{A_{j,1}\}$;
2. Set $\Gamma_v^{(1)} = \Gamma \cup \{A_{k,b}, k \neq j, b \in \{0, 1\}\} \cup \{A_{j,1}\}$ and $\Gamma_i^{(1)} = \{A_{j,0}\}$;
3. Compute the two challenges $(K_0, C_0) \leftarrow \mathsf{Encaps}^*(\mathsf{SK}, (\Gamma_v^{(0)}, \Gamma_i^{(0)}))$ and $(K_1, C_1) \leftarrow \mathsf{Encaps}^*(\mathsf{SK}, (\Gamma_v^{(1)}, \Gamma_i^{(1)}))$;
4. Ask for the decryption K_0' of C_0 and K_1' of C_1 to D;
5. If $K_0' = K_0$ then set $w_j^* \leftarrow 0$, else if $K_1' = K_1$ then set $w_j^* \leftarrow 1$. Otherwise, set $w_j^* \leftarrow 0$.

Eventually, the algorithm runs the tracing algorithm $T(w^*)$ to get S, the set of traitors, that it outputs.

6.4 Security Analysis

Again, we stress that we assume perfect Pirate Decoder D, but relaxed version would be possible. Hence, here, we know that D will successfully decrypt any normal ciphertext when Γ is acceptable for all the traitors. Then, during the tracing procedure, for any index j, there are three possibilities:

- If $w_j = 0$ for all keys in U_c, then the ciphertext C_0 is indistinguishable from the one where $A_{j,1}$ is in Γ_v because of the Attribute-Indistinguishability ($\mathsf{Att\text{-}IND}$) property of the scheme, hence D will always output K_0. We correctly set $w_j^* = 0$.
- If $w_j = 1$ for all keys in U_c, the same argument leads to $w_j^* = 1$, because the Pirate Decoder will output correct K_1', while K_0 will be unpredictable because of the Delegation-Indistinguishability for Encaps^*.
- If w_j has mixed values in 0 and 1 among users in U_c, D can detect that C_0 and C_1 involve active keys. But we could anyway set $w_j^* \leftarrow 0$ or $w_j^* \leftarrow 1$.

This way, the built word w^* satisfies that, for each position j, $w_j^* = w_j$, for some w in U_c: $w^* \in F(U_c)$. If the fingerprinting code is t-secure, since the size of U_c is at most t, $\emptyset \neq T(w^*) \subseteq U_c$. As a consequence, under the Attribute-Indistinguishability and the Delegation-Indistinguishability for Encaps^*, the delegatable KP-ABE is traceable.

6.5 Additional discussion

Our tracing system is presented with basic fingerprinting notions, for the sake of clarity, but more advanced features are possible. In particular, our tracing algorithm works as well with non-perfect Pirate Decoder, at the cost of more calls to D to increase the quality of the estimation. It is also compatible with [BN08], to drastically reduce the ciphertext size. Eventually, one could also let the user to delegate traceable keys to each devices. However, as we do not allow public traceability, only the tracing authority can run the tracing procedure, to trace users or devices.

7 Conclusion

We have designed a KP-ABE scheme that allows an authority to generate keys with specific policies for each users, so that these users can thereafter delegate their keys to their own devices with more restrictive rights. The restrictions might include time-based, identity-based or group-based limitations with more restrictive access-trees. Thanks to the Attribute-Indistinguishability, it can also include key material for tracing a compromised user/device. The strong notion of indistinguishability even requires just a low collusion-resistance level, as all the revoked keys, by any means independent of the leaves involved in the fingerprinting, cannot help to detect and defeat the tracing procedure.

In addition, with Key-Indistinguishability on active leaves and perfect randomization on passive leaves, one achieves anonymity of the devices: one cannot detect whether two keys have been delegated by the same user.

References

AT20. N. Attrapadung and J. Tomida. Unbounded dynamic predicate compositions in ABE from standard assumptions. In *ASIACRYPT 2020, Part III*, *LNCS* 12493, pages 405–436. Springer, Heidelberg, December 2020.

BN08. D. Boneh and M. Naor. Traitor tracing with constant size ciphertext. In *ACM CCS 2008*, pages 501–510. ACM Press, October 2008.

BS95. D. Boneh and J. Shaw. Collusion-secure fingerprinting for digital data (extended abstract). In *CRYPTO'95*, *LNCS* 963, pages 452–465. Springer, Heidelberg, August 1995.

BSW06. D. Boneh, A. Sahai, and B. Waters. Fully collusion resistant traitor tracing with short ciphertexts and private keys. In *EUROCRYPT 2006*, *LNCS* 4004, pages 573–592. Springer, Heidelberg, May / June 2006.

BW06. D. Boneh and B. Waters. A fully collusion resistant broadcast, trace, and revoke system. In *ACM CCS 2006*, pages 211–220. ACM Press, October / November 2006.

CFN94. B. Chor, A. Fiat, and M. Naor. Tracing traitors. In *CRYPTO'94*, *LNCS* 839, pages 257–270. Springer, Heidelberg, August 1994.

CGW15. J. Chen, R. Gay, and H. Wee. Improved dual system ABE in prime-order groups via predicate encodings. In *EUROCRYPT 2015, Part II*, *LNCS* 9057, pages 595–624. Springer, Heidelberg, April 2015.

CGW18. J. Chen, J. Gong, and H. Wee. Improved inner-product encryption with adaptive security and full attribute-hiding. In *ASIACRYPT 2018, Part II*, *LNCS* 11273, pages 673–702. Springer, Heidelberg, December 2018.

CLL+13. J. Chen, H. W. Lim, S. Ling, H. Wang, and H. Wee. Shorter IBE and signatures via asymmetric pairings. In *PAIRING 2012*, *LNCS* 7708, pages 122–140. Springer, Heidelberg, May 2013.

CPP17. J. Chotard, D. H. Phan, and D. Pointcheval. Homomorphic-policy attribute-based key encapsulation mechanisms. In *ISC 2017*, *LNCS* 10599, pages 155–172. Springer, Heidelberg, November 2017.

GPSW06. V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM CCS 2006*, pages 89–98. ACM Press, October / November 2006. Available as Cryptology ePrint Archive Report 2006/309.

KSW08. J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *EUROCRYPT 2008*, *LNCS* 4965, pages 146–162. Springer, Heidelberg, April 2008.

LLLW17. X. Li, K. Liang, Z. Liu, and D. S. Wong. Attribute based encryption: Traitor tracing, revocation and fully security on prime order groups. In *CLOSER 2017*, pages 281–292. SciTePress, 2017.

LOS+10. A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In *EUROCRYPT 2010*, *LNCS* 6110, pages 62–91. Springer, Heidelberg, May / June 2010.

LT18. J. Lai and Q. Tang. Making any attribute-based encryption accountable, efficiently. In *ESORICS 2018, Part II*, *LNCS* 11099, pages 527–547. Springer, Heidelberg, September 2018.

LW10. A. B. Lewko and B. Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In *TCC 2010*, *LNCS* 5978, pages 455–479. Springer, Heidelberg, February 2010.

LW15. Z. Liu and D. S. Wong. Practical ciphertext-policy attribute-based encryption: Traitor tracing, revocation, and large universe. In *ACNS 15*, *LNCS* 9092, pages 127–146. Springer, Heidelberg, June 2015.

OT08. T. Okamoto and K. Takashima. Homomorphic encryption and signatures from vector decomposition. In *PAIRING 2008*, *LNCS* 5209, pages 57–74. Springer, Heidelberg, September 2008.

OT10. T. Okamoto and K. Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In *CRYPTO 2010*, *LNCS* 6223, pages 191–208. Springer, Heidelberg, August 2010.

OT12a. T. Okamoto and K. Takashima. Adaptively attribute-hiding (hierarchical) inner product encryption. In *EUROCRYPT 2012*, *LNCS* 7237, pages 591–608. Springer, Heidelberg, April 2012.

OT12b. T. Okamoto and K. Takashima. Fully secure unbounded inner-product and attribute-based encryption. In *ASIACRYPT 2012*, *LNCS* 7658, pages 349–366. Springer, Heidelberg, December 2012.

Tar03. G. Tardos. Optimal probabilistic fingerprint codes. In *35th ACM STOC*, pages 116–125. ACM Press, June 2003.

Wat09. B. Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In *CRYPTO 2009*, *LNCS* 5677, pages 619–636. Springer, Heidelberg, August 2009.

Appendix

A Dual Pairing Vector Spaces

In this section, we provide a brief review of the Dual Pairing Vector Spaces (DPVS), that have been proposed for efficient constructions with adaptive security [OT08,LOS+10,OT10,OT12b], as Dual Systems [Wat09], in prime-order groups under the DLIN assumption. In [LW10], Dual Systems were using pairing on composite order elliptic curves. Then, prime-order groups have been used with the SXDH assumption, in a pairing-friendly setting of primer order, which means that the DDH assumptions hold in both $\mathbb{G}_1$ and $\mathbb{G}_2$ [CLL+13]. In all theses situations, one exploited indistinguishability of sub-groups or sub-spaces. In this section, for the sake of efficiency, we use the SXDH assumption in a pairing-friendly setting $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, e, G_1, G_2, q)$ of primer order q.

A.1 Pairing Vector Spaces

Let us be given any cyclic group $(\mathbb{G} = \langle G \rangle, +)$ of prime order q, denoted additively. We can define the $\mathbb{Z}_q$ vector space of dimension n,

$$\mathbb{G}^n = \{X = \vec{x} \cdot G \stackrel{\text{def}}{=} (X_1 = x_1 \cdot G, \dots, X_n = x_n \cdot G) \,|\, \vec{x} \in \mathbb{Z}_q^n\},$$

with the following laws:

$$(X_1,\dots,X_n)+(Y_1,\dots,Y_n) \stackrel{\mathsf{def}}{=} (X_1+Y_1,\dots,X_n+Y_n)$$
$$a\cdot(X_1,\dots,X_n) \stackrel{\mathsf{def}}{=} (a\cdot X_1,\dots,a\cdot X_n)$$

Essentially, all the operations between the vectors in $\mathbb{G}^n$ are applied on the vectors in $\mathbb{Z}_q^n$:

$$\vec{x}\cdot G+\vec{y}\cdot G \stackrel{\mathsf{def}}{=} (\vec{x}+\vec{y})\cdot G \qquad\qquad a\cdot(\vec{x}\cdot G) \stackrel{\mathsf{def}}{=} (a\cdot\vec{x})\cdot G$$

where $\vec{x}+\vec{y}$ and $a\cdot\vec{x}$ are the usual internal and external laws of the vector space $\mathbb{Z}_q^n$. For the sake of clarity, vectors will be row-vectors.

If we are using a pairing-friendly setting $(\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_t,e,G_1,G_2,q)$, with a bilinear map e from $\mathbb{G}_1\times\mathbb{G}_2$ into $\mathbb{G}_t$, we can have an additional law between an element $X\in\mathbb{G}_1^n$ and $Y\in\mathbb{G}_2^n$: $X\times Y \stackrel{\mathsf{def}}{=} \prod_i e(X_i,Y_i)$, where $\mathbb{G}_t$ is usually denoted multiplicatively.

Note that if $X=(X_1,\dots,X_n)=\vec{x}\cdot G_1\in\mathbb{G}_1^n$ and $Y=(Y_1,\dots,Y_n)=\vec{y}\cdot G_2\in\mathbb{G}_2^n$:

$$(\vec{x}\cdot G_1)\times(\vec{y}\cdot G_2)=X\times Y=\prod_i e(X_i,Y_i)=\prod_i e(x_i\cdot G_1,y_i\cdot G_2)$$
$$=\prod_i g_t^{x_i\cdot y_i}=g_t^{\vec{x}\cdot\vec{y}^\top}=g_t^{\langle\vec{x},\vec{y}\rangle}$$

where $g_t=e(G_1,G_2)$ and $\langle\vec{x},\vec{y}\rangle$ is the inner product between vectors $\vec{x}$ and $\vec{y}$.

A.2 Dual Pairing Vector Spaces

We define $\mathcal{E}=(\vec{e}_i)_i$ the canonical basis of $\mathbb{Z}_q^n$, where $\vec{e}_i=(\delta_{i,1},\dots,\delta_{i,n})$, with the classical $\delta_{i,j}=1$ if $i=j$ and $\delta_{i,j}=0$ otherwise, for $i,j\in\{1,\dots,n\}$. We can also define $\mathbb{E}=(\mathbf{e}_i)_i$ the canonical basis of $\mathbb{G}^n$, where $\mathbf{e}_i=\vec{e}_i\cdot G=(\delta_{i,j}\cdot G)_j$. More generally, given any basis $\mathcal{B}=(\vec{b}_i)_i$ of $\mathbb{Z}_q^n$, we can define the basis $\mathbb{B}=(\mathbf{b}_i)_i$ of $\mathbb{G}^n$, where $\mathbf{b}_i=\vec{b}_i\cdot G$.

Choosing a random basis $\mathbb{B}$ of $\mathbb{G}^n$ is equivalent to a random choice of an invertible matrix $B\stackrel{\$}{\leftarrow}\mathrm{GL}_n(\mathbb{Z}_q)$, the definition $\mathcal{B}\leftarrow B\times\mathcal{E}$, where $\mathcal{B}=(\vec{b}_i)_i$ is a basis of $\mathbb{Z}_q^n$ (B is essentially the matrix with $\vec{b}_i$ as its i-th row, as $\vec{b}_i=\sum_j B_{i,j}\cdot\vec{e}_j$), and then $\mathbb{B}\leftarrow(\mathbf{b}_i)_i$ where $\mathbf{b}_i=\vec{b}_i\cdot G$: $\mathbb{B}$ is the basis of $\mathbb{G}^n$ associated to the matrix B as

$$\mathbf{b}_i=\vec{b}_i\cdot G=\sum_j B_{i,j}\cdot\vec{e}_j\cdot G=\sum_j B_{i,j}\cdot\mathbf{e}_j : \mathbb{B}=B\cdot\mathbb{E}.$$

In case of pairing-friendly setting, for a dimension n, we will denote $\mathbb{E}=(\mathbf{e}_i)_i$ and $\mathbb{E}^*=(\mathbf{e}_i^*)_i$ the canonical bases of $\mathbb{G}_1^n$ and $\mathbb{G}_2^n$, respectively:

$$\mathbf{e}_i\times\mathbf{e}_j^*=(\vec{e}_i\cdot G_1)\times(\vec{e}_j\cdot G_2)=g_T^{\langle\vec{e}_i,\vec{e}_j\rangle}=g_T^{\delta_{i,j}}.$$

The same way, if we denote $\mathbb{B}=(\mathbf{b}_i)_i=B\cdot\mathbb{E}$ the basis of $\mathbb{G}_1^n$ associated to a matrix B, and $\mathbb{B}^*=(\mathbf{b}_i^*)_i=B'\cdot\mathbb{E}^*$ the basis of $\mathbb{G}_2^n$ associated to the matrix $B'=(B^{-1})^\top$, as $B\cdot B'^\top=I_n$,

$$\mathbf{b}_i\times\mathbf{b}_j^*=(\vec{b}_i\cdot G_1)\times(\vec{b}'_j\cdot G_2)=g_t^{\langle\vec{b}_i,\vec{b}'_j\rangle}=g_t^{\delta_{i,j}}.$$

$\mathbb{B}$ and $\mathbb{B}^*$ are called *Dual Orthogonal Bases.*

We have seen above the canonical bases $\mathbb{E}$ and $\mathbb{E}^*$ are dual orthogonal bases, but for any random invertible matrix $U\stackrel{\$}{\leftarrow}\mathrm{GL}_n(\mathbb{Z}_q)$, the bases $\mathbb{U}$ of $\mathbb{G}_1^n$ associated to the matrix U and $\mathbb{U}^*$ of $\mathbb{G}_2^n$ associated to the matrix $(U^{-1})^\top$ are *Random Dual Orthogonal Bases.*

A pairing-friendly setting $(\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_t,e,G_1,G_2,q)$, with such dual orthogonal bases $\mathbb{U}$ and $\mathbb{U}^*$ of size n, is called a *Dual Pairing Vector Space* (DPVS).

A.3 Change of Basis

Let us consider the basis $\mathbb{U} = (\mathbf{u}_i)_i$ of $\mathbb{G}^n$ associated to a matrix $U \in \mathrm{GL}_n(\mathbb{Z}_q)$, and the basis $\mathbb{B} = (\mathbf{b}_i)_i$ of $\mathbb{G}^n$ associated to the product matrix BU, for any $B \in \mathrm{GL}_n(\mathbb{Z}_q)$, a vector $\vec{x}$ in $\mathbb{B}$, denoted $(\vec{x})_\mathbb{B}$ means

$$\begin{aligned}(\vec{x})_\mathbb{B} &= \sum_i x_i \cdot \mathbf{b}_i = \sum_i x_i \cdot \vec{b}_i \cdot G = \vec{x} \cdot BU \cdot G = (\vec{x} \cdot B) \cdot U \cdot G = \vec{y} \cdot U \cdot G \\ &= \sum_i y_i \cdot \vec{u}_i \cdot G = \sum_i y_i \cdot \mathbf{u}_i = (\vec{y})_\mathbb{U} \text{ where } \vec{y} = \vec{x} \cdot B.\end{aligned}$$

Hence, $(\vec{x})_\mathbb{B} = (\vec{x}\cdot B)_\mathbb{U}$ and $(\vec{x}\cdot B^{-1})_\mathbb{B} = (\vec{x})_\mathbb{U}$ where we denote $\mathbb{B} \stackrel{\mathsf{def}}{=} B\cdot\mathbb{U}$. For any invertible matrix B, if $\mathbb{U}$ is a random basis, then $\mathbb{B} = B\cdot\mathbb{U}$ is also a random basis. Then, with $B^{-1} = (\vec{b'}_1^\top, \ldots, \vec{b'}_n^\top)$, $\vec{x} = \vec{y} \cdot (\vec{b'}_1^\top, \ldots, \vec{b'}_n^\top)$:

$$\mathbb{B} = B \cdot \mathbb{U}, B' = \begin{pmatrix} \vec{b'}_1 \\ \vdots \\ \vec{b'}_n \end{pmatrix}, \text{ and } (\vec{x})_\mathbb{B} = (\vec{y})_\mathbb{U} \Longrightarrow \vec{x} = (\langle \vec{y}, \vec{b'}_1\rangle, \ldots, \langle \vec{y}, \vec{b'}_n\rangle).$$

Let us consider the random dual orthogonal bases $\mathbb{U} = (\mathbf{u}_i)_i$ and $\mathbb{U}^* = (\mathbf{u}_i^*)_i$ of $\mathbb{G}_1^n$ and $\mathbb{G}_2^n$ respectively associated to a matrix U (which means that $\mathbb{U}$ is associated to the matrix U and $\mathbb{U}^*$ is associated to the matrix $(U^{-1})^\top$): the bases $\mathbb{B} = B \cdot \mathbb{U}$ and $\mathbb{B}' = (B^{-1})^\top \cdot \mathbb{U}^*$ are also dual orthogonal bases:

$$\mathbf{b}_i \times \mathbf{b}_j^* = g_t^{\vec{b}_i \cdot \vec{b'}_j^\top} = g_t^{\vec{u}_i \cdot B \cdot (B^{-1})^\top \cdot \vec{u}_j^{*\top}} = g_t^{\vec{u}_i \cdot \vec{u}_j^{*\top}} = g_t^{\delta_{i,j}}.$$

A.4 Partial Change of Basis

We will often just have to partially change a basis, on a few vectors only: the transition matrix

$$B = (t)_{i_1,\ldots,i_m} = \begin{pmatrix} t_{1,1} & \ldots & t_{1,m} \\ \vdots & & \vdots \\ t_{m,1} & \ldots & t_{m,m} \end{pmatrix}_{i_1,\ldots,i_m}$$

means the $n \times n$ matrix B where

$$B_{i,j} = \delta_{i,j}, \text{ if any } i,j \notin \{i_1,\ldots,i_m\} \qquad B_{i_k,i_\ell} = t_{k,\ell}, \text{ for all } k,\ell \in \{1,\ldots,m\}$$

As a consequence, from a basis $\mathbb{U}$, $\mathbb{B} = B \cdot \mathbb{U}$ corresponds to the basis

$$\mathbf{b}_i = \mathbf{u}_i, \text{ if } i \notin \{i_1,\ldots,i_m\} \qquad \mathbf{b}_{i_k} = \sum_\ell t_{k,\ell} \cdot \mathbf{u}_{i_\ell}, \text{ if } k \notin \{i_1,\ldots,i_m\}$$

As we need to have $\mathbb{B}^* = (B^{-1})^\top \cdot \mathbb{U}^*$, we need the dual transition matrix B' to be $B' = (t')_{i_1,\ldots,i_m}$ where $t' = (t^{-1})^\top$. Indeed, in such a case, we have

$$\mathbf{b}_i^* = \mathbf{u}_i^*, \text{ if } i \notin \{i_1,\ldots,i_m\} \qquad \mathbf{b}_{i_k}^* = \sum_\ell t'_{k,\ell} \cdot \mathbf{u}_{i_\ell}^*, \text{ if } k \notin \{i_1,\ldots,i_m\}$$

so,

- if both $i,j \notin \{i_1,\ldots,i_m\}$, $\mathbf{b}_i \times \mathbf{b}_j^* = \mathbf{u}_i \times \mathbf{u}_j^* = g_t^{\delta_{i,j}}$;

– if $i = i_k \in \{i_1, \ldots, i_m\}$, but $j \notin \{i_1, \ldots, i_m\}$,

$$\mathbf{b}_i \times \mathbf{b}^*_j = \mathbf{b}_{i_k} \times \mathbf{u}^*_j = \left(\sum_\ell t_{k,\ell} \cdot \mathbf{u}_{i_\ell}\right) \times \mathbf{u}^*_j = \prod_\ell (\mathbf{u}_{i_\ell} \times \mathbf{u}^*_j)^{t_{k,\ell}} = 1$$

– if $i \notin \{i_1, \ldots, i_m\}$, but $j = i_k \in \{i_1, \ldots, i_m\}$,

$$\mathbf{b}_i \times \mathbf{b}^*_j = \mathbf{u}_i \times \mathbf{b}^*_{i_k} = \mathbf{u}_i \times \left(\sum_\ell t'_{k,\ell} \cdot \mathbf{u}^*_{i_\ell}\right) = \prod_\ell (\mathbf{u}_i \times \mathbf{u}^*_{i_\ell})^{t'_{k,\ell}} = 1$$

– if $i = i_k$ and $j = i_\ell$,

$$\begin{aligned}\mathbf{b}_i \times \mathbf{b}^*_j &= \left(\sum_p t_{k,p} \cdot \mathbf{u}_{i_p}\right) \times \left(\sum_p t'_{\ell,p} \cdot \mathbf{u}^*_{i_p}\right)\\ &= \prod_p (\mathbf{u}_{i_p} \times \mathbf{u}^*_{i_p})^{t_{k,p} \cdot t'_{\ell,p}} = g_t^{\sum_p t_{k,p} \cdot t'_{\ell,p}} = g_t^{\sum_p t_{k,p} \cdot {t'}^\top_{p,\ell}} = g_t^{\delta_{k,\ell}} = g_t^{\delta_{i,j}}\end{aligned}$$

A.5 Particular Changes

Let us consider a triple $(a \cdot G_1, b \cdot G_1, c \cdot G_1)$, that is either a Diffie-Hellman tuple (*i.e.*, $c = ab \bmod q$) or a random tuple (*i.e.*, $c = ab + \tau \bmod q$, for $\tau \xleftarrow{\$} \mathbb{Z}^*_q$). For any random dual orthogonal bases $\mathbb{U}$ and $\mathbb{U}^*$ associated to the matrices U and $U' = (U^{-1})^\top$, respectively, we can set

$$B = \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}_{1,2} \qquad B' = \begin{pmatrix} 1 & 0 \\ -a & 1 \end{pmatrix}_{1,2} \qquad \mathbb{B} = B \cdot \mathbb{U} \qquad \mathbb{B}^* = B' \cdot \mathbb{U}^*$$

Note that we can compute $\mathbb{B} = (\mathbf{b}_i)_i$, as we know $a \cdot G$ and all the scalars in U:

$$\mathbf{b}_i = \sum_k B_{i,k} \cdot \mathbf{u}_k \qquad \mathbf{b}_{i,j} = \sum_k B_{i,k} \cdot \mathbf{u}_{k,j} = \sum_k B_{i,k} U_{k,j} \cdot G_1 = \sum_k U_{k,j} \cdot (B_{i,j} \cdot G_1).$$

Hence, to compute $\mathbf{b}_i$, one needs all the scalars in U, but only the group elements $B_{i,j} \cdot G_1$, and so G_1 and $a \cdot G_1$. This is the same for $\mathbb{B}^*$, excepted for the vector $\mathbf{b}^*_2$ as $a \cdot G_2$ is missing. One can thus publish $\mathbb{B}$ and $\mathbb{B}^* \backslash \{\mathbf{b}^*_2\}$.

Indistinguishability of Sub-Spaces. As already remarked, for such a fixed matrix B, if $\mathbb{U}$ is random, so is $\mathbb{B}$ too, and $(\vec{x})_\mathbb{B} = (\vec{x} \cdot B)_\mathbb{U}$, so $(\vec{x})_\mathbb{U} = (\vec{x} \cdot B^{-1})_\mathbb{B}$. Note that $B^{-1} = {B'}^\top$. So, in particular

$$\begin{aligned}(b, c, 0, \ldots, 0)_\mathbb{U} &+ (x_1, x_2, x_3, \ldots, x_n)_\mathbb{B}\\ &= (b, c - ab, 0, \ldots, 0)_\mathbb{B} + (x_1, x_2, x_3, \ldots, x_n)_\mathbb{B}\\ &= (x_1 + b, x_2 + \tau, x_3, \ldots, x_n)_\mathbb{B}\end{aligned}$$

where τ can be either 0 or random.

Note that whereas we cannot compute $\mathbf{b}^*_2$, this does not exclude this second component in the computed vectors: $(\vec{y})_{\mathbb{U}^*} = (\vec{y} \cdot {B'}^{-1})_{\mathbb{B}^*} = (\vec{y} \cdot B^\top)_{\mathbb{B}^*}$. So, in particular

$$(y_1, \ldots, y_n)_{\mathbb{U}^*} = (y_1 + ay_2, y_2, \ldots, y_n)_{\mathbb{B}^*}.$$

Theorem 18. *Under the DDH Assumption in $\mathbb{G}_1$, for random dual orthogonal bases $\mathbb{B}$ and $\mathbb{B}^*$, once having seen $\mathbb{B}$ and $\mathbb{B}^* \backslash \{\mathbf{b}^*_2\}$, and any vector $(y_1, y_2, \ldots, y_n)_{\mathbb{B}^*}$, for chosen $y_2, \ldots, y_n \in \mathbb{Z}_q$, but unknown random $y_1 \xleftarrow{\$} \mathbb{Z}_q$, one cannot distinguish the vectors $(x_1, x'_2, x_3, \ldots, x_n)_\mathbb{B}$ and $(x_1, x_2, x_3, \ldots, x_n)_\mathbb{B}$, for chosen $x_2, \ldots, x_n \in \mathbb{Z}_q$, but unknown random $x_1, x'_2 \xleftarrow{\$} \mathbb{Z}_q$.*

Using the DSDH assumption instead of the DDH assumption, on two chosen values x_2 and x'_2, one can show that no algorithm can efficiently distinguish $(x_1, x_2, x_3, \ldots, x_n)_{\mathbb{B}}$ from $(x_1, x'_2, x_3, \ldots, x_n)_{\mathbb{B}}$, for chosen $x'_2, x_2, \ldots, x_n \in \mathbb{Z}_q$, but unknown random $x_1 \stackrel{\$}{\leftarrow} \mathbb{Z}_q$:

Theorem 19 (SubSpace-Ind Property). *Under the DSDH Assumption in $\mathbb{G}_1$, for random dual orthogonal bases $\mathbb{B}$ and $\mathbb{B}^*$, once having seen $\mathbb{B}$ and $\mathbb{B}^* \backslash \{\mathbf{b}_2^*\}$, and any vector $(y_1, y_2, \ldots, y_n)_{\mathbb{B}^*}$, for chosen $y_2, \ldots, y_n \in \mathbb{Z}_q$, but unknown random $y_1 \stackrel{\$}{\leftarrow} \mathbb{Z}_q$, one cannot distinguish the vectors $(x_1, x'_2, x_3, \ldots, x_n)_{\mathbb{B}}$ and $(x_1, x_2, x_3, \ldots, x_n)_{\mathbb{B}}$, for chosen $x'_2, x_2, \ldots, x_n \in \mathbb{Z}_q$, but unknown random $x_1 \stackrel{\$}{\leftarrow} \mathbb{Z}_q$.*

We stress that for this property, we only work with $(\mathbf{b}_1, \mathbf{b}_2)$ and $(\mathbf{b}_1^*, \mathbf{b}_2^*)$, but without publishing $\mathbf{b}_2^*$.

Indistinguishability of Position. Let us consider another change of basis:

$$B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ a & -a & 1 \end{pmatrix}_{1,2,3} \qquad B' = \begin{pmatrix} 1 & 0 & -a \\ 0 & 1 & a \\ 0 & 0 & 1 \end{pmatrix}_{1,2,3} \qquad \mathbb{B} = B \cdot \mathbb{U} \qquad \mathbb{B}^* = B' \cdot \mathbb{U}^*$$

In this case, we can compute $\mathbb{B} = (\mathbf{b}_i)_i$, but not the vectors $\mathbf{b}_1^*$ and $\mathbf{b}_2^*$ as $a \cdot G_2$ is missing.

$$\begin{aligned} (c, -c, b, x_4, \ldots, x_n)_{\mathbb{U}} &= (c - ab, -c + ab, b, x_4, \ldots, x_n)_{\mathbb{B}} = (\tau, -\tau, b, x_4, \ldots, x_n)_{\mathbb{B}} \\ (\theta, \theta, y_3, y_4, \ldots, y_n)_{\mathbb{U}^*} &= (\theta, \theta, a\theta - a\theta + y_3, y_4, \ldots, y_n)_{\mathbb{B}^*} = (\theta, \theta, y_3, \ldots, y_n)_{\mathbb{B}^*} \end{aligned}$$

There is the limitation for the first two components in $\mathbb{B}^*$ to be the same:

Theorem 20 (Pos-Ind Property). *Under the DDH Assumption in $\mathbb{G}_1$, for random dual orthogonal bases $\mathbb{B}$ and $\mathbb{B}^*$, once having seen $\mathbb{B}$ and $\mathbb{B}^* \backslash \{\mathbf{b}_1^*, \mathbf{b}_2^*\}$ and $(y_1, y_1, y_3, \ldots, y_n)_{\mathbb{B}^*}$, for chosen $y_1, y_3, \ldots, y_n \in \mathbb{Z}_q$, one cannot distinguish the vectors $(x_1, -x_1, x_3, x_4, \ldots, x_n)_{\mathbb{B}}$ and $(0, 0, x_3, x_4, \ldots, x_n)_{\mathbb{B}}$, for chosen $x_4, \ldots, x_n \in \mathbb{Z}_q$, but unknown random $x_1, x_3 \stackrel{\$}{\leftarrow} \mathbb{Z}_q$.*

We stress again that for this property, we only work with $(\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)$ and $(\mathbf{b}_1^*, \mathbf{b}_2^*, \mathbf{b}_3^*)$, but without publishing $(\mathbf{b}_1^*, \mathbf{b}_2^*)$.

But more useful, using the DSDH assumption on 0 and x_1, which claims indistinguishability between $(a \cdot G, b \cdot G, (ab + 0) \cdot G)$ and $(a \cdot G, b \cdot G, (ab + x_1) \cdot G)$, we have indistinguishability between

$$\begin{aligned} &(0, x_1, x_3, \ldots, x_n)_{\mathbb{B}} + (ab, -ab, b, 0, \ldots, 0)_{\mathbb{U}} \\ &\qquad = (0, x_1, x_3, \ldots, x_n)_{\mathbb{B}} + (ab - ab, -ab + ab, b, 0, \ldots, 0)_{\mathbb{B}} \\ &\qquad = (0, x_1, x_3, \ldots, x_n)_{\mathbb{B}} \\ &(0, x_1, x_3, \ldots, x_n)_{\mathbb{B}} + (ab + x_1, -ab - x_1, b, 0, \ldots, 0)_{\mathbb{U}} \\ &\qquad = (0, x_1, x_3, \ldots, x_n)_{\mathbb{B}} + (ab + x_1 - ab, -ab - x_1 + ab, b, 0, \ldots, 0)_{\mathbb{B}} \\ &\qquad = (x_1, 0, x_3, \ldots, x_n)_{\mathbb{B}} \\ &(y_1, y_1, y_3, y_4, \ldots, y_n)_{\mathbb{U}^*} = (y_1, y_1, ay_1 - ay_1 + y_3, y_4, \ldots, y_n)_{\mathbb{B}^*} \\ &\qquad = (y_1, y_1, y_3, \ldots, y_n)_{\mathbb{B}^*} \end{aligned}$$

Hence,

Theorem 21 (Swap-Ind Property). *Under the DSDH Assumption in $\mathbb{G}_1$, for random dual orthogonal bases $\mathbb{B}$ and $\mathbb{B}^*$, once having seen $\mathbb{B}$ and $\mathbb{B}^* \backslash \{\mathbf{b}_1^*, \mathbf{b}_2^*\}$ and $(y_1, y_1, y_3, \ldots, y_n)_{\mathbb{B}^*}$, for chosen $y_1, y_3, \ldots, y_n \in \mathbb{Z}_q$, one cannot distinguish the vectors $(x_1, 0, x_3, x_4, \ldots, x_n)_{\mathbb{B}}$ and $(0, x_1, x_3, x_4, \ldots, x_n)_{\mathbb{B}}$, for chosen $x_1, x_4, \ldots, x_n \in \mathbb{Z}_q$, but unknown random $x_3 \stackrel{\$}{\leftarrow} \mathbb{Z}_q$.*

Again, for this property, we only work with $(\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)$ and $(\mathbf{b}_1^*, \mathbf{b}_2^*, \mathbf{b}_3^*)$, but without publishing $(\mathbf{b}_1^*, \mathbf{b}_2^*)$.

Indexing and Randomness Amplification. The crucial tool introduced in [OT12b] is the following change of basis, for chosen scalars $t \neq p \in \mathbb{Z}_q$:

$$B = \frac{1}{t-p} \times \begin{pmatrix} t & -p & at \\ -1 & 1 & -a \\ 0 & 0 & t-p \end{pmatrix}_{1,2,3} \qquad\qquad B' = \begin{pmatrix} 1 & 1 & 0 \\ p & t & 0 \\ -a & 0 & 1 \end{pmatrix}_{1,2,3}$$

In this case, we can compute $\mathbb{B} = (\mathbf{b}_i)_i$, but not the vectors $\mathbf{b}_3^*$ as $a \cdot G_2$ is missing.

$$\begin{aligned}
(b, 0, c, x_4, \dots, x_n)_{\mathbb{U}} &= (b, bp, c - ab, x_4, \dots, x_n)_{\mathbb{B}} \\
&= (b \cdot (1, p), \tau, x_4, \dots, x_n)_{\mathbb{B}} \\
((t-p) \cdot (\pi, 0), \delta, y_4, \dots, y_n)_{\mathbb{U}^*} &= (\pi t + at\delta/(t-p), -\pi - a\delta/(t-p), \delta, y_4, \dots, y_n)_{\mathbb{B}^*} \\
&= ((\pi + a\delta/(t-p)) \cdot (t, -1), \delta, y_4, \dots, y_n)_{\mathbb{B}^*}
\end{aligned}$$

There is the limitation for the first two components in $\mathbb{B}$ and $\mathbb{B}^*$ not to be orthogonal: $\langle (1,p), (t,-1) \rangle = (t-p) \neq 0$:

Theorem 22. *Under the DDH Assumption in $\mathbb{G}_1$, for random dual orthogonal bases $\mathbb{B}$ and $\mathbb{B}^*$, once having seen $\mathbb{B}$ and $\mathbb{B}^* \backslash \{\mathbf{b}_3^*\}$, and $(\pi \cdot (t, -1), y_3, \dots, y_n)_{\mathbb{B}^*}$, for chosen $y_3, \dots, y_n \in \mathbb{Z}_q$, but unknown random $\pi \stackrel{\$}{\leftarrow} \mathbb{Z}_q$, and for any chosen $t \neq p \in \mathbb{Z}_q$, one cannot distinguish the vectors $(b \cdot (1,p), \tau, x_4, \dots, x_n)_{\mathbb{B}}$ and $(b \cdot (1,p), 0, x_4, \dots, x_n)_{\mathbb{B}}$, for chosen $x_4, \dots, x_n \in \mathbb{Z}_q$, but unknown random $b, \tau \stackrel{\$}{\leftarrow} \mathbb{Z}_q$.*

As above, we can have a more convenient theorem under the DSDH assumption:

Theorem 23 (Index-Ind Property). *Under the DSDH Assumption in $\mathbb{G}_1$, for random dual orthogonal bases $\mathbb{B}$ and $\mathbb{B}^*$, once having seen $\mathbb{B}$ and $\mathbb{B}^* \backslash \{\mathbf{b}_3^*\}$, and $(\pi \cdot (t, -1), y_3, \dots, y_n)_{\mathbb{B}^*}$, for chosen $y_3, \dots, y_n \in \mathbb{Z}_q$, but unknown random $\pi \stackrel{\$}{\leftarrow} \mathbb{Z}_q$, and for any chosen $t \neq p \in \mathbb{Z}_q$, one cannot distinguish the vectors $(\sigma \cdot (1,p), x_3, x_4, \dots, x_n)_{\mathbb{B}}$ and $(\sigma \cdot (1,p), x_3', x_4, \dots, x_n)_{\mathbb{B}}$, for chosen $x_3', x_3, x_4, \dots, x_n \in \mathbb{Z}_q$, but unknown random $\sigma \stackrel{\$}{\leftarrow} \mathbb{Z}_q$.*

For this property, we only work with $(\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)$ and $(\mathbf{b}_1^*, \mathbf{b}_2^*, \mathbf{b}_3^*)$, but without publishing $\mathbf{b}_3^*$. For a fixed t, we can iteratively update all the other other indices $p \neq t$.

B KP-ABE Scheme

Our ultimate goal is the design of a new KP-ABE scheme with Switchable Attributes. We will start from a variation of the fully-secure attribute-based encryption from [OT12b], that provides some kind of attribute-hiding property, but to handle both delegation and switchable attributes, on-demand. First, we provide a KP-ABE, which is in the same vein as [GPSW06], as it can handle an unbounded number of (classical) attributes and delegation, but with adaptive-set security.

B.1 Description of the KP-ABE Scheme

For the construction, we will use two DPVS, of dimensions 3 and 6 respectively, in a pairing-friendly setting $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, e, G_1, G_2, q)$, using the notations introduced in Section 2.3:

$\mathsf{Setup}(1^\kappa)$**.** The algorithm chooses two random dual orthogonal bases

$$\begin{aligned}
\mathbb{B} &= (\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3) & \mathbb{B}^* &= (\mathbf{b}_1^*, \mathbf{b}_2^*, \mathbf{b}_3^*) \\
\mathbb{D} &= (\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6) & \mathbb{D}^* &= (\mathbf{d}_1^*, \mathbf{d}_2^*, \mathbf{d}_3^*, \mathbf{d}_4^*, \mathbf{d}_5^*, \mathbf{d}_6^*).
\end{aligned}$$

It sets the public parameters $\mathsf{PK} = \{(\mathbf{b}_1, \mathbf{b}_3, \mathbf{b}_1^*), (\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_1^*, \mathbf{d}_2^*, \mathbf{d}_3^*)\}$, whereas the master secret key $\mathsf{MK} = \{\mathbf{b}_3^*\}$. Other basis vectors are kept hidden.

$\mathsf{KeyGen}(\mathsf{MK}, \mathcal{T})$**.** For an access-tree $\mathcal{T}$, the algorithm first chooses a random $a_0 \stackrel{\$}{\leftarrow} \mathbb{Z}_q$, and a random a_0-labeling $(a_\lambda)_\lambda$ of the access-tree $\mathcal{T}$, and builds the key:

$$\mathbf{k}_0^* = (a_0, 0, 1)_{\mathbb{B}^*} \qquad \mathbf{k}_\lambda^* = (\pi_\lambda(1, t_\lambda), a_\lambda, 0, 0, 0)_{\mathbb{D}^*}$$

for all the leaves λ, where $t_\lambda = A(\lambda)$ and $\pi_\lambda \stackrel{\$}{\leftarrow} \mathbb{Z}_q$. The decryption key $\mathsf{dk}_\mathcal{T}$ is then $(\mathbf{k}_0^*, (\mathbf{k}_\lambda^*)_\lambda)$.

$\mathsf{Delegate}(\mathsf{dk}_\mathcal{T}, \mathcal{T}')$**.** The algorithm first generates zero-label credentials for the new attributes, with $\mathbf{k}_\lambda^* \leftarrow (\pi_\lambda \cdot (1, t_\lambda), 0, 0, 0, 0)_{\mathbb{D}^*}$, with $\pi_\lambda \stackrel{\$}{\leftarrow} \mathbb{Z}_q$, for a new leaf. Keeping only the credentials useful in $\mathcal{T}'$, it gets a valid key from $\mathsf{dk}_\mathcal{T}$. It can thereafter be randomized with a random $a_0' \stackrel{\$}{\leftarrow} \mathbb{Z}_q$ and a random a_0'-labeling (a_λ') of $\mathcal{T}'$, with $\mathbf{k}_0^* \leftarrow \mathbf{k}_0^* + (a_0', 0, 0)_{\mathbb{B}^*}$, and $\mathbf{k}_\lambda^* \leftarrow \mathbf{k}_\lambda^* + (\pi_\lambda' \cdot (1, t_\lambda), a_\lambda', 0, 0, 0)_{\mathbb{D}^*}$, for $\pi_\lambda' \stackrel{\$}{\leftarrow} \mathbb{Z}_q$.

$\mathsf{Encaps}(\mathsf{PK}, \Gamma)$**.** For the set Γ of attributes, the algorithm first chooses random scalars $\omega, \xi \stackrel{\$}{\leftarrow} \mathbb{Z}_q$. It then sets $K = g_t^\xi$ and generates the ciphertext $C = (\mathbf{c}_0, (\mathbf{c}_t)_{t\in\Gamma})$ where $\mathbf{c}_0 = (\omega, 0, \xi)_\mathbb{B}$ and $\mathbf{c}_t = (\sigma_t(t, -1), \omega, 0, 0, 0)_\mathbb{D}$, for all the attributes $t \in \Gamma$ and $\sigma_t \stackrel{\$}{\leftarrow} \mathbb{Z}_q$.

$\mathsf{Decaps}(\mathsf{dk}_\mathcal{T}, C)$**.** The algorithm first selects an evaluation pruned tree $\mathcal{T}'$ of $\mathcal{T}$ that is satisfied by Γ. This means that the labels a_λ for all the leaves λ in $\mathcal{T}'$ allow to reconstruct a_0 by simple additions.
Note that from every leaf λ in $\mathcal{T}'$ and $t = t_\lambda = A(\lambda) \in \Gamma$, it can compute

$$\mathbf{c}_t \times \mathbf{k}_t^* = g_t^{\sigma_t \cdot \pi_\lambda \cdot \langle (t,-1),(1,t)\rangle + \omega\cdot a_\lambda} = g_t^{\omega \cdot a_\lambda}.$$

Hence, it can derive $g_t^{\omega\cdot a_0}$. From $\mathbf{c}_0$ and $\mathbf{k}_0^*$, it gets $\mathbf{c}_0 \times \mathbf{k}_0^* = g_t^{\omega\cdot a_0 + \xi}$ which then easily leads to $K = g_t^\xi$.

We stress that in the above decryption, one can recover $g_t^{\omega\cdot a_0}$ if and only if there is an evaluation pruned tree $\mathcal{T}'$ of $\mathcal{T}$ that is satisfied by Γ. And this holds if and only if $\mathcal{T}(\Gamma) = 1$. Additionally, since $\mathbf{b}_3^*$ is not public but in MK only, for the key issuer, only the latter can issue keys, but anybody can delegate a key for a tree $\mathcal{T}$ into a key for a more restrictive tree $\mathcal{T}'$. As everything can be randomized (the random coins π_λ and the labeling), the delegated keys are perfectly indistinguishable from fresh keys. Hence, given two keys possibly delegated from a common key, one cannot decide whether they have been independently generated or delegated.

B.2 Security Analysis of the KP-ABE

We first consider the security analysis, without delegation, as it is quite similar to [OT12b], but under the SXDH assumption instead of the DLIN assumption:

Theorem 24. *Under the SXDH assumption, no adversary can win the IND security game (without delegation) against our KP-ABE scheme, in the Adaptive-Set setting, with non-negligible advantage.*

This theorem is proven in details in the appendix B.3, with exact bound for an adversary with running time bounded by t, with at most P attributes involved in the full experiment and at most K queries to the OKeyGen-oracle:

$$\begin{aligned}\mathsf{Adv}^{\mathsf{ind}}(\mathcal{A}) &\leq 2(KP^2+1) \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t) + (3P+1)K \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t) \\ &\leq (2KP^2 + 3KP + K + 2) \times \mathsf{Adv}^{\mathsf{sxdh}}(t)\end{aligned}$$

The global sequence of games is described on Figure 10, with another sequence of sub-games on Figure 11. In the two first games $\mathbf{G}_1$ and $\mathbf{G}_2$, one is preparing the floor with a random τ and random masks z_t in the ciphertexts $\mathbf{c}_t$ (actually, the challenge ciphertext corresponding to the attribute t). Note that until the challenge query is asked, one does not exactly know the attributes in Γ (as we are in the adaptive-set setting), but we prepare all the possible $\mathbf{c}_t$, and only the ones corresponding to attributes in Γ will be provided to the adversary. The main step

$\mathbf{G}_0$ Real IND-Security game (without delegation)

$$\begin{array}{rlllll}
\mathbf{c}_0 = (\ \omega & 0 & \xi \ ) & \mathbf{c}_t = (\ \sigma_t(1,t) & \omega \mid 0 & 0 \quad 0 \ ) \\
\mathbf{k}^*_{\ell,0} = (\ a_{\ell,0} & 0 & 1 \ ) & \mathbf{k}^*_{\ell,\lambda} = (\ \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) & a_{\ell,\lambda} \mid 0 & 0 \quad 0 \ )
\end{array}$$

$\mathbf{G}_1$ $\mathsf{SubSpace\text{-}Ind}$ Property, on $(\mathbb{B},\mathbb{B}^*)_{1,2}$ and $(\mathbb{D},\mathbb{D}^*)_{3,4}$, between 0 and $\tau \xleftarrow{\$} \mathbb{Z}_q$

$$\begin{array}{rlllll}
\mathbf{c}_0 = (\ \omega & \tau & \xi \ ) & \mathbf{c}_t = (\ \sigma_t(1,t) & \omega \mid \tau & 0 \quad 0 \ ) \\
\mathbf{k}^*_{\ell,0} = (\ a_{\ell,0} & 0 & 1 \ ) & \mathbf{k}^*_{\ell,\lambda} = (\ \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) & a_{\ell,\lambda} \mid 0 & 0 \quad 0 \ )
\end{array}$$

$\mathbf{G}_2$ $\mathsf{SubSpace\text{-}Ind}$ Property, on $(\mathbb{D},\mathbb{D}^*)_{1,2,6}$, between 0 and τz_t

$$\begin{array}{rlllll}
\mathbf{c}_0 = (\ \omega & \tau & \xi \ ) & \mathbf{c}_t = (\ \sigma_t(1,t) & \omega \mid \tau & 0 \quad \tau z_t \ ) \\
\mathbf{k}^*_{\ell,0} = (\ a_{\ell,0} & 0 & 1 \ ) & \mathbf{k}^*_{\ell,\lambda} = (\ \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) & a_{\ell,\lambda} \mid 0 & 0 \quad 0 \ )
\end{array}$$

$\mathbf{G}_3$ Introduction of an additional random-labeling. See Figure 11

$$\begin{array}{rlllll}
\mathbf{c}_0 = (\ \omega & \tau & \xi \ ) & \mathbf{c}_t = (\ \sigma_t(1,t) & \omega \mid \tau & 0 \quad \tau z_t \ ) \\
\mathbf{k}^*_{\ell,0} = (\ a_{\ell,0} & r_{\ell,0} & 1 \ ) & \mathbf{k}^*_{\ell,\lambda} = (\ \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) & a_{\ell,\lambda} \mid 0 & 0 \quad s_{\ell,\lambda}/z_{t_{\ell,\lambda}} \ )
\end{array}$$

$\mathbf{G}_4$ Formal basis change, on $(\mathbb{B},\mathbb{B}^*)_{2,3}$, to randomize ξ

$$\begin{array}{rlllll}
\mathbf{c}_0 = (\ \omega & \tau & \xi'' \ ) & \mathbf{c}_t = (\ \sigma_t(1,t) & \omega \mid \tau & 0 \quad \tau z_t \ ) \\
\mathbf{k}^*_{\ell,0} = (\ a_{\ell,0} & r_{\ell,0} & 1 \ ) & \mathbf{k}^*_{\ell,\lambda} = (\ \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) & a_{\ell,\lambda} \mid 0 & 0 \quad s_{\ell,\lambda}/z_{t_{\ell,\lambda}} \ )
\end{array}$$

Gray cells x mean they have been changed in this game.

Fig. 10: Global sequence of games for the IND-security proof of the $\mathsf{KP\text{-}ABE}$

$\mathbf{G}_{2.k.0}$ Hybrid game for $\mathbf{G}_2$, with $1 \leq k \leq K+1$ (from Figure 10)

$$\begin{array}{lrlllll}
 & \mathbf{c}_0 = (\ \omega & \tau & \xi \ ) & \mathbf{c}_t = (\ \sigma_t(1,t) & \omega \mid \tau & 0 \quad \tau z_t \ ) \\
\ell < k & \mathbf{k}^*_{\ell,0} = (\ a_{\ell,0} & r_{\ell,0} & 1 \ ) & \mathbf{k}^*_{\ell,\lambda} = (\ \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) & a_{\ell,\lambda} \mid 0 & 0 \quad s_{\ell,\lambda}/z_{t_{\ell,\lambda}} \ ) \\
\ell \geq k & \mathbf{k}^*_{\ell,0} = (\ a_{\ell,0} & 0 & 1 \ ) & \mathbf{k}^*_{\ell,\lambda} = (\ \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) & a_{\ell,\lambda} \mid 0 & 0 \quad 0 \ )
\end{array}$$

$\mathbf{G}_{2.k.1}$ $\mathsf{SubSpace\text{-}Ind}$ Property, on $(\mathbb{B}^*,\mathbb{B})_{1,2}$ and $(\mathbb{D}^*,\mathbb{D})_{3,4}$, between 0 and $s_{k,*}$

$$\begin{array}{rlllll}
\mathbf{k}^*_{k,0} = (\ a_{k,0} & s_{k,0} & 1 \ ) & \mathbf{k}^*_{k,\lambda} = (\ \pi_{k,\lambda}(t_{k,\lambda},-1) & a_{k,\lambda} \mid s_{k,\lambda} & 0 \quad 0 \ )
\end{array}$$

$\mathbf{G}_{2.k.2}$ Masking of the labeling. See Figure 12

$$\begin{array}{rlllll}
\mathbf{k}^*_{k,0} = (\ a_{k,0} & s_{k,0} & 1 \ ) & \mathbf{k}^*_{k,\lambda} = (\ \pi_{k,\lambda}(t_{k,\lambda},-1) & a_{k,\lambda} \mid 0 & 0 \quad s_{k,\lambda}/z_{t_{k,\lambda}} \ )
\end{array}$$

$\mathbf{G}_{2.k.3}$ Limitations on KeyGen-queries: $s_{k,0}$ unpredictable, replaced by a random $r_{k,0}$

$$\begin{array}{rlllll}
\mathbf{k}^*_{k,0} = (\ a_{k,0} & r_{k,0} & 1 \ ) & \mathbf{k}^*_{k,\lambda} = (\ \pi_{k,\lambda}(t_{k,\lambda},-1) & a_{k,\lambda} \mid 0 & 0 \quad s_{k,\lambda}/z_{t_{k,\lambda}} \ )
\end{array}$$

Fig. 11: Sequence of games on the K keys for the IND-security proof of the $\mathsf{KP\text{-}ABE}$

is to get to Game $\mathbf{G}_3$, with an additional labeling $(s_{\ell,0},(s_{\ell,\lambda})_\lambda)$, using hybrid games starting from Game $\mathbf{G}_2$. The sequence on Figure 11 gives more details: the new labelling is added in each ℓ-th key (in $\mathbf{G}_{2.k.1}$), then each label is masked by the random z_t for each attribute t (in $\mathbf{G}_{2.k.2}$). In order to go to game $\mathbf{G}_{2.k.3}$ one exploits the limitations one expects from the adversary in the security game: the adversary cannot ask keys on access-trees $\mathcal{T}$ such that $\mathcal{T}(\Gamma)=1$, for the challenge set Γ.

We stress that this construction makes more basis vectors public, and only $\mathbf{b}_3^*$ is for the key issuer, contrarily to the original proof. This is the reason why we can deal with delegation for any user. In addition, as delegation provides keys that are perfectly indistinguishable from fresh keys, one can easily get the full result:

Corollary 25. *Under the SXDH assumption, no adversary can win the Del-IND security game against the KP-ABE scheme, in the Adaptive-Set setting, with non-negligible advantage.*

The bound is the same, expect K is the global number of OKeyGen and ODelegate queries.

B.3 IND-Security Proof of the KP-ABE Scheme

In this section, we will focus on the IND-security proof of the KP-ABE scheme, where the definition is quite similar to Definition 4, but without the Delegation-Oracle.

Definition 26 (Indistinguishability). *IND-security for KP-ABE is defined by the following game:*

Initialize: *The challenger runs the* Setup *algorithm of KP-ABE and gives the public parameters* PK *to the adversary;*

OKeyGen($\mathcal{T}$**:** *The adversary is allowed to issue* KeyGen*-queries for any access-tree $\mathcal{T}$ of its choice, and gets back the decryption key* $\mathsf{dk}_{\mathcal{T}}$*;*

RoREncaps(Γ)**:** *The adversary submits one real-or-random encapsulation query on a set of attributes Γ. The challenger asks for an encapsulation query on Γ and receives (K_0, C). It also generates a random key K_1. It eventually flips a random coin b, and outputs (K_b, C) to the adversary;*

Finalize(b'): *The adversary outputs a guess b' for b. If for some access-tree $\mathcal{T}$ asked to the* OKeyGen*-oracle, $\mathcal{T}(\Gamma)=1$, on the challenge set Γ, $\beta \xleftarrow{\$} \{0,1\}$, otherwise one sets $\beta = b'$. One outputs β.*

The advantage of an adversary $\mathcal{A}$ in this game is defined as

$$\mathsf{Adv}^{\mathsf{ind}}(\mathcal{A}) = \Pr[\beta = 1|b = 1] - \Pr[\beta = 1|b = 0].$$

The global sequence of games will follow the steps shown on Figure 10. But while the first steps (from $\mathbf{G}_0$ to $\mathbf{G}_2$) will be simple, the big step from $\mathbf{G}_2$ to $\mathbf{G}_3$ will need multiple hybrid games, presented on Figure 11. All theses games work in a pairing-friendly setting $(\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_t,e,G_1,G_2,q)$, with two random dual orthogonal bases $(\mathbb{B},\mathbb{B}^*)$ and $(\mathbb{D},\mathbb{D}^*)$ of size 3 and 6, respectively.

In the following proof, we will use t to denote attributes, and thus the indices for the possible ciphertexts $\mathbf{c}_t$ associated to each attribute in the challenge ciphertext. We indeed anticipate all the possible $\mathbf{c}_t$, before knowing the exact set Γ, as we are in the adaptive setting. The variable p will be used in hybrid proofs to specify a particular attribute. We will denote P the maximal number of attributes involved in the game (either in a ciphertext or in a key). Then $1 \le t, p \le P$. Similarly, we will use ℓ to denote key queries, and thus the index of the global ℓ-th key $\mathbf{k}_\ell^*$, whereas λ will we used for the leaf in the tree of the key-query: $\mathbf{k}_{\ell,\lambda}^*$ is thus the specific key for leaf λ in the global ℓ-th key. The variable k will be used in hybrid proofs to specify a particular key-query index. We will denote K the maximal number of key-queries. Then $1 \le \ell, k \le K$.

Game $\mathbf{G}_0$: This is the real game where the simulator generates all the private information and sets $\mathsf{PK} = \{(\mathbf{b}_1, \mathbf{b}_3, \mathbf{b}_1^*), (\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_1^*, \mathbf{d}_2^*, \mathbf{d}_3^*)\}$ and $\mathsf{MK} = \{\mathbf{b}_3^*\}$. The public parameters PK are provided to the adversary

$\mathsf{OKeyGen}(\mathcal{T}_\ell)$**:** The adversary is allowed to issue KeyGen-queries on an access-tree $\mathcal{T}_\ell$ (for the ℓ-th query), for which the challenger chooses a random scalar $a_{\ell,0} \stackrel{\$}{\leftarrow} \mathbb{Z}_q$ and a random $a_{\ell,0}$-labeling $(a_{\ell,\lambda})_\lambda$ of the access-tree $\mathcal{T}_\ell$, and builds the key:

$$\mathbf{k}^*_{\ell,0} = (a_{\ell,0}, 0, 1)_{\mathbb{B}^*} \qquad \mathbf{k}^*_{\ell,\lambda} = (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, 0, 0)_{\mathbb{D}^*}$$

for all the leaves λ, where $t_{\ell,\lambda} = A(\lambda)$ is the attribute associated to the leaf λ in $\mathcal{T}_\ell$ and $\pi_{\ell,\lambda} \stackrel{\$}{\leftarrow} \mathbb{Z}_q$. The decryption key dk_ℓ is then $(\mathbf{k}^*_{\ell,0}, (\mathbf{k}^*_{\ell,\lambda})_\lambda)$;

$\mathsf{RoREncaps}(\Gamma)$**:** On the unique query on a set of attributes Γ, the challenger chooses random scalars $\omega, \xi, \xi' \stackrel{\$}{\leftarrow} \mathbb{Z}_q$. It then sets $K_0 = g_t^\xi$ and $K_1 = g_t^{\xi'}$. It generates the ciphertext $C = (\mathbf{c}_0, (\mathbf{c}_t)_{t\in\Gamma})$ where

$$\mathbf{c}_0 = (\omega, 0, \xi)_{\mathbb{B}} \qquad \mathbf{c}_t = (\sigma_t(1, t), \omega, 0, 0, 0)_{\mathbb{D}}$$

for all the attributes $t \in \Gamma$ and $\sigma_t \stackrel{\$}{\leftarrow} \mathbb{Z}_q$. According to the real or random game (bit $b \stackrel{\$}{\leftarrow} \{0, 1\}$), one outputs (K_b, C).

Eventually, on adversary's guess b' for b, if for some $\mathcal{T}_\ell$, $\mathcal{T}_\ell(\Gamma) = 1$, then $\beta \stackrel{\$}{\leftarrow} \{0, 1\}$, otherwise $\beta = b'$. Then $\mathsf{Adv}_0 = \Pr[\beta = 1|b = 1] - \Pr[\beta = 1|b = 0]$.

In the next games, we gradually modify the simulations of OKeyGen and RoREncaps oracles, but always (at least) with random $\omega, \xi, \xi', (\sigma_t) \stackrel{\$}{\leftarrow} \mathbb{Z}_q$, $(a_{\ell,0}), (\pi_{\ell,\lambda}) \stackrel{\$}{\leftarrow} \mathbb{Z}_q$, and random $a_{\ell,0}$-labeling $(a_{\ell,\lambda})_\lambda$ of the access-tree $\mathcal{T}_\ell$ for each OKeyGen-query.

Game $\mathbf{G}_1$: One chooses random $\tau \stackrel{\$}{\leftarrow} \mathbb{Z}_q$, and sets (which differs for the ciphertext only)

$$\mathbf{c}_0 = (\omega, \tau, \xi)_{\mathbb{B}} \qquad \mathbf{c}_t = (\sigma_t(1, t), \omega, \tau, 0, 0)_{\mathbb{D}}$$
$$\mathbf{k}^*_{\ell,0} = (a_{\ell,0}, 0, 1)_{\mathbb{B}^*} \qquad \mathbf{k}^*_{\ell,\lambda} = (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, 0, 0)_{\mathbb{D}^*}$$

The previous game and this game are indistinguishable under the DDH assumption in $\mathbb{G}_1$: one applies the SubSpace-Ind property from Theorem 19, on $(\mathbb{B}, \mathbb{B}^*)_{1,2}$ and $(\mathbb{D}, \mathbb{D}^*)_{3,4}$. Indeed, we can consider a triple $(a \cdot G_1, b \cdot G_1, c \cdot G_1)$, where $c = ab + \tau \bmod q$ with either $\tau = 0$ or $\tau \stackrel{\$}{\leftarrow} \mathbb{Z}_q^*$, which are indistinguishable situations under the DDH assumption.

Let us assume we start from random dual orthogonal bases $(\mathbb{U}, \mathbb{U}^*)$ and $(\mathbb{V}, \mathbb{V}^*)$ of size 3 and 6 respectively. Then we define the matrices

$$B = \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}_{1,2} \qquad B' = \begin{pmatrix} 1 & 0 \\ -a & 1 \end{pmatrix}_{1,2} \qquad D = \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}_{3,4} \qquad D' = \begin{pmatrix} 1 & 0 \\ -a & 1 \end{pmatrix}_{3,4}$$
$$\mathbb{B} = B \cdot \mathbb{U} \qquad \mathbb{B}^* = B' \cdot \mathbb{U}^* \qquad \mathbb{D} = D \cdot \mathbb{V} \qquad \mathbb{D}^* = D' \cdot \mathbb{V}^*$$

Note that we can compute all the basis vectors excepted $\mathbf{b}_2^*$ and $\mathbf{d}_4^*$, that nobody needs: the vectors below have these coordinates at zero. So one can set

$$\mathbf{c}_0 = (b, c, \xi)_{\mathbb{U}} = (b, \tau, \xi)_{\mathbb{B}} \qquad \mathbf{c}_t = (\sigma_t(1, t), b, c, 0, 0)_{\mathbb{V}} = (\sigma_t(1, t), b, \tau, 0, 0)_{\mathbb{D}}$$
$$\mathbf{k}^*_{\ell,0} = (a_{\ell,0}, 0, 1)_{\mathbb{B}^*} \qquad \mathbf{k}^*_{\ell,\lambda} = (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, 0, 0)_{\mathbb{D}^*}$$

When $\tau = 0$, this is exactly the previous game, with $\omega = b$, for a random τ, this is the current game: $\mathsf{Adv}_0 - \mathsf{Adv}_1 \leq \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t)$.

Game $\mathbf{G}_2$: One continues to modify the ciphertext, with random $\tau, (z_t) \stackrel{\$}{\leftarrow} \mathbb{Z}_q$:

$$\mathbf{c}_0 = (\omega, \tau, \xi)_{\mathbb{B}} \qquad \mathbf{c}_t = (\sigma_t(1, t), \omega, \tau, 0, \tau z_t)_{\mathbb{D}}$$
$$\mathbf{k}^*_{\ell,0} = (a_{\ell,0}, 0, 1)_{\mathbb{B}^*} \qquad \mathbf{k}^*_{\ell,\lambda} = (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, 0, 0)_{\mathbb{D}^*}$$

The previous game and this game are indistinguishable under the DDH assumption in $\mathbb{G}_1$: one applies again the SubSpace-Ind property from Theorem 19, on $(\mathbb{D},\mathbb{D}^*)_{(1,2),6}$. Indeed, we can consider a triple $(a\cdot G_1, b\cdot G_1, c\cdot G_1)$, where $c = ab+\zeta \bmod q$, with either $\zeta = 0$ or $\zeta \overset{\$}{\leftarrow} \mathbb{Z}_q^*$, which are indistinguishable situations under the DDH assumption.
Let us assume we start from random dual orthogonal bases $(\mathbb{B},\mathbb{B}^*)$ and $(\mathbb{V},\mathbb{V}^*)$ of size 3 and 6 respectively. Then we define the matrices

$$D = \begin{pmatrix} 1 & 0 & a \\ 0 & 1 & a \\ 0 & 0 & 1 \end{pmatrix}_{1,2,6} \qquad D' = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -a & -a & 1 \end{pmatrix}_{1,2,6} \qquad \mathbb{D} = D\cdot\mathbb{V} \qquad \mathbb{D}^* = D'\cdot\mathbb{V}^*$$

Note that we can compute all the basis vectors excepted $\mathbf{d}_6^*$, that nobody needs: the vectors below have these coordinates at zero. One chooses additional random scalars $\alpha_t,\beta_t \overset{\$}{\leftarrow} \mathbb{Z}_q$ to virtually set $b_t = \alpha_t\cdot b+\beta_t$ and $c_t = \alpha_t\cdot c+\beta_t\cdot a$, which makes $c_t - ab_t = \alpha_t\cdot\zeta$. One can set

$$\begin{aligned}
\mathbf{c}_0 &= (\omega,\tau,\xi)_{\mathbb{B}} & \mathbf{c}_t &= (b_t(1,t),\omega,\tau,0,c_t(t+1))_{\mathbb{V}}\\
&& &= (b_t(1,t),\omega,\tau,0,c_t(t+1)-ab_t-ab_tt)_{\mathbb{D}}\\
&& &= (b_t(1,t),\omega,\tau,0,c_t(t+1)-ab_t(1+t))_{\mathbb{D}}\\
&& &= (b_t(1,t),\omega,\tau,0,\alpha_t\cdot\zeta\cdot(t+1))_{\mathbb{D}}\\
&& &= (b_t(1,t),\omega,\tau,0,\tau z_t)_{\mathbb{D}}\\
\mathbf{k}_{\ell,0}^* &= (a_{\ell,0},0,1)_{\mathbb{B}^*} & \mathbf{k}_{\ell,\lambda}^* &= (\pi_{\ell,\lambda}(t_{\ell,\lambda},-1),a_{\ell,\lambda},0,0,0)_{\mathbb{D}^*}
\end{aligned}$$

where $z_t = \alpha_t\cdot\zeta\cdot(t+1)/\tau$. When $\zeta = 0$, this is exactly the previous game, as $z_t = 0$, with $\pi_t = b_t = \alpha_t\cdot b+\beta_t$, whereas for a random ζ, this is the current game: $\mathsf{Adv}_1 - \mathsf{Adv}_2 \le \mathsf{Adv}_{\mathbb{G}_1}^{\mathsf{ddh}}(t)$.

Game $\mathbf{G}_3$: We introduce a second independent $s_{\ell,0}$-labeling $s_{\ell,\lambda}$ for each access-tree $\mathcal{T}_\ell$ and a random $r_{\ell,0}$ to define

$$\mathbf{k}_{\ell,0}^* = (a_{\ell,0},r_{\ell,0},1)_{\mathbb{B}^*} \qquad \mathbf{k}_{\ell,\lambda}^* = (\pi_{\ell,\lambda}(t_{\ell,\lambda},-1),a_{\ell,\lambda},0,0,s_{\ell,\lambda}/z_{t_{k,\lambda}})_{\mathbb{D}^*}$$

But to this, we move to a sub-sequence of hybrid games, with distinct ways for answering the $k-1$ first key queries and the last ones, as explained on Figure 11: for the ℓ-th key generation query on $\mathcal{T}_\ell$, the challenger chooses three random scalars $a_{\ell,0}, r_{\ell,0}, s_{\ell,0} \overset{\$}{\leftarrow} \mathbb{Z}_q$, and two random $a_{\ell,0}$-labeling $(a_{\ell,\lambda})_\lambda$ and $s_{\ell,0}$-labeling $(s_{\ell,\lambda})_\lambda$ of the access-tree $\mathcal{T}_\ell$, and builds the key $(\mathbf{k}_{\ell,0}^*,(\mathbf{k}_{\ell,\lambda}^*)_\lambda)$, with $\pi_{\ell,\lambda} \overset{\$}{\leftarrow} \mathbb{Z}_q$:

$$\begin{aligned}
&\ell < k & \mathbf{k}_{\ell,0}^* &= (a_{\ell,0},r_{\ell,0},1)_{\mathbb{B}^*} & \mathbf{k}_{\ell,\lambda}^* &= (\pi_{\ell,\lambda}(t_{\ell,\lambda},-1),a_{\ell,\lambda},0,0,s_{\ell,\lambda}/z_{t_{\ell,\lambda}})_{\mathbb{D}^*}\\
&\ell \ge k & \mathbf{k}_{\ell,0}^* &= (a_{\ell,0},0,1)_{\mathbb{B}^*} & \mathbf{k}_{\ell,\lambda}^* &= (\pi_{\ell,\lambda}(t_{\ell,\lambda},-1),a_{\ell,\lambda},0,0,0)_{\mathbb{D}^*}
\end{aligned}$$

For this game, we have to anticipate the values z_t, for each attribute t, before knowing Γ, for the challenge ciphertext, as we have to introduce $z_{t_{\ell,\lambda}}$ during the creation of the leaves. These z_t are thus random values chosen as soon as an attribute t is involved in the security game.
When $k = 1$, this is exactly the game $\mathbf{G}_2$: $\mathbf{G}_2 = \mathbf{G}_{2.1.0}$, whereas for $k = K+1$ this is exactly the expected game $\mathbf{G}_3$: $\mathbf{G}_3 = \mathbf{G}_{2.K+1.0}$. We now consider any $k \in \{1,\ldots,K\}$, to show that $\mathbf{G}_{2.k.3} = \mathbf{G}_{2.k+1.0}$, where all the keys for $\ell \ne k$ will be defined using the basis vectors of $(\mathbb{B}^*,\mathbb{D}^*)$ and known scalars. We only focus on the k-th key and the ciphertext, but still with random $\omega,\tau,\xi,\xi',(\sigma_t),(z_t) \overset{\$}{\leftarrow} \mathbb{Z}_q$, random $a_{k,0},(\pi_{k,\lambda}) \overset{\$}{\leftarrow} \mathbb{Z}_q$, as well as a random $a_{k,0}$-labeling $(a_{k,\lambda})_\lambda$ of the access-tree $\mathcal{T}_k$, but also $s_{k,0} \overset{\$}{\leftarrow} \mathbb{Z}_q$ and a second independent random $s_{k,0}$-labeling $(s_{k,\lambda})_\lambda$ of the access-tree $\mathcal{T}_k$:

Game $\mathbf{G}_{2.k.0}$ This is exactly as described above, for $\ell = k$:

$$\begin{aligned}
\mathbf{c}_0 &= (\omega,\tau,\xi)_{\mathbb{B}} & \mathbf{c}_t &= (\sigma_t(1,t),\omega,\tau,0,\tau z_t)_{\mathbb{D}}\\
\mathbf{k}_{k,0}^* &= (a_{k,0},0,1)_{\mathbb{B}^*} & \mathbf{k}_{k,\lambda}^* &= (\pi_{k,\lambda}(t_{k,\lambda},-1),a_{k,\lambda},0,0,0)_{\mathbb{D}^*}
\end{aligned}$$

Game $\mathbf{G}_{2.k.1}$ One now introduces the second labeling:

$$\begin{aligned} \mathbf{c}_0 &= (\omega, \tau, \xi)_{\mathbb{B}} & \mathbf{c}_t &= (\sigma_t(1,t), \omega, \tau, 0, \tau z_t)_{\mathbb{D}} \\ \mathbf{k}_0^* &= (a_{k,0}, s_{k,0}, 1)_{\mathbb{B}^*} & \mathbf{k}_{k,\lambda}^* &= (\pi_{k,\lambda}(t_{k,\lambda}, -1), a_{k,\lambda}, s_{k,\lambda}, 0, 0)_{\mathbb{D}^*} \end{aligned}$$

This game is indistinguishable from the previous one under the DDH assumption in $\mathbb{G}_2$: one applies the SubSpace-Ind property from Theorem 19 on $(\mathbb{B}^*, \mathbb{B})_{1,2}$ and $(\mathbb{D}^*, \mathbb{D})_{3,4}$. Indeed, we can consider a triple $(a \cdot G_2, b \cdot G_2, c \cdot G_2)$, where $c = ab + \rho \bmod q$, with either $\rho = 0$ or $\rho \overset{\$}{\leftarrow} \mathbb{Z}_q^*$, which are indistinguishable situations under the DDH assumption.
Let us assume we start from random dual orthogonal bases $(\mathbb{U}, \mathbb{U}^*)$ and $(\mathbb{V}, \mathbb{V}^*)$ of size 3 and 6 respectively. Then we define the matrices

$$\begin{aligned} B' &= \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}_{1,2} & B &= \begin{pmatrix} 1 & 0 \\ -a & 1 \end{pmatrix}_{1,2} & D' &= \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}_{3,4} & D &= \begin{pmatrix} 1 & 0 \\ -a & 1 \end{pmatrix}_{3,4} \\ \mathbb{B}^* &= B' \cdot \mathbb{U}^* & \mathbb{B} &= B \cdot \mathbb{U} & \mathbb{D}^* &= D' \cdot \mathbb{V}^* & \mathbb{D} &= D \cdot \mathbb{V} \end{aligned}$$

Note that we can compute all the basis vectors excepted $\mathbf{b}_2$ and $\mathbf{d}_4$. But we can define the ciphertext vectors in the original bases $(\mathbb{U}, \mathbb{V})$, and all the keys in bases $(\mathbb{B}^*, \mathbb{D}^*)$, excepted the k-th one:

$$\begin{aligned} \mathbf{c}_0 &= (\omega, \tau, \xi)_{\mathbb{U}} = (\omega + a\tau, \tau, \xi)_{\mathbb{B}} \\ \mathbf{c}_t &= (\sigma_t(1,t), \omega, \tau, 0, \tau z_t)_{\mathbb{V}} = (\sigma_t(1,t), \omega + a\tau, \tau, 0, \tau z_t)_{\mathbb{D}} \\ \mathbf{k}_{k,0}^* &= (b_0, 0, 1)_{\mathbb{B}^*} + (b, c, 0)_{\mathbb{U}^*} = (b_0, 0, 1)_{\mathbb{B}^*} + (b, \rho, 0)_{\mathbb{B}^*} = (b_0 + b, \rho, 1)_{\mathbb{B}^*} \\ \mathbf{k}_{k,\lambda}^* &= (\pi_{k,\lambda}(t_{k,\lambda}, -1), b_\lambda, 0, 0, 0)_{\mathbb{D}^*} + (0, 0, b \cdot b'_\lambda, c \cdot b'_\lambda, 0, 0)_{\mathbb{V}^*} \\ &= (\pi_{k,\lambda}(t_{k,\lambda}, -1), b_\lambda, 0, 0, 0)_{\mathbb{D}^*} + (0, 0, b \cdot b'_\lambda, \rho \cdot b'_\lambda, 0, 0)_{\mathbb{D}^*} \\ &= (\pi_{k,\lambda}(t_{k,\lambda}, -1), b_\lambda + b \cdot b'_\lambda, \rho \cdot b'_\lambda, 0, 0)_{\mathbb{D}^*} \end{aligned}$$

with $b_0 \overset{\$}{\leftarrow} \mathbb{Z}_q$, a random b_0-labeling $(b_\lambda)_\lambda$, and a random 1-labeling $(b'_\lambda)_\lambda$ of $\mathcal{T}_k$. When $\rho = 0$, this is exactly the previous game, with $\omega = \omega + a\tau$, and $a_{k,0} = b_0 + b$, $a_{k,\lambda} = b_\lambda + b \cdot b'_\lambda$, whereas for a random ρ, this is the current game, with additional $s_{k,0} = \rho$, $s_{k,\lambda} = \rho \cdot b'_\lambda$: $\mathsf{Adv}_{2.k.0} - \mathsf{Adv}_{2.k.1} \leq \mathsf{Adv}_{\mathbb{G}_2}^{\mathsf{ddh}}(t)$.

Game $\mathbf{G}_{2.k.2}$ With the same inputs, one just changes as follows

$$\begin{aligned} \mathbf{c}_0 &= (\omega, \tau, \xi)_{\mathbb{B}} & \mathbf{c}_t &= (\sigma_t(1,t), \omega, \tau, 0, \tau z_k)_{\mathbb{D}} \\ \mathbf{k}_{k,0}^* &= (a_{k,0}, s_{k,0}, 1)_{\mathbb{B}^*} & \mathbf{k}_{k,\lambda}^* &= (\pi_{k,\lambda}(t_{k,\lambda}, -1), a_{k,\lambda}, 0, 0, s_{k,\lambda}/z_k)_{\mathbb{D}^*} \end{aligned}$$

Unfortunately, for the latter gap, which intuitively exploits the Swap-Ind property from Theorem 21, we cannot do all the changes at once. Then, the Index-Ind property will be applied first, with Theorem 23.
We will thus describe another sequence of games, as shown on Figure 12, where $\mathbf{G}_{2.k.1.p.0}$ with $p = 1$ is the previous game: $\mathbf{G}_{2.k.1} = \mathbf{G}_{2.k.1.1.0}$; for any p, $\mathbf{G}_{2.k.1.p.5}$ is $\mathbf{G}_{2.k.1.p+1.0}$; and $\mathbf{G}_{2.k.1.p.0}$ with $p = P + 1$ is the current game: $\mathbf{G}_{2.k.2} = \mathbf{G}_{2.k.1.P+1.0}$. For each p, we prove that

$$\mathsf{Adv}_{2.k.1.p.0} - \mathsf{Adv}_{2.k.1.p.5} \leq 2P \times \mathsf{Adv}_{\mathbb{G}_1}^{\mathsf{ddh}}(t) + 3 \times \mathsf{Adv}_{\mathbb{G}_2}^{\mathsf{ddh}}(t).$$

Hence, globally, we have

$$\mathsf{Adv}_{2.k.1} - \mathsf{Adv}_{2.k.2} \leq 2P^2 \times \mathsf{Adv}_{\mathbb{G}_1}^{\mathsf{ddh}}(t) + 3P \times \mathsf{Adv}_{\mathbb{G}_2}^{\mathsf{ddh}}(t).$$

But before proving this huge gap, let us conclude the analysis.

Game $\mathbf{G}_{2.k.3}$ In the above game, to be a legitimate attack (that does not output a random bit β in the Finalize procedure, but the actual output b' of the adversary), for all the key queries $\mathcal{T}_\ell$, one must have $\mathcal{T}_\ell(\Gamma) = 0$. In particular, $\mathcal{T}_k(\Gamma) = 0$: this means that there are missing attributes in the ciphertext, and thus false leaves to make the access-tree no acceptable. More concretely, a missing attribute t means $\mathbf{c}_t$ is not provided to the adversary, and so no information about z_t is leaked. As the key only contains $s_{k,\lambda}/z_{t_{k,\lambda}}$, the missing $z_{t_{k,\lambda}}$ guarantees that no information leaks about $s_{k,\lambda}$: all the false leaves λ correspond to these $s_{k,\lambda}$ that are unknown: only $(s_{k,\lambda})_{\lambda\in\mathcal{L}_\Gamma}$ is known, and so the root $s_{k,0}$ is unpredictable.

Remark 27. One may wonder whether previous keys that involve those $z_{t_{k,\lambda}}$ could leak some information and contradict the above argument. Let us focus on the leaf λ associated to the attribute p, and so the information one could get about z_p when $\mathbf{c}_p$ is not part of the challenge ciphertext. At least, this argument holds for the first key generation, when we are in the first sequence of games, in $\mathbf{G}_{2.k.2}$ with $k = 1$: z_p is only used in $\mathbf{c}_p$, that is not revealed to the adversary, and so $s_{1,\lambda}/z_p$ does not leak any information about $s_{1,\lambda}$. And this is the same for all the leaves associated to missing attributes. Then $s_{1,0}$ can definitely be replaced by a random and independent $r_{1,0}$: which is the current game $\mathbf{G}_{2.k.3}$ for $k = 1$. When we are in $\mathbf{G}_{2.k.2}$ for $k = 2$, the adversary may now have some information about $s_{1,\lambda}/z_p$ and $s_{2,\lambda}/z_p$, but no information about $s_{1,0}$ that has already been replaced by a random $r_{1,0}$, which makes $s_{1,\lambda}$ unpredictable, and so no additional information leaks about z_p: $s_{2,\lambda}$ is unpredictable. Again, the same argument holds for all the leaves associated to missing attributes: $s_{2,0}$ can also be replaced by a random and independent $r_{2,0}$.
This is the reason of this hybrid sequence of game: if we would have first introduced the z_p in all the keys, it would not have been possible to replace all the $s_{\ell,0}$ by $r_{\ell,0}$ in the end. This is only true when all the previous keys have already been modified.
One can thus modify the key generation algorithm for the k-th key, with an independent $r_{k,0} \stackrel{\$}{\leftarrow} \mathbb{Z}_q$:

$$\mathbf{c}_0 = (\omega, \tau, \xi)_{\mathbb{B}} \qquad \mathbf{c}_t = (\sigma_t(1,t), \omega, \tau, 0, \tau z_t)_{\mathbb{D}}$$
$$\mathbf{k}^*_{k,0} = (a_{k,0}, r_{k,0}, 1)_{\mathbb{B}^*} \qquad \mathbf{k}^*_{k,\lambda} = (\pi_{k,\lambda}(t_{k,\lambda}, -1), a_{k,\lambda}, 0, 0, s_{k,\lambda}/z_{t_{k,\lambda}})_{\mathbb{D}^*}$$

This concludes this sequence of sub-games with, for each k,

$$\mathsf{Adv}_{2.k.0} - \mathsf{Adv}_{2.k.3} \leq 2P^2 \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t) + (3P+1) \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t).$$

Hence, globally, we have

$$\mathsf{Adv}_2 - \mathsf{Adv}_3 \leq 2KP^2 \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t) + (3P+1)K \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t).$$

Game $\mathbf{G}_4$: In this game, one chooses a random θ to define the matrices

$$B = \begin{pmatrix} 1 & -\theta \\ 0 & 1 \end{pmatrix}_{2,3} \qquad B' = \begin{pmatrix} 1 & 0 \\ \theta & 1 \end{pmatrix}_{2,3} \qquad \mathbb{B} = B \cdot \mathbb{U} \qquad \mathbb{B}^* = B' \cdot \mathbb{U}^*$$

which only modifies $\mathbf{b}_2$, which is hidden, and $\mathbf{b}^*_3$, which is kept secret:

$$\mathbf{c}_0 = (\omega, \tau, \xi)_{\mathbb{U}} = (\omega, \tau, \tau\theta + \xi)_{\mathbb{B}} = (\omega, \tau, \xi'')_{\mathbb{B}}$$
$$\mathbf{k}^*_{\ell,0} = (a_{\ell,0}, r_{\ell,0}, 1)_{\mathbb{U}^*} = (a_{\ell,0}, r'_{\ell,0}, 1)_{\mathbb{D}^*}$$

As a consequence, any value for θ can be used, without impacting the view of the adversary, as $r'_{\ell,0}$ is indeed independent of the other variables. In this last game, a random value ξ'' is used in the ciphertext, whereas $K_0 = g_t^{\xi}$ and $K_1 = g_t^{\xi'}$: the advantage of any adversary is 0 in this last game.

If we combine all the steps:

$$\begin{aligned}
\mathsf{Adv}_0 &= \mathsf{Adv}_0 - \mathsf{Adv}_4 \\
&\leq \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t) + \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t) + 2KP^2 \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t) + (3P+1)K \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t) \\
&\leq 2(KP^2+1) \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t) + (3P+1)K \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t)
\end{aligned}$$

We now present the sub-sequence of games for proving the gap from the above $\mathbf{G}_{2.k.1}$ to $\mathbf{G}_{2.k.2}$.

$$\begin{array}{lllllllll}
 & & & \mathbf{c}_0 = (& \omega & \tau & \xi) & \mathbf{h}^*_0 = (& \delta \quad \rho \quad 0) \\
\mathbf{G}_{2.k.1.p.0} & \multicolumn{8}{l}{\text{Hybrid game for } \mathbf{G}_{2.k.1}, \text{ with } 1 \leq p \leq P+1 \text{ (from Figure 11)}} \\
 & & \mathbf{c}_t = (& \sigma_t(1,t) & \omega & | \ \tau & 0 & \tau z_t &) \\
 & t < p & \mathbf{h}^*_t = (& \pi_t(t,-1) & \delta & | \ 0 & 0 & \rho/z_t &) \\
 & t \geq p & \mathbf{h}^*_t = (& \pi_t(t,-1) & \delta & | \ \rho & 0 & 0 &) \\
\mathbf{G}_{2.k.1.p.1} & \multicolumn{8}{l}{\text{Formal basis change, on } (\mathbb{D},\mathbb{D}^*)_{4,5}, \text{ to duplicate } \tau} \\
 & & \mathbf{c}_t = (& \sigma_t(1,t) & \omega & | \ \tau & \tau & \tau z_t &) \\
\mathbf{G}_{2.k.1.p.2} & \multicolumn{8}{l}{\textsf{Swap-Ind} \text{ Property, on } (\mathbb{D}^*,\mathbb{D})_{2,4,5}, \text{ for } 0 \text{ and } \rho \text{ in } \mathbf{h}^*_p \text{ only}} \\
 & & \mathbf{c}_t = (& \sigma_t(1,t) & \omega & | \ \tau & \tau & \tau z_t &) \\
 & & \mathbf{h}^*_p = (& \pi_p(p,-1) & \delta & | \ 0 & \rho & 0 &) \\
\mathbf{G}_{2.k.1.p.3} & \multicolumn{8}{l}{\textsf{Index-Ind} \text{ Property, on } (\mathbb{D},\mathbb{D}^*)_{1,2,5}, \text{ between } \tau \text{ and } \tau z_t/z_p} \\
 & & \mathbf{c}_p = (& \sigma_p(1,p) & \omega & | \ \tau & \tau & \tau z_p &) \\
 & t \neq p & \mathbf{c}_t = (& \sigma_t(1,t) & \omega & | \ \tau & \tau z_t/z_p & \tau z_t &) \\
\mathbf{G}_{2.k.1.p.4} & \multicolumn{8}{l}{\text{Formal basis change, on } (\mathbb{D},\mathbb{D}^*)_{5,6}, \text{ to cancel } \tau} \\
 & & \mathbf{c}_t = (& \sigma_t(1,t) & \omega & | \ \tau & 0 & \tau z_t &) \\
 & & \mathbf{h}^*_p = (& \pi_p(p,-1) & \delta & | \ 0 & \alpha & \rho/z_p &) \\
\mathbf{G}_{2.k.1.p.5} & \multicolumn{8}{l}{\textsf{SubSpace-Ind} \text{ Property, on } (\mathbb{D}^*,\mathbb{D})_{2,5}, \text{ between } \alpha \text{ and } 0} \\
 & & \mathbf{c}_t = (& \sigma_t(1,t) & \omega & | \ \tau & 0 & \tau z_t &) \\
 & t < p & \mathbf{h}^*_t = (& \pi_t(t,-1) & \delta & | \ 0 & 0 & \rho/z_t &) \\
 & & \mathbf{h}^*_p = (& \pi_p(p,-1) & \delta & | \ 0 & 0 & \rho/z_p &) \\
 & t > p & \mathbf{h}^*_t = (& \pi_t(t,-1) & \delta & | \ \rho & 0 & 0 &)
\end{array}$$

Fig. 12: Sequence of sub-games on the P attributes for the IND-security proof of our $\mathsf{KP\text{-}ABE}$, where $\mathbf{k}^*_{\ell,0} = (a_{\ell,0}, 0, 1)_{\mathbb{B}^*} + s_{\ell,0} \cdot \mathbf{h}^*_0$ and $\mathbf{k}^*_{\ell,\lambda} = (\varPi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, 0, 0)_{\mathbb{D}^*} + s_{\ell,\lambda} \cdot \mathbf{h}^*_{t_{k,\lambda}}$, for all the leaves λ of all the keys ℓ, with $\mathbf{h}^*_0 = (\delta, \rho, 0)_{\mathbb{B}^*}$ and $\mathbf{h}^*_0 = (\pi_t(t,-1), \delta, \rho, 0, 0)_{\mathbb{D}^*}$ for all the possible attributes t. We only make the latter $(\mathbf{h}^*_0, (\mathbf{h}^*_t)_t)$ to evolve along this sequence.

We still focus on the challenge ciphertext $(\mathbf{c}_0, (\mathbf{c}_t))$ and the k-th key we will denote, for the sake of clarity, as

$$\begin{aligned}
\mathbf{k}^*_{k,0} &= (a_0, 0, 1)_{\mathbb{B}^*} + s_0 \cdot \mathbf{h}^*_0 \\
\mathbf{k}^*_{k,\lambda} &= (\varPi_{k,\lambda}(t_{k,\lambda}, -1), a_\lambda, 0, 0, 0)_{\mathbb{D}^*} + s_\lambda \cdot \mathbf{h}^*_{t_{k,\lambda}}
\end{aligned}$$

where $\mathbf{h}^*_0 = (\delta, \rho, 0)_{\mathbb{B}^*}$ and $\mathbf{h}^*_t = (\pi_t(t,-1), \delta, \rho, 0, 0)_{\mathbb{D}^*}$ for all the possible attributes. This corresponds to

$$\begin{aligned}
a_{k,0} &= a_0 + \delta \cdot s_0 & a_{k,\lambda} &= a_\lambda + \delta \cdot s_\lambda \\
s_{k,0} &= \rho \cdot s_0 & s_{k,\lambda} &= \rho \cdot s_\lambda \\
 & & \pi_{k,\lambda} &= \varPi_{k,\lambda} + s_\lambda \cdot \pi_{t_{k,\lambda}}
\end{aligned}$$

All the other keys will be generated using the basis vectors: we stress that they all have a zero 5-th component, then $\mathbf{d}^*_5$ will not be needed. In the new hybrid game, the critical point will be the p-th attribute, where, when $p = 1$, $\mathbf{G}_{2.k.1.p.0}$ is exactly the above Game $\mathbf{G}_{2.k.1}$, and when $p = P+1$ this is the above Game $\mathbf{G}_{2.k.2}$. And it will be clear, for any p, that $\mathbf{G}_{2.k.1.p.5} = \mathbf{G}_{2.k.1.p+1.0}$: with random $\omega, \tau, \xi, \xi', \delta, \rho, (z_t), (\sigma_t), (\pi_t) \xleftarrow{\$} \mathbb{Z}_q$,

Game $\mathbf{G}_{2.k.1.p.0}$: One defines the hybrid game for p:

$$\begin{aligned} \mathbf{c}_0 &= (\omega, \tau, \xi)_{\mathbb{B}} & \mathbf{c}_t &= (\sigma_t(1,t), \omega, \tau, 0, \tau z_t)_{\mathbb{D}} & \\ \mathbf{h}_0^* &= (\delta, \rho, 0)_{\mathbb{B}^*} & \mathbf{h}_t^* &= (\pi_t(t,-1), \delta, 0, 0, \rho/z_t)_{\mathbb{D}^*} & t < p \\ & & \mathbf{h}_t^* &= (\pi_t(t,-1), \delta, \rho, 0, 0)_{\mathbb{D}^*} & t \geq p \end{aligned}$$

Game $\mathbf{G}_{2.k.1.p.1}$: One defines the matrices

$$D = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}_{4,5} \qquad D' = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}_{4,5} \qquad \mathbb{D} = D \cdot \mathbb{V} \qquad \mathbb{D}^* = D' \cdot \mathbb{V}^*$$

which modifies the hidden vectors $\mathbf{d}_4$ and $\mathbf{d}_5^*$, and so are not in the view of the adversary:

$$\begin{aligned} \mathbf{c}_t &= (\sigma_t(1,t), \omega, \tau, 0, \tau z_t)_{\mathbb{V}} = (\sigma_t(1,t), \omega, \tau, \tau, \tau z_t)_{\mathbb{D}} & \\ \mathbf{h}_t^* &= (\pi_t(t,-1), \delta, 0, 0, \rho/z_t)_{\mathbb{V}^*} = (\pi_t(t,-1), \delta, 0, 0, \rho/z_t)_{\mathbb{D}^*} & t < p \\ \mathbf{h}_t^* &= (\pi_t(t,-1), \delta, \rho, 0, 0)_{\mathbb{V}^*} = (\pi_t(t,-1), \delta, \rho, 0, 0)_{\mathbb{D}^*} & t \geq p \end{aligned}$$

For all the other keys, as the 5-th component is 0, the writing in basis $\mathbb{V}^*$ is the same in basis $\mathbb{D}^*$. Hence, the perfect indistinguishability between the two games: $\mathsf{Adv}_{2.k.1.p.1} = \mathsf{Adv}_{2.k.1.p.0}$.

Game $\mathbf{G}_{2.k.1.p.2}$: We apply the $\mathsf{Swap\text{-}Ind}$ property from Theorem 21, on $(\mathbb{D}^*, \mathbb{D})_{2,4,5}$: Indeed, we can consider a triple $(a \cdot G_2, b \cdot G_2, c \cdot G_2)$, where $c = ab + \theta \bmod q$ with either $\theta = 0$ or $\theta = \rho$, which are indistinguishable situations under the DSDH assumption. Let us assume we start from random dual orthogonal bases $(\mathbb{B}, \mathbb{B}^*)$ and $(\mathbb{V}, \mathbb{V}^*)$ of size 3 and 6 respectively. Then we define the matrices

$$D' = \begin{pmatrix} 1 & a & -a \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}_{2,4,5} \qquad D = \begin{pmatrix} 1 & 0 & 0 \\ -a & 1 & 0 \\ a & 0 & 1 \end{pmatrix}_{2,4,5} \qquad \mathbb{D}^* = D' \cdot \mathbb{V}^* \qquad \mathbb{D} = D \cdot \mathbb{V}$$

Note that we can compute all the basis vectors excepted $\mathbf{d}_4, \mathbf{d}_5$, but we define the ciphertext on the original basis $\mathbb{V}$:

$$\begin{aligned} \mathbf{c}_t &= (\sigma_t(1,t), \omega, \tau, \tau, \tau z_t)_{\mathbb{V}} = (\sigma_t, \sigma_t t + a\tau - a\tau, \omega, \tau, \tau, \tau z_t)_{\mathbb{D}} & \\ &= (\sigma_t(1,t), \omega, \tau, \tau, \tau z_t)_{\mathbb{D}} & \\ \mathbf{h}_t^* &= (\pi_t(t,-1), \delta, 0, 0, \rho/z_t)_{\mathbb{D}^*} & t < p \\ \mathbf{h}_p^* &= (\pi_p(p,-1), \delta, \rho, 0, 0)_{\mathbb{D}^*} + (b(p,-1), 0, -c, c, 0)_{\mathbb{V}^*} & \\ &= (\pi_p(p,-1), \delta, \rho, 0, 0)_{\mathbb{D}^*} + (b(p,-1), 0, ab - c, -ab + c, 0)_{\mathbb{D}^*} & \\ &= (\pi_p(p,-1), \delta, \rho - \theta, \theta, 0)_{\mathbb{D}^*} & \\ \mathbf{h}_t^* &= (\pi_t(t,-1), \delta, \rho, 0, 0)_{\mathbb{D}^*} & t > p \end{aligned}$$

With $\theta = 0$, this is as in the previous game, with $\theta = \rho$, this is the current game: $\mathsf{Adv}_{2.k.1.p.1} - \mathsf{Adv}_{2.k.1.p.2} \leq 2 \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t)$.

Game $\mathbf{G}_{2.k.1.p.3}$: We keep the τ value (at the 5-th hidden position) in the ciphertext for the p-th attribute only, and replace all the other values by $\tau z_t / z_p$:

$$\begin{aligned} \mathbf{c}_p &= (\sigma_t(1,t), \omega, \tau, \tau, \tau z_t)_{\mathbb{D}} & \\ \mathbf{c}_t &= (\sigma_t(1,t), \omega, \tau, \tau z_t / z_p, \tau z_t)_{\mathbb{D}} & t \neq p \end{aligned}$$

To show this is possible without impacting the other vectors, we use the $\mathsf{Index\text{-}Ind}$ property from Theorem 23, but in another level of sequence of hybrid games, for $\gamma \in \{1, \dots, P\} \backslash \{p\}$:

Game $\mathbf{G}_{2.k.1.p.2.\gamma}$: We consider

$$\begin{aligned}
\mathbf{c}_p &= (\sigma_p(1,p), \omega, \tau, \tau, \tau z_p)_{\mathbb{D}} && \\
\mathbf{c}_t &= (\sigma_t(1,t), \omega, \tau, \tau z_t/z_p, \tau z_t)_{\mathbb{D}} && p \neq t < \gamma \\
\mathbf{c}_t &= (\sigma_t(1,t), \omega, \tau, \tau, \tau z_t)_{\mathbb{D}} && t \geq \gamma \\
\mathbf{h}^*_t &= (\pi_t(t,-1), \delta, 0, 0, \rho/z_t)_{\mathbb{D}^*} && t < p \\
\mathbf{h}^*_p &= (\pi_p(p,-1), \delta, 0, \rho, 0)_{\mathbb{D}^*} && \\
\mathbf{h}^*_t &= (\pi_t(t,-1), \delta, \rho, 0, 0)_{\mathbb{D}^*} && t > p
\end{aligned}$$

When $\gamma = 1$, this is the previous game: $\mathbf{G}_{2.k.1.p.2.1} = \mathbf{G}_{2.k.1.p.2}$, whereas with $\gamma = P+1$, this is the current game: $\mathbf{G}_{2.k.1.p.2.P+1} = \mathbf{G}_{2.k.1.p.3}$.
For any $\gamma \in \{1, \ldots, P\}$, we consider a triple $(a \cdot G_1, b \cdot G_1, c \cdot G_1)$, where $c = ab + \zeta \bmod q$, with either $\zeta = 0$ or $\zeta = \tau(z_\gamma/z_p - 1)$, which are indistinguishable situations under the DSDH assumption. We define the matrices

$$D = \frac{1}{p-\gamma} \times \begin{pmatrix} p & -\gamma & ap \\ -1 & 1 & -a \\ 0 & 0 & p-\gamma \end{pmatrix}_{1,2,5} \qquad D' = \begin{pmatrix} 1 & 1 & 0 \\ \gamma & p & 0 \\ -a & 0 & 1 \end{pmatrix}_{1,2,5}$$

and then $\mathbb{D} = D \cdot \mathbb{V}$, $\mathbb{D}^* = D' \cdot \mathbb{V}^*$: we cannot compute $\mathbf{d}^*_5$, but the components on this vector are all 0 excepted for $\mathbf{h}^*_p$ we will define in $\mathbb{V}^*$:

$$\begin{aligned}
\mathbf{c}_p &= (\sigma_p(1,p), \omega, \tau, \tau, \tau z_p)_{\mathbb{D}} && \\
\mathbf{c}_\gamma &= (b, 0, \omega, \tau, \tau + c, \tau z_\gamma)_{\mathbb{V}} = (b, b\gamma, \omega, \tau, \tau + c - ab, \tau z_\gamma)_{\mathbb{D}} && \\
&= (b(1,\gamma), \omega, \tau, \tau + \zeta, \tau z_\gamma)_{\mathbb{D}} && \\
\mathbf{c}_t &= (\sigma_t(1,t), \omega, \tau, \tau z_t/z_p, \tau z_t)_{\mathbb{D}} && p \neq t < \gamma \\
\mathbf{c}_t &= (\sigma_t(1,t), \omega, \tau, \tau, \tau z_t)_{\mathbb{D}} && t > \gamma \\
\mathbf{h}^*_t &= (\pi_t(t,-1), \delta, 0, 0, \rho/z_t)_{\mathbb{D}^*} && t < p \\
\mathbf{h}^*_p &= ((p-\gamma) \cdot (\pi, 0), \delta, 0, \rho, 0)_{\mathbb{V}^*} && \\
&= (p \cdot \pi + ap\rho, -\pi - a\rho, \delta, 0, \rho, 0)_{\mathbb{D}^*} && \\
&= ((\pi + a\rho) \cdot (p, -1), \delta, 0, \rho, 0)_{\mathbb{D}^*} && \\
\mathbf{h}^*_t &= (\pi_t(t,-1), \delta, \rho, 0, 0)_{\mathbb{D}^*} && t > p
\end{aligned}$$

which is the hybrid game with $\pi_p = \pi + a\rho$ and the 5-th component of $\mathbf{c}_\gamma$ is $\tau + \zeta$, which is either τ when $\zeta = 0$, and thus the game $\mathbf{G}_{2.k.1.p.2.\gamma}$ or $\tau z_\gamma/z_p$ when $\zeta = \tau z_\gamma/z_p - \tau$, which is $\mathbf{G}_{2.k.1.p.2.\gamma+1}$: hence, the distance between two consecutive games is bounded by $\mathsf{Adv}^{\mathsf{dsdh}}_{\mathbb{G}_1}(t)$.
Hence, we have $\mathsf{Adv}_{2.k.1.p.2} - \mathsf{Adv}_{2.k.1.p.3} \leq 2P \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t)$.

Game $\mathbf{G}_{2.k.1.p.4}$: We can now insert $1/z_p$ in the p-th last component, and then make some cleaning with the matrices, for $\alpha \stackrel{\$}{\leftarrow} \mathbb{Z}_q^*$

$$D = \begin{pmatrix} \alpha/\rho & 0 \\ 1/z_p & 1 \end{pmatrix}_{5,6} \qquad D' = \begin{pmatrix} \rho/\alpha & -\rho/\alpha z_p \\ 0 & 1 \end{pmatrix}_{5,6}$$

and then $\mathbb{D} = D \cdot \mathbb{V}$, $\mathbb{D}^* = D' \cdot \mathbb{V}^*$. As the four vectors $\mathbf{d}_5, \mathbf{d}_6$ and $\mathbf{d}^*_5, \mathbf{d}^*_6$ are hidden, the modifications will not impact the view of the adversary. This consists in applying successively the matrices :

$$D = \begin{pmatrix} 1/z_p & 0 \\ 0 & 1 \end{pmatrix}_{5,6} \qquad D = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}_{5,6} \qquad D = \begin{pmatrix} \alpha z_p/\rho & 0 \\ 0 & 1 \end{pmatrix}_{5,6}$$

Then, working in $(\mathbb{V}, \mathbb{V}^*)$ gives, in $(\mathbb{D}, \mathbb{D}^*)$:

$$\begin{aligned}
\mathbf{c}_p &= (\sigma_p(1,p), \omega, \tau, \tau, \tau z_p)_{\mathbb{V}} = (\sigma_p(1,p), \omega, \tau, 0, \tau z_p)_{\mathbb{D}} \\
\mathbf{c}_t &= (\sigma_t(1,t), \omega, \tau, \tau z_t/z_p, \tau z_t)_{\mathbb{V}} \\
&= (\sigma_t(1,t), \omega, \tau, (\tau z_t/z_p - \tau z_t/z_p) \cdot \rho/\alpha, \tau z_t)_{\mathbb{D}} && t \neq p \\
&= (\sigma_t(1,t), \omega, \tau, 0, \tau z_t)_{\mathbb{D}} \\
\mathbf{h}^*_t &= (\pi_t(t,-1), \delta, 0, 0, \rho/z_t)_{\mathbb{V}^*} = (\pi_t(t,-1), \delta, 0, 0, \rho/z_t)_{\mathbb{D}^*} && t < p \\
\mathbf{h}^*_p &= (\pi_p(p,-1), \delta, 0, \rho, 0)_{\mathbb{V}^*} = (\pi_p(p,-1), \delta, 0, \alpha, \rho/z_p)_{\mathbb{D}^*} \\
\mathbf{h}^*_t &= (\pi_t(t,-1), \delta, \rho, 0, 0)_{\mathbb{V}^*} = (\pi_t(t,-1), \delta, \rho, 0, 0)_{\mathbb{V}^*} && t > p
\end{aligned}$$

We stress again that for all the other keys, as the 5-th component is 0, the writing in basis $\mathbb{V}^*$ is the same in basis $\mathbb{D}^*$. Hence, the perfect indistinguishability between the two games: $\mathsf{Adv}_{2.k.1.p.4} = \mathsf{Adv}_{2.k.1.p.3}$.

Game $\mathbf{G}_{2.k.1.p.5}$: We can now remove the α value in the p-th element of the key: We can consider a triple $(a \cdot G_2, b \cdot G_2, c \cdot G_2)$, where $c = ab + \alpha \bmod q$, with either $\alpha = 0$ or $\alpha \stackrel{\$}{\leftarrow} \mathbb{Z}_q^*$, which are indistinguishable situations under the DDH assumption. We define the matrices

$$D' = \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}_{2,5} \qquad D = \begin{pmatrix} 1 & 0 \\ -a & 1 \end{pmatrix}_{2,5}$$

and then $\mathbb{D} = D \cdot \mathbb{V}$, $\mathbb{D}^* = D' \cdot \mathbb{V}^*$: we cannot compute $\mathbf{d}_5$, but the components on this vector are all 0:

$$\begin{aligned}
\mathbf{c}_p &= (\sigma_t(1,t), \omega, \tau, 0, \tau z_t)_{\mathbb{D}} \\
\mathbf{c}_t &= (\sigma_t(1,t), \omega, \tau, 0, \tau z_t)_{\mathbb{D}} && t \neq p \\
\mathbf{h}^*_t &= (\pi_t(t,-1), \delta, 0, 0, \rho/z_t)_{\mathbb{D}^*} && t < p \\
\mathbf{h}^*_p &= (-b(p,-1), \delta, 0, c, \rho/z_p)_{\mathbb{V}^*} = (-b(p,-1), \delta, 0, c - ab, \rho/z_p)_{\mathbb{D}^*} \\
&= (-b(p,-1), \delta, 0, \alpha, \rho/z_p)_{\mathbb{D}^*} \\
\mathbf{h}^*_t &= (\pi_t(t,-1), \delta, \rho, 0, 0)_{\mathbb{D}^*} && t > p
\end{aligned}$$

which is the either the previous game when $\alpha \neq 0$ or the current game with $\alpha = 0$, where $\pi_p = -b$: $\mathsf{Adv}_{2.k.1.p.4} - \mathsf{Adv}_{2.k.1.p.5} \leq \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t)$.

C Security Proofs of our SA-KP-ABE Scheme

C.1 Proof of Theorem 10 – Del-IND-Security for Encaps

Proof. We will proceed to prove this by a succession of games. At some point, our game will be in the same state as Game $\mathbf{G}_0$ in the proof of IND for the KP-ABE scheme, in the appendix B.3, which allows us to conclude.

Game $\mathbf{G}_0$: The first game is the real game, where the simulator honestly runs the setup, with $\mathsf{PK} = \{(\mathbf{b}_1, \mathbf{b}_3, \mathbf{b}^*_1), (\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \mathbf{d}^*_1, \mathbf{d}^*_2, \mathbf{d}^*_3)\}$, $\mathsf{SK} = \{\mathbf{d}_7\}$, and $\mathsf{MK} = \{\mathbf{b}^*_3, \mathbf{d}^*_7\}$, from random dual orthogonal bases.

$\mathsf{OKeyGen}(\tilde{\mathcal{T}}_\ell)$**:** The adversary is allowed to issue KeyGen-queries on an access-tree $\tilde{\mathcal{T}}_\ell$ (for the ℓ-th query), for which the simulator chooses a random scalar $a_{\ell,0} \stackrel{\$}{\leftarrow} \mathbb{Z}_q$ and a random $a_{\ell,0}$-labeling $(a_{\ell,\lambda})_\lambda$ of the access-tree $\tilde{\mathcal{T}}_\ell$, and builds the key:

$$\mathbf{k}^*_{\ell,0} = (a_{\ell,0}, 0, 1)_{\mathbb{B}^*} \qquad \mathbf{k}^*_{\ell,\lambda} = (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, 0, 0, r_{\ell,\lambda})_{\mathbb{D}^*}$$

for all the leaves λ, where $t_{\ell,\lambda} = A(\lambda)$ in $\mathcal{T}_\ell$, $\pi_{\ell,\lambda} \stackrel{\$}{\leftarrow} \mathbb{Z}_q$ and $r_{\ell,\lambda} \stackrel{\$}{\leftarrow} \mathbb{Z}_q^*$ if λ is an active leave or $r_{\ell,\lambda} = 0$ if it is passive. The decryption key $\mathsf{dk}_\ell = (\mathbf{k}^*_{\ell,0}, (\mathbf{k}^*_{\ell,\lambda})_\lambda)$ is kept private, and will be used for delegation queries;

$\mathsf{ODelegate}(\tilde{\mathcal{T}}, \tilde{\mathcal{T}}')$: The adversary is allowed to issue Delegate-queries for an access-tree $\tilde{\mathcal{T}}'$, of an already queried decryption key with access-tree $\tilde{\mathcal{T}} = \tilde{\mathcal{T}}_\ell$, with the only condition that $\tilde{\mathcal{T}}' \leq \tilde{\mathcal{T}}$. From $\mathsf{dk}_\ell = (k_0^*, (k_\lambda^*)_\lambda)$, for $\lambda \in \mathcal{L}$, then the simulator computes the delegated key as:

$$\mathbf{k}_0'^* = \mathbf{k}_0^* + (a_0', 0, 0)_{\mathbb{B}^*} \qquad \mathbf{k}_\lambda'^* = \mathbf{k}_\lambda^* + (\pi_\lambda'(t_\lambda, -1), a_\lambda', 0, 0, 0, 0)_{\mathbb{D}^*}, \forall \lambda \in \mathcal{L}',$$

where $\mathbf{k}_\lambda^* = (0,0,0,0,0,0,0)_{\mathbb{D}^*}$ if λ was not in $\mathcal{L}$, and $a_0' \stackrel{\$}{\leftarrow} \mathbb{Z}_q$ and $(a_\lambda')_\lambda$ is an a_0'-labeling of $\mathcal{T}'$.

$\mathsf{RoREncaps}(\Gamma_v, \Gamma_i = \emptyset)$: On the unique query on a set of attributes $\Gamma = \Gamma_v$, the simulator chooses random scalars $\omega, \xi, \xi' \stackrel{\$}{\leftarrow} \mathbb{Z}_q$. It then sets $K_0 = g_t^\xi$ and $K_1 = g_t^{\xi'}$. It generates the ciphertext $C = (\mathbf{c}_0, (\mathbf{c}_t)_{t \in \Gamma})$ where

$$\mathbf{c}_0 = (\omega, 0, \xi)_{\mathbb{B}} \qquad \mathbf{c}_t = (\sigma_t(1, t), \omega, 0, 0, 0, 0)_{\mathbb{D}}$$

for all the attributes $t \in \Gamma$ and $\sigma_t \stackrel{\$}{\leftarrow} \mathbb{Z}_q$. According to the real or random game (bit $b \stackrel{\$}{\leftarrow} \{0,1\}$), one outputs (K_b, C).

From the adversary's guess b' for b, if for some $\tilde{\mathcal{T}}'$ asked as a delegation-query, $\tilde{\mathcal{T}}'(\Gamma_v, \Gamma_i) = 1$, then $\beta \stackrel{\$}{\leftarrow} \{0,1\}$, otherwise $\beta = b'$. We denote $\mathsf{Adv}_0 = \Pr[\beta = 1 | b = 1] - \Pr[\beta = 1 | b = 0]$. We stress that in this game, we deal with delegation queries, but only want to show they do not help to break indistinguishability of the encapsulated keys with the official Encaps algorithm, and not the private Encaps* one. Hence, $\Gamma_i = \emptyset$ in the challenge ciphertext.

Game $\mathbf{G}_1$: We now show it can be reduced to Game $\mathbf{G}_0$ from the IND security game on the KP-ABE, in the proof provided in the appendix B.3. The challenge ciphertext is already exactly the same, as we only consider Encaps. But we have to emulate the key-generation and key-delegation oracles OKeyGen and ODelegate using only the key-generation oracle from Game $\mathbf{G}_0$ in the proof provided in the appendix B.3, we denote OKeyGen′, as it only partially generates our new keys, with a 7-th coordinate $r_{\ell,\lambda}$. First, we instantiate a list Λ.

$\mathsf{OKeyGen}(\tilde{\mathcal{T}}_\ell)$. The simulator calls the oracle $\mathsf{OKeyGen}'(\tilde{\mathcal{T}}_\ell)$, and chooses $r_{\ell,\lambda} \stackrel{\$}{\leftarrow} \mathbb{Z}_q^*$ or sets $r_{\ell,\lambda} \leftarrow 0$ according to whether $\lambda \in \mathcal{L}_a$ or $\lambda \in \mathcal{L}_p$. It then adds the last component $r_{\ell,\lambda}$ on every $\mathbf{k}_{\ell,\lambda}^*$ using $\mathbf{d}_7^*$ which is known to the simulator. Finally, it updates Λ with a new entry $\Lambda_\ell = (r_{\ell,\lambda})_\lambda$;

$\mathsf{ODelegate}(\tilde{\mathcal{T}}_\ell, \tilde{\mathcal{T}}')$. The simulator calls the oracle $\mathsf{OKeyGen}'(\mathcal{T}')$ to get the decryption key dk. As already noted, in the KP-ABE, a delegated key is indistinguishable from a fresh key. Then, we pick the entry $r_{\ell,\lambda}$ from Λ_ℓ, to the last component $r_{\ell,\lambda}$ on every $\mathbf{k}_\lambda^*$ using $\mathbf{d}_7^*$ which is known to the simulator. We stress that for any new leaf, not present in $\tilde{\mathcal{T}}_\ell$ is necessarily passive in the delegated tree $\tilde{\mathcal{T}}'$. So, if a leaf is not in Λ_ℓ, $r_{\ell,\lambda} = 0$.

In this new game, we are exactly using the security game from the IND security on the KP-ABE, and simulating the 7-th component using $\mathbf{d}_7^*$. As this component does not change nor intervene at all in any of the games from the proof in the appendix B.3, and this is exactly the same situation as in Game $\mathbf{G}_0$ in that proof, we conclude by following those security games, which leads to the adversary having zero advantage in the last game.

We stress that this simulation of ODelegate will be used in all the following proofs: a delegated key is identical to a fresh key, excepted the common $r_{\ell,\lambda}$ for keys delegated from the same original key.

C.2 Proof of Theorem 11 – Del-IND-Security for Encaps*

Proof. The proof will proceed by games, with exactly the same sequence as in the previous proof following the IND-security proof of the KP-ABE in the appendix B.3, except the RoREncaps-challenge that allows non-empty Γ_i. For the same reason, the OEncaps-queries on pairs (Γ_v, Γ_i), with $\mathit{\Gamma}_i \neq \emptyset$ can be simulated. Indeed, as above, everything on the 7-th component can be done independently, knowing both $\mathbf{d}_7$ and $\mathbf{d}_7^*$, as these vectors will be known to the simulator, almost all the time, excepted in some specific gaps. In theses cases, we will have to make sure how to simulate the OEncaps ciphertexts. As explained in the proof, Section C.1, we can simulate ODelegate-queries as OKeyGen-queries, since a delegated key is identical to a fresh key, excepted the common $r_{\ell,\lambda}$ for keys delegated from the same original key. We thus just have to take care about the way we choose $r_{\ell,\lambda}$. This will be critical in $\mathbf{G}_{2.k.2.3.p.6}$, and it will be correct as the same constraint will be applied to $y_{\ell,\lambda}$ introduced in $\mathbf{G}_{2.k.2.2}$

As in the IND-security proof of the KP-ABE, the idea of the sequence is to introduce an additional labeling $(s_{\ell,0}, (s_{\ell,\lambda})_\lambda)$ in each ℓ-th key (in $\mathbf{G}_{2.k.1}$, from Figure 11), where each label is masked by a random z_t for each attribute t (in $\mathbf{G}_{2.k.2}$).

However, in order to go to game $\mathbf{G}_{2.k.3}$, one cannot directly conclude that $s_{k,0}$ is independent from the view of the adversary: we only know $\tilde{\mathcal{T}}_k(\mathit{\Gamma}_v, \mathit{\Gamma}_i) = 0$, but not necessarily $\mathcal{T}_k(\mathit{\Gamma}_v \cup \mathit{\Gamma}_i) = 0$, as in the previous proof.

$\mathbf{G}_{2.k.2.0}$ Intermediate sequence from $\mathbf{G}_{2.k.2}$ (from Figure 11)

$$\begin{array}{lrcllllllll}
 & \mathbf{c}_t & = (& \sigma_t(1,t) & \omega & | & \tau & 0 & \tau z_t & u_t &) \\
\ell < k & \mathbf{k}^*_{\ell,0} & = (& a_{\ell,0} & r_{\ell,0} & 1\ ) & & & & & \\
 & \mathbf{k}^*_{\ell,\lambda} & = (& \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) & a_{\ell,\lambda} & | & 0 & 0 & s'_{\ell,\lambda}/z_{t_{\ell,\lambda}} & r_{\ell,\lambda} &) \\
\ell = k & \mathbf{k}^*_{k,0} & = (& a_{k,0} & s_{k,0} & 1\ ) & & & & & \\
 & \mathbf{k}^*_{k,\lambda} & = (& \pi_{k,\lambda}(t_{k,\lambda},-1) & a_{k,\lambda} & | & 0 & 0 & s_{k,\lambda}/z_{t_{k,\lambda}} & r_{k,\lambda} &) \\
\ell > k & \mathbf{k}^*_{\ell,0} & = (& a_{\ell,0} & 0 & 1\ ) & & & & & \\
 & \mathbf{k}^*_{\ell,\lambda} & = (& \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) & a_{\ell,\lambda} & | & 0 & 0 & 0 & r_{\ell,\lambda} &)
\end{array}$$

$s'_{\ell,\lambda}$ is either the label $s_{\ell,\lambda}$ when $r_{\ell,\lambda} \cdot u_{t_{\ell,\lambda}} = 0$, or a random scalar in $\mathbb{Z}_q$ otherwise

$\mathbf{G}_{2.k.2.1}$ SubSpace-Ind Property, on $(\mathbb{D}, \mathbb{D}^*)_{4,5}$, between τ and 0

$$\mathbf{c}_t = (\quad \sigma_t(1,t) \quad \omega \quad | \quad 0 \quad 0 \quad \tau z_t \quad u_t \quad)$$

$\mathbf{G}_{2.k.2.2}$ SubSpace-Ind Property, on $(\mathbb{D}^*, \mathbb{D})_{2,4}$, between 0 and $y_{\ell,\lambda}$

$$\begin{array}{rcllllllll}
\mathbf{k}^*_{k,0} & = (& a_{k,0} & s_{k,0} & 1\ ) & & & & & \\
\mathbf{k}^*_{k,\lambda} & = (& \pi_{k,\lambda}(t_{k,\lambda},-1) & a_{k,\lambda} & | & y_{k,\lambda} & 0 & s_{k,\lambda}/z_{t_{k,\lambda}} & r_{k,\lambda} &)
\end{array}$$

$\mathbf{G}_{2.k.2.3}$ Formal basis change, on $(\mathbb{D}, \mathbb{D}^*)_{5,7}$, to duplicate $r_{\ell,\lambda}$

$$\begin{array}{lllllll}
\ell < k & \mathbf{k}^*_{\ell,0} = (\ a_{\ell,0}\ \ r_{\ell,0}\ \ 1\ ) & \mathbf{k}^*_{\ell,\lambda} = (\ldots \mid y_{\ell,\lambda} & r_{\ell,\lambda} & s'_{\ell,\lambda}/z_{t_{\ell,\lambda}} & r_{\ell,\lambda} &) \\
\ell = k & \mathbf{k}^*_{k,0} = (\ a_{k,0}\ \ s_{k,0}\ \ 1\ ) & \mathbf{k}^*_{k,\lambda} = (\ldots \mid y_{k,\lambda} & r_{k,\lambda} & s_{k,\lambda}/z_{t_{k,\lambda}} & r_{k,\lambda} &) \\
\ell > k & \mathbf{k}^*_{\ell,0} = (\ a_{\ell,0}\ \ 0\ \ 1\ ) & \mathbf{k}^*_{\ell,\lambda} = (\ldots \mid y_{\ell,\lambda} & r_{\ell,\lambda} & 0 & r_{\ell,\lambda} &)
\end{array}$$

$\mathbf{G}_{2.k.2.4}$ Alteration of the labeling. See Figure 14

$$\begin{array}{lllllll}
\ell < k & \mathbf{k}^*_{\ell,0} = (\ a_{\ell,0}\ \ r_{\ell,0}\ \ 1\ ) & \mathbf{k}^*_{\ell,\lambda} = (\ldots \mid y_{\ell,\lambda} & 0 & s'_{\ell,\lambda}/z_{t_{\ell,\lambda}} & r_{l,\lambda} &) \\
\ell = k & \mathbf{k}^*_{k,0} = (\ a_{k,0}\ \ s_{k,0}\ \ 1\ ) & \mathbf{k}^*_{k,\lambda} = (\ldots \mid y_{k,\lambda} & 0 & s'_{k,\lambda}/z_{t_{k,\lambda}} & r_{k,\lambda} &) \\
\ell > k & \mathbf{k}^*_{\ell,0} = (\ a_{\ell,0}\ \ 0\ \ 1\ ) & \mathbf{k}^*_{\ell,\lambda} = (\ldots \mid y_{\ell,\lambda} & 0 & 0 & r_{\ell,\lambda} &)
\end{array}$$

$\mathbf{G}_{2.k.2.5}$ Limitations on KeyGen-queries: $s_{k,0}$ unpredictable, replaced by a random $r_{k,0}$

$$\begin{array}{lllllll}
\ell = k & \mathbf{k}^*_{k,0} = (\ a_{k,0}\ \ r_{k,0}\ \ 1\ ) & \mathbf{k}^*_{k,\lambda} = (\ldots \mid y_{k,\lambda} & 0 & s'_{k,\lambda}/z_{t_{k,\lambda}} & r_{k,\lambda} &)
\end{array}$$

$\mathbf{G}_{2.k.2.6}$ SubSpace-Ind Property, on $(\mathbb{D}^*, \mathbb{D})_{2,4}$, between $y_{\ell,\lambda}$ and 0

$$\begin{array}{rcllllllll}
\mathbf{k}^*_{k,0} & = (& a_{k,0} & r_{k,0} & 1\ ) & & & & & \\
\mathbf{k}^*_{k,\lambda} & = (& \pi_{k,\lambda}(t_{k,\lambda},-1) & a_{k,\lambda} & | & 0 & 0 & s'_{k,\lambda}/z_{t_{k,\lambda}} & r_{k,\lambda} &)
\end{array}$$

$\mathbf{G}_{2.k.2.7}$ SubSpace-Ind Property, on $(\mathbb{D}, \mathbb{D}^*)_{4,5}$, between 0 and τ

$$\mathbf{c}_t = (\quad \sigma_t(1,t) \quad \omega \quad | \quad \tau \quad 0 \quad \tau z_t \quad u_t \quad)$$

Fig. 13: Sequence of games on the keys for the Del-IND-security proof of our SA-KP-ABE (recall of Figure 8)

Hence, we revisit this gap with an additional sequence presented in the Figure 8 where we focus on the k-th key and the ciphertext, with random $\omega, \tau, \xi, \xi', (\sigma_t), (z_t) \xleftarrow{\$} \mathbb{Z}_q$, but for all

$\mathbf{G}_{2.k.2.3.p.0}$ Hybrid game for $\mathbf{G}_{2.k.2.3}$, with $1 \leq p \leq P+1$ (from Figure 13)

$$
\begin{array}{lll}
 & \mathbf{c}_0 = (\; \omega \quad \tau \quad \xi \;) & \\
 & \mathbf{c}_t = (\; \sigma_t(1,t) \quad \omega & | \quad 0 \quad 0 \quad \tau z_t \quad u_t \;) \\
\ell < k & \mathbf{k}^*_{\ell,0} = (\; a_{\ell,0} \quad r_{\ell,0} \quad 1 \;) & \\
 & \mathbf{k}^*_{\ell,\lambda} = (\; \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) \;\; a_{\ell,\lambda} & | \quad y_{\ell,\lambda} \;\; r_{\ell,\lambda} \quad s'_{\ell,\lambda}/z_{t_{\ell,\lambda}} \quad r_{\ell,\lambda} \;) \\
\ell = k & \mathbf{k}^*_{k,0} = (\; a_{k,0} \quad s_{k,0} \quad 1 \;) & \\
\quad t_{k,\lambda} < p & \mathbf{k}^*_{k,\lambda} = (\; \pi_{k,\lambda}(t_{k,\lambda},-1) \;\; a_{k,\lambda} & | \quad y_{k,\lambda} \;\; r_{k,\lambda} \quad s'_{k,\lambda}/z_{t_{k,\lambda}} \quad r_{k,\lambda} \;) \\
\quad t_{k,\lambda} \geq p & \mathbf{k}^*_{k,\lambda} = (\; \pi_{k,\lambda}(t_{k,\lambda},-1) \;\; a_{k,\lambda} & | \quad y_{k,\lambda} \;\; r_{k,\lambda} \quad s_{k,\lambda}/z_{t_{k,\lambda}} \quad r_{k,\lambda} \;) \\
\ell > k & \mathbf{k}^*_{\ell,0} = (\; a_{\ell,0} \quad 0 \quad 1 \;) & \\
 & \mathbf{k}^*_{\ell,\lambda} = (\; \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) \;\; a_{\ell,\lambda} & | \quad y_{\ell,\lambda} \;\; r_{\ell,\lambda} \quad 0 \quad r_{\ell,\lambda} \;)
\end{array}
$$

$s'_{\ell,\lambda}$ is either the label $s_{\ell,\lambda}$ when $r_{\ell,\lambda} \cdot u_{t_{\ell,\lambda}} = 0$, or a random scalar in $\mathbb{Z}_q$ otherwise

$\mathbf{G}_{2.k.2.3.p.1}$ Swap-Ind Property, on $(\mathbb{D},\mathbb{D}^*)_{5,7}$, for 0 and u_p in $\mathbf{c}_p$ only

$$\mathbf{c}_p = (\ldots| \; 0 \quad u_p \quad \tau z_p \quad 0 \;) \qquad \mathbf{c}_t = (\ldots| \; 0 \quad 0 \quad \tau z_t \quad u_t \;)$$

$\mathbf{G}_{2.k.2.3.p.2}$ Index-Ind Property, on $(\mathbb{D}^*,\mathbb{D})_{1,2,5}$, between $r_{\ell,\lambda}$ and 0, for all $t_{\ell,\lambda} \neq p$

$$
\begin{array}{ll}
\mathbf{c}_p = (\ldots| \; 0 \quad u_p \quad \tau z_p \quad 0 \;) & \mathbf{c}_t = (\ldots| \; 0 \quad 0 \quad \tau z_t \quad u_t \;) \\
t_{\ell,\lambda} \neq p, \ell < k & \mathbf{k}^*_{\ell,\lambda} = (\ldots| \; y_{\ell,\lambda} \quad 0 \quad s'_{\ell,\lambda}/z_{t_{\ell,\lambda}} \quad r_{\ell,\lambda} \;) \\
\mathbf{k}^*_{\ell,\lambda} = (\ldots| \; y_{\ell,\lambda} \quad r_{\ell,\lambda} \quad s'_{\ell,\lambda}/z_p \quad r_{\ell,\lambda} \;) & t_{\ell,\lambda} = p, \ell < k \\
t_{k,\lambda} < p, \ell = k & \mathbf{k}^*_{k,\lambda} = (\ldots| \; y_{k,\lambda} \quad 0 \quad s'_{k,\lambda}/z_{t_{k,\lambda}} \quad r_{k,\lambda} \;) \\
\mathbf{k}^*_{k,\lambda} = (\ldots| \; y_{k,\lambda} \quad r_{k,\lambda} \quad s_{k,\lambda}/z_p \quad r_{k,\lambda} \;) & t_{k,\lambda} = p, \ell = k \\
t_{k,\lambda} > p, \ell = k & \mathbf{k}^*_{k,\lambda} = (\ldots| \; y_{k,\lambda} \quad 0 \quad s_{k,\lambda}/z_{t_{k,\lambda}} \quad r_{k,\lambda} \;) \\
t_{\ell,\lambda} \neq p, \ell > k & \mathbf{k}^*_{\ell,\lambda} = (\ldots| \; y_{\ell,\lambda} \quad 0 \quad 0 \quad r_{\ell,\lambda} \;) \\
\mathbf{k}^*_{\ell,\lambda} = (\ldots| \; y_{\ell,\lambda} \quad r_{\ell,\lambda} \quad 0 \quad r_{\ell,\lambda} \;) & t_{\ell,\lambda} = p, \ell > k
\end{array}
$$

$\mathbf{G}_{2.k.2.3.p.3}$ Formal change of basis on column 5, multiplying ciphertext by $\tau z_p/u_p$

$$
\begin{array}{ll}
\mathbf{c}_p = (\ldots| \; 0 \quad \tau z_p \quad \tau z_p \quad 0 \;) & \mathbf{c}_t = (\ldots| \; 0 \quad 0 \quad \tau z_t \quad u_t \;) \\
\mathbf{k}^*_{\ell,\lambda} = (\ldots| \; y_{\ell,\lambda} \quad r_{\ell,\lambda}u_p/\tau z_p \quad s'_{\ell,\lambda}/z_p \quad r_{\ell,\lambda} \;) & t_{\ell,\lambda} = p, \ell < k \\
\mathbf{k}^*_{k,\lambda} = (\ldots| \; y_{k,\lambda} \quad r_{k,\lambda}u_p/\tau z_p \quad s_{k,\lambda}/z_p \quad r_{k,\lambda} \;) & t_{k,\lambda} = p, \ell = k \\
\mathbf{k}^*_{\ell,\lambda} = (\ldots| \; y_{\ell,\lambda} \quad r_{\ell,\lambda}u_p/\tau z_p \quad 0 \quad r_{\ell,\lambda} \;) & t_{\ell,\lambda} = p, \ell > k
\end{array}
$$

$\mathbf{G}_{2.k.2.3.p.4}$ Index-Ind Property, on $(\mathbb{D},\mathbb{D}^*)_{1,2,5}$, between 0 and τz_t, for $t \neq p$

$$\mathbf{c}_p = (\ldots| \; 0 \quad \tau z_p \quad \tau z_p \quad 0 \;) \qquad \mathbf{c}_t = (\ldots| \; 0 \quad \tau z_t \quad \tau z_t \quad u_t \;)$$

$\mathbf{G}_{2.k.2.3.p.5}$ Swap-Ind Property, on $(\mathbb{D}^*,\mathbb{D})_{4,5,6}$, between $r_{k,\lambda}u_p/\tau z_p$ and 0, for $t_{k,\lambda} = p$ only

$$
\begin{array}{ll}
\mathbf{c}_p = (\ldots| \; 0 \quad \tau z_p \quad \tau z_p \quad 0 \;) & \mathbf{c}_t = (\ldots| \; 0 \quad \tau z_t \quad \tau z_t \quad u_t \;) \\
\mathbf{k}^*_{\ell,\lambda} = (\ldots| \; y_{\ell,\lambda} \quad r_{\ell,\lambda}u_p/\tau z_p \quad s'_{\ell,\lambda}/z_p \quad r_{\ell,\lambda} \;) & t_{\ell,\lambda} = p, \ell < k \\
\mathbf{k}^*_{k,\lambda} = (\ldots| \; y_{k,\lambda} \quad 0 \quad \frac{s_{k,\lambda}+r_{k,\lambda}u_p/\tau}{z_p} \quad r_{k,\lambda} \;) & t_{k,\lambda} = p, \ell = k \\
\mathbf{k}^*_{\ell,\lambda} = (\ldots| \; y_{\ell,\lambda} \quad r_{\ell,\lambda}u_p/\tau z_p \quad 0 \quad r_{\ell,\lambda} \;) & t_{\ell,\lambda} = p, \ell > k
\end{array}
$$

Fig. 14: Sequence of sub-games on the attributes for the Del-IND-security proof of our SA-KP-ABE

$\mathbf{G}_{2.k.2.3.p.6}$ SubSpace-Ind Property, on $(\mathbb{D}^*, \mathbb{D})_{4,7}$, to randomize $r_{k,\lambda}$ for $t_{k,\lambda} = p$

$$
\begin{array}{lllll|lllll}
\mathbf{c}_p = (\ldots| & 0 & \tau z_p & \tau z_p & 0\ ) & \mathbf{c}_t = (\ldots| & 0 & \tau z_t & \tau z_t & u_t\ ) \\
 & t_{\ell,\lambda} \neq p, \ell < k & & & & \mathbf{k}^*_{\ell,\lambda} = (\ldots| & y_{\ell,\lambda} & 0 & s'_{\ell,\lambda}/z_{t_{\ell,\lambda}} & r_{\ell,\lambda}\ ) \\
 & t_{k,\lambda} < p, \ell = k & & & & \mathbf{k}^*_{k,\lambda} = (\ldots| & y_{k,\lambda} & 0 & s'_{k,\lambda}/z_{t_{k,\lambda}} & r_{k,\lambda}\ ) \\
\mathbf{k}^*_{\ell,\lambda} = (\ldots| y_{\ell,\lambda} & r_{\ell,\lambda} u_p/\tau z_p & s'_{\ell,\lambda}/z_p & r_{\ell,\lambda}\ ) & & t_{\ell,\lambda} = p, \ell < k \\
\mathbf{k}^*_{k,\lambda} = (\ldots| y_{k,\lambda} & 0 & s'_{k,\lambda}/z_p & r'_{k,\lambda}\ ) & & t_{k,\lambda} = p, \ell = k \\
\mathbf{k}^*_{\ell,\lambda} = (\ldots| y_{\ell,\lambda} & r_{\ell,\lambda} u_p/\tau z_p & 0 & r_{\ell,\lambda}\ ) & & t_{\ell,\lambda} = p, \ell > k \\
 & t_{k,\lambda} > p, \ell = k & & & & \mathbf{k}^*_{k,\lambda} = (\ldots| & y_{k,\lambda} & 0 & s_{k,\lambda}/z_{t_{k,\lambda}} & r_{k,\lambda}\ ) \\
 & t_{\ell,\lambda} \neq p, \ell > k & & & & \mathbf{k}^*_{\ell,\lambda} = (\ldots| & y_{\ell,\lambda} & 0 & 0 & r_{\ell,\lambda}\ )
\end{array}
$$

$\mathbf{G}_{2.k.2.3.p.7}$ Swap-Ind Property, on $(\mathbb{D}^*, \mathbb{D})_{4,5,6}$, between 0 and $r'_{k,\lambda} u_p/\tau z_p$, for $t_{k,\lambda} = p$ only

$$
\begin{array}{lllll|lllll}
\mathbf{c}_p = (\ldots| & 0 & \tau z_p & \tau z_p & 0\ ) & \mathbf{c}_t = (\ldots| & 0 & \tau z_t & \tau z_t & u_t\ ) \\
\mathbf{k}^*_{\ell,\lambda} = (\ldots| & y_{\ell,\lambda} & r_{\ell,\lambda} u_p/\tau z_p & s'_{\ell,\lambda}/z_p & r_{\ell,\lambda}\ ) & t_{\ell,\lambda} = p, \ell < k \\
\mathbf{k}^*_{k,\lambda} = (\ldots| & y_{k,\lambda} & r'_{k,\lambda} u_p/\tau z_p & \frac{s'_{k,\lambda} - r'_{k,\lambda} u_p/\tau}{z_p} & r'_{k,\lambda}\ ) & t_{k,\lambda} = p, \ell = k \\
\mathbf{k}^*_{\ell,\lambda} = (\ldots| & y_{\ell,\lambda} & r_{\ell,\lambda} u_p/\tau z_p & 0 & r_{\ell,\lambda}\ ) & t_{\ell,\lambda} = p, \ell > k
\end{array}
$$

$\mathbf{G}_{2.k.2.3.p.8}$ Index-Ind Property, on $(\mathbb{D}, \mathbb{D}^*)_{1,2,5}$, between τz_t and 0

$$
\begin{array}{lllll|lllll}
\mathbf{c}_p = (\ldots| & 0 & \tau z_p & \tau z_p & 0\ ) & \mathbf{c}_t = (\ldots| & 0 & 0 & \tau z_t & u_t\ ) \\
 & t_{\ell,\lambda} \neq p, \ell < k & & & & \mathbf{k}^*_{\ell,\lambda} = (\ldots| & y_{\ell,\lambda} & 0 & s'_{\ell,\lambda}/z_{t_{\ell,\lambda}} & r_{\ell,\lambda}\ ) \\
 & t_{k,\lambda} < p, \ell = k & & & & \mathbf{k}^*_{k,\lambda} = (\ldots| & y_{k,\lambda} & 0 & s'_{k,\lambda}/z_{t_{k,\lambda}} & r_{k,\lambda}\ ) \\
\mathbf{k}^*_{\ell,\lambda} = (\ldots| y_{\ell,\lambda} & r_{\ell,\lambda} u_p/\tau z_p & s'_{\ell,\lambda}/z_p & r_{\ell,\lambda}\ ) & & t_{\ell,\lambda} = p, \ell \leq k \\
\mathbf{k}^*_{\ell,\lambda} = (\ldots| y_{\ell,\lambda} & r_{\ell,\lambda} u_p/\tau z_p & 0 & r_{\ell,\lambda}\ ) & & t_{\ell,\lambda} = p, \ell > k \\
 & t_{k,\lambda} > p, \ell = k & & & & \mathbf{k}^*_{k,\lambda} = (\ldots| & y_{k,\lambda} & 0 & s_{k,\lambda}/z_{t_{k,\lambda}} & r_{k,\lambda}\ ) \\
 & t_{\ell,\lambda} \neq p, \ell > k & & & & \mathbf{k}^*_{\ell,\lambda} = (\ldots| & y_{\ell,\lambda} & 0 & 0 & r_{\ell,\lambda}\ )
\end{array}
$$

$\mathbf{G}_{2.k.2.3.p.9}$ Formal change of basis

$$
\begin{array}{lllll|lllll}
\mathbf{c}_p = (\ldots| & 0 & u_p & \tau z_p & 0\ ) & \mathbf{c}_t = (\ldots| & 0 & 0 & \tau z_t & u_t\ ) \\
\mathbf{k}^*_{\ell,\lambda} = (\ldots| & y_{\ell,\lambda} & r_{\ell,\lambda} & s'_{\ell,\lambda}/z_p & r_{\ell,\lambda}\ ) & t_{\ell,\lambda} = p, \ell \leq k \\
\mathbf{k}^*_{\ell,\lambda} = (\ldots| & y_{\ell,\lambda} & r_{\ell,\lambda} & 0 & r_{\ell,\lambda}\ ) & t_{\ell,\lambda} = p, \ell > k
\end{array}
$$

$\mathbf{G}_{2.k.2.3.p.10}$ Index-Ind Property, on $(\mathbb{D}^*, \mathbb{D})_{1,2,5}$, between 0 and $r_{\ell,\lambda}$, for all $t_{\ell,\lambda} \neq p$

$$
\begin{array}{lllll|lllll}
\mathbf{c}_p = (\ldots| & 0 & u_p & \tau z_p & 0\ ) & \mathbf{c}_t = (\ldots| & 0 & 0 & \tau z_t & u_t\ ) \\
 & t_{\ell,\lambda} \neq p, \ell < k & & & & \mathbf{k}^*_{\ell,\lambda} = (\ldots| & y_{\ell,\lambda} & r_{\ell,\lambda} & s'_{\ell,\lambda}/z_{t_{\ell,\lambda}} & r_{\ell,\lambda}\ ) \\
 & t_{k,\lambda} < p, \ell = k & & & & \mathbf{k}^*_{k,\lambda} = (\ldots| & y_{k,\lambda} & r_{k,\lambda} & s'_{k,\lambda}/z_{t_{k,\lambda}} & r_{k,\lambda}\ ) \\
\mathbf{k}^*_{\ell,\lambda} = (\ldots| y_{\ell,\lambda} & r_{\ell,\lambda} & s'_{\ell,\lambda}/z_p & r_{\ell,\lambda}\ ) & & t_{\ell,\lambda} = p, \ell \leq k \\
\mathbf{k}^*_{\ell,\lambda} = (\ldots| 0 & r_{\ell,\lambda} & 0 & r_{\ell,\lambda}\ ) & & t_{\ell,\lambda} = p, \ell > k \\
 & t_{k,\lambda} > p, \ell = k & & & & \mathbf{k}^*_{k,\lambda} = (\ldots| & y_{k,\lambda} & r_{k,\lambda} & s_{k,\lambda}/z_{t_{k,\lambda}} & r_{k,\lambda}\ ) \\
 & t_{\ell,\lambda} \neq p, \ell > k & & & & \mathbf{k}^*_{\ell,\lambda} = (\ldots| & y_{\ell,\lambda} & r_{\ell,\lambda} & 0 & r_{\ell,\lambda}\ )
\end{array}
$$

$\mathbf{G}_{2.k.2.3.p.11}$ Swap-Ind Property, on $(\mathbb{D}, \mathbb{D}^*)_{5,7}$, for 0 and u_p

$$
\begin{array}{lllll|lllll}
\mathbf{c}_p = (\ldots| & 0 & 0 & \tau z_p & u_p\ ) & \mathbf{c}_t = (\ldots| & 0 & 0 & \tau z_t & u_t\ )
\end{array}
$$

Fig. 14: Sequence of sub-games on the attributes for the Del-IND-security proof of our SA-KP-ABE (Cont'ed)

the OKeyGen-query, random $a_{\ell,0}, (\pi_{\ell,\lambda}) \stackrel{\$}{\leftarrow} \mathbb{Z}_q$, as well as a random $a_{\ell,0}$-labeling $(a_{\ell,\lambda})_\lambda$ of the access-tree $\mathcal{T}_k$, but also $s_{\ell,0} \stackrel{\$}{\leftarrow} \mathbb{Z}_q$ and a second independent random $s_{\ell,0}$-labeling $(s_{\ell,\lambda})_\lambda$ of the access-tree $\mathcal{T}_k$, and an independent random $r_{\ell,0} \stackrel{\$}{\leftarrow} \mathbb{Z}_q$. The goal is to replace each label $s_{k,\lambda}$ by a random independent value $s'_{k,\lambda}$ when $u_{t_{k,\lambda}} \cdot r_{k,\lambda} \neq 0$. As a consequence, we will consider below that $s'_{k,\lambda}$ denotes either the label $s_{k,\lambda}$ when $u_{t_{k,\lambda}} \cdot r_{k,\lambda} = 0$ or a random scalar:

Game $\mathbf{G}_{2.k.2.0}$ The first game is exactly $\mathbf{G}_{2.k.2}$, where the simulator honestly runs the setup, with $\mathsf{PK} = \{(\mathbf{b}_1, \mathbf{b}_3, \mathbf{b}_1^*), (\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_1^*, \mathbf{d}_2^*, \mathbf{d}_3^*)\}$, $\mathsf{SK} = \{\mathbf{d}_7\}$, and $\mathsf{MK} = \{\mathbf{b}_3^*, \mathbf{d}_7^*\}$, from random dual orthogonal bases.

$\mathsf{OKeyGen}(\mathcal{T}_\ell)$ (or ODelegate-queries): The simulator builds the ℓ-th key:

$$\begin{array}{ll}
\ell < k & \mathbf{k}^*_{\ell,0} = (a_{\ell,0}, r_{\ell,0}, 1)_{\mathbb{B}^*} \\
 & \mathbf{k}^*_{\ell,\lambda} = (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, 0, s'_{\ell,\lambda}/z_{t_{\ell,\lambda}}, r_{\ell,\lambda})_{\mathbb{D}^*} \\
\ell = k & \mathbf{k}^*_{k,0} = (a_{k,0}, s_{k,0}, 1)_{\mathbb{B}^*} \\
 & \mathbf{k}^*_{k,\lambda} = (\pi_{k,\lambda}(t_{k,\lambda}, -1), a_{k,\lambda}, 0, 0, s_{k,\lambda}/z_{t_{k,\lambda}}, r_{k,\lambda})_{\mathbb{D}^*} \\
\ell > k & \mathbf{k}^*_{\ell,0} = (a_{\ell,0}, 0, 1)_{\mathbb{B}^*} \\
 & \mathbf{k}^*_{\ell,\lambda} = (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, 0, 0, r_{\ell,\lambda})_{\mathbb{D}^*}
\end{array}$$

with $r_{\ell,\lambda} \stackrel{\$}{\leftarrow} \mathbb{Z}_q$ if $\lambda \in \mathcal{L}_a$ or $r_{\ell,\lambda} = 0$ if $\lambda \in \mathcal{L}_p$. The decryption key dk_ℓ is then $(\mathbf{k}^*_{\ell,0}, (\mathbf{k}^*_{\ell,\lambda})_\lambda)$.

$\mathsf{OEncaps}(\Gamma_{m,v}, \Gamma_{m,i})$**:** The simulator builds the m-th ciphertext using all the known vectors of the basis:

$$\mathbf{c}_{m,0} = (\omega_m, 0, \xi_m)_{\mathbb{B}} \qquad \mathbf{c}_{m,t} = (\sigma_{m,t}(t, -1), \omega_m, 0, 0, 0, u_{m,t})_{\mathbb{D}}$$

with $\omega_m, \xi_m \stackrel{\$}{\leftarrow} \mathbb{Z}_q$, $\sigma_{m,t} \stackrel{\$}{\leftarrow} \mathbb{Z}_q$ and $u_{m,t} \stackrel{\$}{\leftarrow} \mathbb{Z}_q^*$ if $t \in \Gamma_{m,i}$ or $u_{m,t} \leftarrow 0$ if $t \in \Gamma_{m,v}$. The ciphertext C_m is then $(\mathbf{c}_{m,0}, (\mathbf{c}_{m,t})_t)$;

$\mathsf{RoREncaps}(\Gamma_v, \Gamma_i)$**:** On the unique query on a set of attributes $(\Gamma_v \cup \Gamma_i)$, the simulator generates the ciphertext $C = (\mathbf{c}_0, (\mathbf{c}_t)_{t \in (\Gamma_v \cup \Gamma_i)})$ where

$$\mathbf{c}_0 = (\omega, \tau, \xi)_{\mathbb{B}} \qquad \mathbf{c}_t = (\sigma_t(1, t), \omega, \tau, 0, \tau z_t, u_t)_{\mathbb{D}}$$

for all the attributes $t \in (\Gamma_v \cup \Gamma_i)$, with $u_t \stackrel{\$}{\leftarrow} \mathbb{Z}_q$ if $t \in \Gamma_i$ or $u_t = 0$ if $t \in \Gamma_v$. According to the real or random game (bit $b \stackrel{\$}{\leftarrow} \{0,1\}$), one outputs (K_b, C).

From the adversary's guess b' for b, if for some $\tilde{\mathcal{T}}$, $\tilde{\mathcal{T}}(\Gamma_v, \Gamma_i) = 1$, then $\beta \stackrel{\$}{\leftarrow} \{0,1\}$, otherwise $\beta = b'$. We denote $\mathsf{Adv}_{2.k.2.0} = \Pr[\beta = 1|b = 1] - \Pr[\beta = 1|b = 0]$. The goal of this sequence of games is to replace $s_{k,0}$, that can be derived by an acceptable set of $s_{k,\lambda}$, by a random and independent value $r_{k,0}$, in the key generated during the k-th OKeyGen-query.

Indeed, to be a legitimate attack (that does not randomize the adversary's guess b'), for all the key queries $\tilde{\mathcal{T}}_\ell$, one must have $\tilde{\mathcal{T}}_\ell(\Gamma_v, \Gamma_i) = 0$. In particular, $\tilde{\mathcal{T}}_k(\Gamma_v, \Gamma_i) = 0$: this means that

- either the regular access-tree policy is not met, *i.e.*, $\mathcal{T}_k(\Gamma_v \cup \Gamma_i) = 0$.
- or the regular access-tree policy is met, but one active key leaf matches one invalid ciphertext attribute: $\forall \mathcal{T}' \in \mathsf{EPT}(\mathcal{T}_k, \Gamma_v \cup \Gamma_i), \exists \lambda \in \mathcal{T}' \cap \mathcal{L}_a, A(\lambda) \in \Gamma_i$, and from the assumptions, for any such tree $\mathcal{T}'$, the active leave is an independent leave.

In both cases, we will use the same technique to show $s_{k,0}$ is independent from any other value. But first, we will replace all the active leaves associated to invalid ciphertexts in the challenge ciphertext by inactive leaves.

Of course, in the following sequence, we will have to take care of the simulation of the challenge ciphertext, but also of the OEncaps-oracle. For the latter, we will have to make precise this simulation when public vectors $(\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3)$ or the private vector $\mathbf{d}_7$ are impacted.

Game $\mathbf{G}_{2.k.2.1}$ In this game, we first clean the 4-th column of the ciphertext from the τ. To this aim, we are given a tuple $(a \cdot G_1, b \cdot G_1, c \cdot G_1)$ in $\mathbb{G}_1$, where $c = ab + \mu \bmod q$ with either $\mu = 0$ or $\mu = \tau$ (fixed from $\mathbf{c}_0$). When we start from random dual orthogonal bases $(\mathbb{U}, \mathbb{U}^*)$ and $(\mathbb{V}, \mathbb{V}^*)$ of size 3 and 7 respectively, one considers the matrices:

$$D = \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}_{3,4} \qquad D' = \begin{pmatrix} 1 & 0 \\ -a & 1 \end{pmatrix}_{3,4} \qquad \mathbb{D}^* = D' \cdot \mathbb{V}^* \qquad \mathbb{D} = D \cdot \mathbb{V}$$

We can calculate all vectors but $\mathbf{d}_3^*$. Hence, there is no problem for simulating the $\mathsf{OEncaps}$-queries. For the challenge ciphertext, we exploit the DSDH assumption:

$$\begin{aligned}\mathbf{c}_t &= (\sigma_t(1,t), b, c, 0, \tau z_t, u_t)_{\mathbb{V}} = (\sigma_t(1,t), b, c - ab, 0, \tau z_t, u_t)_{\mathbb{D}} \\ &= (\sigma_t(1,t), b, \mu, 0, \tau z_t, u_t)_{\mathbb{D}}\end{aligned}$$

which is correct, with $\omega = b$ and according to μ, this is either τ, as in the previous game or 0 as in this game. For the keys, one notes that the 4-th component is 0, and so the change of basis has no impact on the 3-rd component, when using basis $\mathbb{V}^*$:

$$\mathbf{k}^*_{\ell,\lambda} = (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, \ldots)_{\mathbb{V}^*} = (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, \ldots)_{\mathbb{D}^*}$$

Then, we have $\mathsf{Adv}_{2.k.2.0} - \mathsf{Adv}_{2.k.2.1} \leq 2 \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t)$.

Game $\mathbf{G}_{2.k.2.2}$ In this game, we can now introduce noise in the 4-th column the keys. In order to properly deal with delegated keys, as for $r_{\ell,\lambda}$ that have to be the same values for all the leaves delegated from the same initial key, we will also set the same random $y_{\ell,\lambda}$. To this aim, we are given a tuple $(a \cdot G_2, b \cdot G_2, c \cdot G_2)$ in $\mathbb{G}_2$, where $c = ab + \zeta \bmod q$ with either $\zeta = 0$ or $\zeta \stackrel{\$}{\leftarrow} \mathbb{Z}_q^*$. We choose additional random scalars $\alpha_{\ell,\lambda}, \beta_{\ell,\lambda} \stackrel{\$}{\leftarrow} \mathbb{Z}_q$ (but the same $\alpha_{\ell,\lambda}$ for all the leaves delegated from the same initial key), to virtually set $b_{\ell,\lambda} = \alpha_{\ell,\lambda} \cdot b + \beta_{\ell,\lambda}$ and $c_{\ell,\lambda} = \alpha_{\ell,\lambda} \cdot c + \beta_{\ell,\lambda} \cdot a$, then $c_{\ell,\lambda} - ab_{\ell,\lambda} = \zeta \cdot \alpha_{\ell,\lambda}$, which are either 0 or independent random values. When we start from random dual orthogonal bases $(\mathbb{U}, \mathbb{U}^*)$ and $(\mathbb{V}, \mathbb{V}^*)$ of size 3 and 7 respectively, one considers the matrices:

$$D = \begin{pmatrix} 1 & 0 \\ a & 1 \end{pmatrix}_{2,4} \qquad D' = \begin{pmatrix} 1 & -a \\ 0 & 1 \end{pmatrix}_{2,4} \qquad \mathbb{D}^* = D' \cdot \mathbb{V}^* \qquad \mathbb{D} = D \cdot \mathbb{V}$$

We can calculate all vectors but $\mathbf{d}_4$, which is not used anywhere. Then, for the keys, we exploit the DDH assumption:

$$\begin{aligned}\mathbf{k}^*_{\ell,\lambda} &= (b_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, c_{\ell,\lambda}, \ldots)_{\mathbb{V}^*} = (b_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, c_{\ell,\lambda} - ab_{\ell,\lambda}, \ldots)_{\mathbb{D}^*} \\ &= (b_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, \zeta \cdot \alpha_{\ell,\lambda}, \ldots)_{\mathbb{D}^*}\end{aligned}$$

Which is either the previous game, with $\pi_{\ell,\lambda} = b_{\ell,\lambda}$, when $\zeta = 0$, or the current game with $y_{\ell,\lambda} = \zeta \cdot \alpha_{\ell,\lambda}$ (the same random $y_{\ell,\lambda}$ for all the leaves delegated from the same initial key): $\mathsf{Adv}_{2.k.2.1} - \mathsf{Adv}_{2.k.2.2} \leq \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t)$.

Game $\mathbf{G}_{2.k.2.3}$ In this game, we duplicate every $r_{\ell,\lambda}$ into the 5-th column of the key. To this aim, one defines the matrices

$$D = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}_{5,7} \qquad D' = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}_{5,7} \qquad \mathbb{D}^* = D' \cdot \mathbb{V}^* \qquad \mathbb{D} = D \cdot \mathbb{V}$$

which only modifies $\mathbf{d}_5$, which is hidden, and $\mathbf{d}_7^*$, which is secret, so the change is indistinguishable for the adversary. One can compute the keys and ciphertexts as follows, for all leaves λ of each query ℓ of the adversary:

$$\begin{aligned}\mathbf{k}^*_{\ell,\lambda} &= (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, y_{\ell,\lambda}, 0, s_{\ell,\lambda}/z_{t_{\ell,\lambda}}, r_{\ell,\lambda})_{\mathbb{V}^*} \\ &= (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, y_{\ell,\lambda}, r_{\ell,\lambda}, s_{\ell,\lambda}/z_{t_{\ell,\lambda}}, r_{\ell,\lambda})_{\mathbb{D}^*} \\ \mathbf{c}_t &= (\sigma_t(1,t), \omega, 0, 0, \tau z_t, u_t)_{\mathbb{V}} = (\sigma_t(1,t), \omega, 0, 0, \tau z_t, u_t)_{\mathbb{D}}\end{aligned}$$

As the 5-th component in the ciphertext is 0, the change of basis makes no change. And this is the same for the ciphertexts generated by the $\mathsf{OEncaps}$-simulation. Hence, the perfect indistinguishability between the two games: $\mathsf{Adv}_{2.k.2.3} = \mathsf{Adv}_{2.k.2.2}$.

Game $\mathbf{G}_{2.k.2.4}$ In this game, we target the k-th $\mathsf{OKeyGen}$-query, and replace $s_{k,\lambda}$ by an independent $s'_{k,\lambda}$ for all the active leaves that correspond to an invalid attribute in the challenge ciphertext. For the sake of simplicity, $s'_{\ell,\lambda}$ is either the label $s_{\ell,\lambda}$ when $r_{\ell,\lambda} \cdot u_{t_{\ell,\lambda}} = 0$, or a random independent scalar in $\mathbb{Z}_q$:

$$\begin{aligned}
\mathbf{k}^*_{k,0} &= (a_{k,0}, s_{k,0}, 1)_{\mathbb{B}^*} \\
\mathbf{k}^*_{k,\lambda} &= (\pi_{k,\lambda}(t_{k,\lambda}, -1), a_{k,\lambda}, y_{k,\lambda}, 0, s'_{k,\lambda}/z_{t_{k,\lambda}}, r_{k,\lambda})_{\mathbb{D}^*}
\end{aligned}$$

But to this aim, we will need an additional sequence of sub-games $\mathbf{G}_{2.k.2.3.p.*}$, that will operate iteratively on each attribute p, to convert $\mathbf{G}_{2.k.2.3}$ into $\mathbf{G}_{2.k.2.4}$, as presented in the Figure 14. But we first complete the first sequence, and details the sub-sequence afterwards.

Game $\mathbf{G}_{2.k.2.5}$ For the k-th key query, one can now replace $s_{k,0}$ by $r_{k,0}$. Indeed, as explained in the Remark 27, for missing ciphertexts in the challenge ciphertext, the associated leaves in the key have unpredictable $s_{k,\lambda}$. In addition, for active leaves that correspond to invalid attributes in the challenge ciphertext, $s_{k,\lambda}$ have been transformed into $s'_{k,\lambda}$, random independent values. Then, we can consider that all the leaves associated to attributes not in Γ are false, but also active leaves associated to attributes in Γ_i are false. As $\tilde{\mathcal{T}}_k(\Gamma_v, \Gamma_i) = 0$, the root label is unpredictable. One thus generates the k-th key query as:

$$\begin{aligned}
\mathbf{k}^*_{k,0} &= (a_{k,0}, r_{k,0}, 1)_{\mathbb{B}^*} \\
\mathbf{k}^*_{k,\lambda} &= (\pi_{k,\lambda}(t_{k,\lambda}, -1), a_{k,\lambda}, y_{k,\lambda}, 0, s'_{k,\lambda}/z_{t_{k,\lambda}}, r_{k,\lambda})_{\mathbb{D}^*}
\end{aligned}$$

Game $\mathbf{G}_{2.k.2.6}$ We can now invert the above step, when we added $y_{\ell,\lambda}$: $\mathsf{Adv}_{2.k.2.5} - \mathsf{Adv}_{2.k.2.6} \leq \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t)$.

Game $\mathbf{G}_{2.k.2.7}$ We can now invert the above step, when we removed τ from the ciphertext: $\mathsf{Adv}_{2.k.2.6} - \mathsf{Adv}_{2.k.2.7} \leq 2 \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t)$.

We now detail the sub-sequence starting from $\mathbf{G}_{2.k.2.3.p.0}$ to prove the indistinguishability between $\mathbf{G}_{2.k.2.3}$ and $\mathbf{G}_{2.k.2.4}$. In the new hybrid game $\mathbf{G}_{2.k.2.3.p.0}$, the critical point will be the p-th ciphertext, where, when $p = 1$, this is exactly the above Game $\mathbf{G}_{2.k.2.3}$, and when $p = P + 1$, this is the above Game $\mathbf{G}_{2.k.2.4}$. And it will be clear, for any p, that $\mathbf{G}_{2.k.2.3.p.11} = \mathbf{G}_{2.k.2.3.p+1.0}$.

With random $\omega, \tau, \xi, \xi', (\sigma_t), (z_t) \stackrel{\$}{\leftarrow} \mathbb{Z}_q$, but for all the $\mathsf{OKeyGen}$-query, random $a_{\ell,0}, (y_\lambda)$, $(\pi_{\ell,\lambda}) \stackrel{\$}{\leftarrow} \mathbb{Z}_q$, as well as a random $a_{\ell,0}$-labeling $(a_{\ell,\lambda})_\lambda$ of the access-tree $\mathcal{T}_k$, but also $s_{\ell,0} \stackrel{\$}{\leftarrow} \mathbb{Z}_q$ and a second independent random $s_{\ell,0}$-labeling $(s_{\ell,\lambda})_\lambda$ of the access-tree $\mathcal{T}_k$, and an independent random $r_{\ell,0} \stackrel{\$}{\leftarrow} \mathbb{Z}_q$:

Game $\mathbf{G}_{2.k.2.3.p.0}$: One defines the hybrid game for p:

$$\begin{array}{ll}
 & \mathbf{k}^*_{k,0} = (a_{k,0}, s_{k,0}, 1)_{\mathbb{B}^*} \\
t_{k,\lambda} < p & \mathbf{k}^*_{k,\lambda} = (\pi_{k,\lambda}(t_{k,\lambda}, -1), a_{k,\lambda}, y_{k,\lambda}, r_{k,\lambda}, s'_{k,\lambda}/z_{t_{k,\lambda}}, r_{k,\lambda})_{\mathbb{D}^*} \\
t_{k,\lambda} \geq p & \mathbf{k}^*_{k,\lambda} = (\pi_{k,\lambda}(t_{k,\lambda}, -1), a_{k,\lambda}, y_{k,\lambda}, r_{k,\lambda}, s_{k,\lambda}/z_{t_{k,\lambda}}, r_{k,\lambda})_{\mathbb{D}^*}
\end{array}$$

where $s'_{\ell,\lambda}$ is either the label $s_{\ell,\lambda}$ when $r_{\ell,\lambda} \cdot u_{t_{\ell,\lambda}} = 0$, or a random independent scalar in $\mathbb{Z}_q$ (when this is an active leaf that corresponds to an invalid ciphertext).

So one can note that if at the challenge query $p \in \Gamma_v$, then $u_p = 0$, and so we can jump to $\mathbf{G}_{2.k.2.3.p.11}$, but we do not know it before the challenge-query is asked, whereas we have to simulate the keys. This is the reason why we need to know the super sets A_v and A_i: the challenge ciphertext is anticipated with $u_p = 0$ if $p \in A_v$ or with $u_p \stackrel{\$}{\leftarrow} \mathbb{Z}_q^*$ if $p \in A_i$.

Game $\mathbf{G}_{2.k.2.3.p.1}$: The previous game and this game are indistinguishable under the DDH assumption in $\mathbb{G}_1$: one essentially uses theorem 21. Given a tuple $(a\cdot G_1, b\cdot G_1, c\cdot G_1)$ in $\mathbb{G}_1$, where $c = ab + \mu \bmod q$ with either $\mu = 0$ or $\mu = u_p$, the 7-th component of the p-th ciphertext. When we start from random dual orthogonal bases $(\mathbb{U}, \mathbb{U}^*)$ and $(\mathbb{V}, \mathbb{V}^*)$ of size 3 and 7 respectively, one considers the matrices:

$$D = \begin{pmatrix} 1 & a & -a \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}_{1,5,7} \qquad D' = \begin{pmatrix} 1 & 0 & 0 \\ -a & 1 & 0 \\ a & 0 & 1 \end{pmatrix}_{1,5,7} \qquad \mathbb{D} = D\cdot\mathbb{V} \qquad \mathbb{D}^* = D'\cdot\mathbb{V}^*$$

We can calculate all vectors but $\mathbf{d}_5^*$ and $\mathbf{d}_7^*$, which are not in the public key. Through $\mathbb{V}$, we calculate the challenge ciphertext for the attribute of the p-th ciphertext

$$\begin{aligned}\mathbf{c}_p &= (0,0,\omega,0,0,\tau z_p,u_p)_{\mathbb{D}} + (b(1,p),0,0,c,0,-c)_{\mathbb{V}}\\ &= (0,0,\omega,0,0,\tau z_p,u_p)_{\mathbb{D}} + (b(1,p),0,0,c-ab,0,ab-c)_{\mathbb{D}}\\ &= (b(1,p),\omega,0,\mu,\tau z_p,u_p-\mu)_{\mathbb{D}}\end{aligned}$$

If $\mu = 0$, we are in the previous game. If $\mu = u_p$, then we are in the current game. Then, every other ciphertext is computed directly in $\mathbb{D}$:

$$\forall t \neq p, \mathbf{c}_t = (\sigma_t(1,t),\omega,0,0,\tau z_t,u_t)_{\mathbb{D}}$$

as well as the answers to OEncaps-queries. The keys are calculated through $\mathbb{V}^*$ but are unchanged by the change of basis because the 5-th and 7-th components are exactly the same for every key query ℓ, and thus cancel themselves in the 1st component. We thus have $\mathsf{Adv}_{2.k.2.3.p.0} - \mathsf{Adv}_{2.k.2.3.p.1} \leq 2\cdot\mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t)$.

Game $\mathbf{G}_{2.k.2.3.p.2}$: We keep the $r_{\ell,\lambda}$ value (at the 5-th hidden position) in the keys such that $t_{\ell,\lambda} = p$, and replace it in all other keys by 0, in order to prepare the possibility to later modify the ciphertexts on this component. To show this is possible without impacting the other vectors, we use the Index-Ind property from Theorem 23, but in another level of sequence of hybrid games, for $\gamma \in \{1,\dots,P\}\backslash\{p\}$:

Game $\mathbf{G}_{2.k.2.3.p.1.\gamma}$: We consider the following hybrid game, where the first satisfied condition on the indices is applied:

$$\mathbf{c}_p = (\sigma_p(1,p),\omega,0,u_p,\tau z_p,0)_{\mathbb{D}} \qquad \mathbf{c}_t = (\sigma_t(1,t),\omega,0,0,\tau z_t,u_t)_{\mathbb{D}} \qquad t\neq p$$

$$\begin{aligned}&\mathbf{k}^*_{\ell,\lambda} = (\pi_{\ell,\lambda}(t_{\ell,\lambda},-1),a_{\ell,\lambda},y_{\ell,\lambda},r_{\ell,\lambda},s^*_{\ell,\lambda}/z_{t_{\ell,\lambda}},r_{\ell,\lambda})_{\mathbb{D}^*} && t_{\ell,\lambda} = p\\ &\mathbf{k}^*_{\ell,\lambda} = (\pi_{\ell,\lambda}(t_{\ell,\lambda},-1),a_{\ell,\lambda},y_{\ell,\lambda},0,s^*_{\ell,\lambda}/z_{t_{\ell,\lambda}},r_{\ell,\lambda})_{\mathbb{D}^*} && p\neq t_{\ell,\lambda} < \gamma\\ &\mathbf{k}^*_{\ell,\lambda} = (\pi_{\ell,\lambda}(t_{\ell,\lambda},-1),a_{\ell,\lambda},y_{\ell,\lambda},r_{\ell,\lambda},s^*_{\ell,\lambda}/z_{t_{\ell,\lambda}},r_{\ell,\lambda})_{\mathbb{D}^*} && p\neq t_{\ell,\lambda} \geq \gamma\end{aligned}$$

where $s^*_{\ell,\lambda}$ is either $s'_{\ell,\lambda}$, $s_{\ell,\lambda}$, or 0:

$$\begin{aligned}&s^*_{\ell,\lambda} = s'_{\ell,\lambda} && \text{if } \ell < k, \text{ or } \ell = k, t_{k,\lambda} < p\\ &s^*_{\ell,\lambda} = s_{\ell,\lambda} && \text{if } \ell = k, t_{k,\lambda} \geq p\\ &s^*_{\ell,\lambda} = 0 && \text{if } \ell > k\end{aligned}$$

When $\gamma = 1$, this is the previous game: $\mathbf{G}_{2.k.2.3.p.1.1} = \mathbf{G}_{2.k.2.3.p.1}$, whereas with $\gamma = P+1$, this is the current game: $\mathbf{G}_{2.k.2.3.p.1.P+1} = \mathbf{G}_{2.k.2.3.p.2}$.

We will gradually replace the $r_{\ell,\lambda}$ values, at the 5-th hidden position, by 0 (when $t_{\ell,\lambda} \neq p$): in this game, we deal with the case $t_{\ell,\lambda} = \gamma$, for all the ℓ-th keys. We consider a triple

$(a \cdot G_2, b \cdot G_2, c \cdot G_2)$, where $c = ab + \zeta \bmod q$ with either $\zeta = 0$ or $\zeta = 1$, which are indistinguishable under the DSDH assumption. We define the matrices

$$D = \begin{pmatrix} -p & -\gamma & 0 \\ 1 & 1 & 0 \\ 0 & a & 1 \end{pmatrix}_{1,2,5} \qquad D' = \frac{1}{\gamma - p} \times \begin{pmatrix} 1 & -1 & a \\ \gamma & -p & ap \\ 0 & 0 & \gamma - p \end{pmatrix}_{1,2,5}$$

and then $\mathbb{D} = D \cdot \mathbb{V}$, $\mathbb{D}^* = D' \cdot \mathbb{V}^*$: we cannot compute $\mathbf{d}_5$, but this component is always 0, except for $\mathbf{c}_p$ which will be defined in the original basis $\mathbb{V}$. The other ciphertexts will be directly generated in $\mathbb{D}$. Similarly, we will define all the keys $\mathbf{k}^*_{\ell,\lambda}$ for $t_{\ell,\lambda} \neq \gamma$ in $\mathbb{D}^*$, but for $t_{\ell,\lambda} = \gamma$, we choose additional random scalars $\beta_{\ell,\lambda} \stackrel{\$}{\leftarrow} \mathbb{Z}_q$, to virtually set $b_{\ell,\lambda} = r_{\ell,\lambda} \cdot b + \beta_{\ell,\lambda}$ and $c_{\ell,\lambda} = r_{\ell,\lambda} \cdot c + \beta_{\ell,\lambda} \cdot a$, then $c_{\ell,\lambda} - ab_{\ell,\lambda} = \zeta \cdot r_{\ell,\lambda}$, which is either 0 or $r_{\ell,\lambda}$. One can set

$$\begin{aligned}
\mathbf{c}_p &= (0, 0, \omega, 0, 0, \tau z_p, 0)_{\mathbb{D}} + ((p - \gamma)\sigma(0, 1), 0, 0, u_p, 0, 0)_{\mathbb{V}} \\
&= (0, 0, \omega, 0, 0, \tau z_p, 0)_{\mathbb{D}} + (\sigma(1, p) + au_p(1, p)/(\gamma - p), 0, 0, u_p, 0, 0)_{\mathbb{D}} \\
&= ((\sigma + au_p/(\gamma - p))(1, p), \omega, 0, u_p, \tau z_p, 0)_{\mathbb{D}} \\
&= (\sigma_p(1, p), \omega, 0, u_p, \tau z_p, 0)_{\mathbb{D}} \\
\mathbf{k}^*_{\ell,\lambda} &= (0, 0, a_{\ell,\lambda}, y_{\ell,\lambda}, 0, s^*_{\ell,\lambda}/z_{t_{\ell,\lambda}}, r_{\ell,\lambda})_{\mathbb{D}^*} + (b_{\ell,\lambda} \cdot (0, -1), 0, 0, c_{\ell,\lambda}, 0, 0)_{\mathbb{V}^*} \\
&= (0, 0, a_{\ell,\lambda}, y_{\ell,\lambda}, 0, s^*_{\ell,\lambda}/z_{t_{\ell,\lambda}}, r_{\ell,\lambda})_{\mathbb{D}^*} + (b_{\ell,\lambda} \cdot (\gamma, -1), 0, 0, c_{\ell,\lambda} - ab_{\ell,\lambda}, 0, 0)_{\mathbb{D}^*} \\
&= (b_{\ell,\lambda} \cdot (\gamma, -1), a_{\ell,\lambda}, y_{\ell,\lambda}, \zeta \cdot r_{\ell,\lambda}, s^*_{\ell,\lambda}/z_{t_{\ell,\lambda}}, r_{\ell,\lambda})_{\mathbb{D}^*}
\end{aligned}$$

when $t_{\ell,\lambda} = \gamma$. This is either the current game $\mathbf{G}_{2.k.2.3.p.1.\gamma}$, if $\zeta = 1$, or the next game $\mathbf{G}_{2.k.2.3.p.1.\gamma+1}$, if $\zeta = 0$.

With all this sequence, we have $\mathsf{Adv}_{2.k.2.3.p.1} - \mathsf{Adv}_{2.k.2.3.p.2} \leq 2P \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t)$.

Game $\mathbf{G}_{2.k.2.3.p.3}$: The previous game (in bases $(\mathbb{U}, \mathbb{U}^*, \mathbb{V}, \mathbb{V}^*)$) and this game (in bases $(\mathbb{B}, \mathbb{B}^*, \mathbb{D}, \mathbb{D}^*)$) are perfectly indistinguishable by using a formal change of basis, on hidden vectors, with

$$D = \left(\tfrac{\tau z_p}{u_p} \right)_5 \qquad D' = \left(\tfrac{u_p}{\tau z_p} \right)_5 \qquad \mathbb{D} = D \cdot \mathbb{V} \qquad \mathbb{D}^* = D' \cdot \mathbb{V}^*$$

The challenge ciphertext and keys that are impacted become:

$$\begin{aligned}
\mathbf{c}_p &= (\sigma_p(1, p), \omega, 0, u_p, \tau z_p, 0)_{\mathbb{V}} \\
&= (\sigma_p(1, p), \omega, 0, \tau z_p, \tau z_p, 0)_{\mathbb{D}} \\
\forall \ell, t_{\ell,\lambda} = p, \quad \mathbf{k}^*_{\ell,\lambda} &= (\pi_{\ell,\lambda}(p, -1), a_{\ell,\lambda}, y_{\ell,\lambda}, r_{\ell,\lambda}, s^*_{\ell,\lambda}/z_p, r_{\ell,\lambda})_{\mathbb{V}^*} \\
&= (\pi_{\ell,\lambda}(p, -1), a_{\ell,\lambda}, y_{\ell,\lambda}, r_{\ell,\lambda} u_p/\tau z_p, s^*_{\ell,\lambda}/z_p, r_{\ell,\lambda})_{\mathbb{D}^*}
\end{aligned}$$

All the other vectors have a zero in these components (included the $\mathsf{OEncaps}$-ciphertexts). Hence, $\mathsf{Adv}_{2.k.2.3.p.3} = \mathsf{Adv}_{2.k.2.3.p.2}$. Note however this is because of this game the security result requires the semi-adaptive super-set setting: the change of basis needs to know that $u_p \neq 0$.

Game $\mathbf{G}_{2.k.2.3.p.4}$: We keep the τz_p value (at the 5-th hidden position) in the ciphertext for the p-th attribute only, and replace all the other values from 0 to τz_t, which is the same value as in the 6-th component of each ciphertext, to allow a later swap of the key elements from the 6-th component to the 5-th:

$$\begin{aligned}
\mathbf{c}_p &= (\sigma_t(1, t), \omega, 0, \tau z_p, \tau z_p, 0)_{\mathbb{D}} \\
\mathbf{c}_t &= (\sigma_t(1, t), \omega, 0, \tau z_t, \tau z_t, u_t)_{\mathbb{D}} & t \neq p
\end{aligned}$$

To show this is possible without impacting the other vectors, we use the $\mathsf{Index\text{-}Ind}$ property from Theorem 23, but in another level of sequence of hybrid games, for $\gamma \in \{1, \ldots, P\} \backslash \{p\}$:

Game $\mathbf{G}_{2.k.2.3.p.4.\gamma}$: We consider

$$\begin{aligned}
\mathbf{c}_p &= (\sigma_p(1,p), \omega, 0, \tau z_p, \tau z_p, 0)_{\mathbb{D}} && \\
\mathbf{c}_t &= (\sigma_t(1,t), \omega, 0, \tau z_t, \tau z_t, u_t)_{\mathbb{D}} && p \neq t < \gamma \\
\mathbf{c}_t &= (\sigma_t(1,t), \omega, 0, 0, \tau z_t, u_t)_{\mathbb{D}} && p \neq t \geq \gamma \\
\mathbf{k}^*_{\ell,\lambda} &= (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, y_{\ell,\lambda}, r_{\ell,\lambda}, s^*_{\ell,\lambda}/z_{t_{\ell,\lambda}}, r_{\ell,\lambda})_{\mathbb{D}^*} && t_{\ell,\lambda} = p \\
\mathbf{k}^*_{\ell,\lambda} &= (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, y_{\ell,\lambda}, 0, s^*_{\ell,\lambda}/z_{t_{\ell,\lambda}}, r_{\ell,\lambda})_{\mathbb{D}^*} && t_{\ell,\lambda} \neq p
\end{aligned}$$

When $\gamma = 1$, this is the previous game: $\mathbf{G}_{2.k.2.3.p.4.1} = \mathbf{G}_{2.k.2.3.p.3}$, whereas with $\gamma = P+1$, this is the current game: $\mathbf{G}_{2.k.2.3.p.4.P+1} = \mathbf{G}_{2.k.2.3.p.4}$.
For any $\gamma \in \{1,\dots,P\}\backslash\{p\}$, we consider a triple $(a\cdot G_1, b\cdot G_1, c\cdot G_1)$, where $c = ab+\zeta \bmod q$ with either $\zeta = 0$ or $\zeta = \tau z_t$, which are indistinguishable under the DSDH assumption. We define the matrices

$$D = \frac{1}{p-\gamma} \times \begin{pmatrix} p & -\gamma & ap \\ -1 & 1 & -a \\ 0 & 0 & p-\gamma \end{pmatrix}_{1,2,5} \qquad D' = \begin{pmatrix} 1 & 1 & 0 \\ \gamma & p & 0 \\ -a & 0 & 1 \end{pmatrix}_{1,2,5}$$

and then $\mathbb{D} = D\cdot\mathbb{V}$, $\mathbb{D}^* = D'\cdot\mathbb{V}^*$: we cannot compute $\mathbf{d}^*_5$, but the components on this vector are all 0 except for $\mathbf{k}^*_{\ell,\lambda}$ with $t_{\ell,\lambda} = p$ which will be defined in $\mathbb{V}^*$, for all ℓ. All other keys are directly defined in $\mathbb{D}^*$, as well as the ciphertexts that are directly defined in $\mathbb{D}$ excepted $\mathbf{c}_\gamma$:

$$\begin{aligned}
\mathbf{c}_\gamma &= (0,0,\omega,0,0,\tau z_\gamma, u_\gamma)_{\mathbb{D}} + (b,0,0,0,c,0,0)_{\mathbb{V}} \\
&= (0,0,\omega,0,0,\tau z_\gamma, u_\gamma)_{\mathbb{D}} + (b,b\gamma,0,0,c-ab,0,0)_{\mathbb{D}} \\
&= (b(1,\gamma),\omega,0,\zeta,\tau z_\gamma,u_\gamma)_{\mathbb{D}} \\
\mathbf{k}^*_{\ell,\lambda} &= (0,0,a_{\ell,\lambda},y_{\ell,\lambda},0,s^*_{\ell,\lambda}/z_p,r_{\ell,\lambda})_{\mathbb{D}^*} + ((p-\gamma)\cdot(\pi,0),0,0,r_{\ell,\lambda},0,0)_{\mathbb{V}^*} \\
&= (0,0,a_{\ell,\lambda},y_{\ell,\lambda},0,s^*_{\ell,\lambda}/z_p,r_{\ell,\lambda})_{\mathbb{D}^*} + (p\pi + apr_{\ell,\lambda}, -\pi - ar_{\ell,\lambda},0,0,r_{\ell,\lambda},0,0)_{\mathbb{D}^*} \\
&= ((\pi + ar_{\ell,\lambda})\cdot(p,-1),a_{\ell,\lambda},y_{\ell,\lambda},r_{\ell,\lambda},s^*_{\ell,\lambda}/z_p,r_{\ell,\lambda})_{\mathbb{D}^*}
\end{aligned}$$

which is the hybrid game with $\pi_{\ell,\lambda} = \pi + ar_{\ell,\lambda}$ and the 5-th component of $\mathbf{c}_\gamma$ is ζ, which is either 0 and thus the game $\mathbf{G}_{2.k.2.3.p.4.\gamma}$ or τz_γ and thus the game $\mathbf{G}_{2.k.2.3.p.4.\gamma+1}$: hence, the distance between two consecutive games is bounded by $\mathsf{Adv}^{\mathsf{dsdh}}_{\mathbb{G}_1}(t)$.
With all this sequence, we have $\mathsf{Adv}_{2.k.2.3.p.3} - \mathsf{Adv}_{2.k.2.3.p.4} \leq 2P\cdot\mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t)$.

Game $\mathbf{G}_{2.k.2.3.p.5}$: All ciphertexts now have exactly the same value in 5-th and 6-th positions. We will thus use $r_{\ell,\lambda}$ in the 5-th position, for keys with $t_{\ell,\lambda} = p$, to modify the 6-th position of said keys with a swap. The previous game and this game are indistinguishable under the DDH assumption in $\mathbb{G}_2$: one essentially uses theorem 21. We consider a triple $(a\cdot G_2, b\cdot G_2, c\cdot G_2)$, where $c = ab+\zeta \bmod q$ with either $\zeta = 0$ or $\zeta = u_p/\tau z_p$, which are indistinguishable under the DSDH assumption. When we start from random dual orthogonal bases $(\mathbb{U},\mathbb{U}^*)$ and $(\mathbb{V},\mathbb{V}^*)$ of size 3 and 7 respectively, one considers the matrices:

$$D = \begin{pmatrix} 1 & 0 & 0 \\ -a & 1 & 0 \\ a & 0 & 1 \end{pmatrix}_{1,5,6} \qquad D' = \begin{pmatrix} 1 & a & -a \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}_{1,5,6} \qquad \mathbb{D} = D\cdot\mathbb{V} \qquad \mathbb{D}^* = D'\cdot\mathbb{V}^*$$

We can calculate all vectors but $\mathbf{d}_5$ and $\mathbf{d}_6$, which are not in the public key: However the challenge ciphertext computation through $\mathbb{V}$ is trivial since the 5-th and 6-th components cancel each other out. We can thus simulate them in $\mathbb{D}$.
For challenge ciphertexts, we set

$$\begin{aligned}
\mathbf{c}_p &= (\sigma_t(1,t),\omega,0,\tau z_p,\tau z_p,0)_{\mathbb{V}} = (\sigma_t(1,t),\omega,0,\tau z_p,\tau z_p,0)_{\mathbb{D}} && \\
\mathbf{c}_t &= (\sigma_t(1,t),\omega,0,\tau z_t,\tau z_t,u_t)_{\mathbb{V}} = (\sigma_t(1,t),\omega,0,\tau z_t,\tau z_t,u_t)_{\mathbb{D}} && t \neq p
\end{aligned}$$

The only keys that are calculated through $\mathbb{V}^*$ are the ones from the k-th query so that $t_{k,\lambda} = p$. We choose additional random scalars $\beta_{k,\lambda} \stackrel{\$}{\leftarrow} \mathbb{Z}_q$, to virtually set $b_{k,\lambda} = r_{k,\lambda} \cdot b + \beta_{k,\lambda}$ and $c_{k,\lambda} = r_{k,\lambda} \cdot c + \beta_{k,\lambda} \cdot a$, then $c_{k,\lambda} - ab_{k,\lambda} = \zeta \cdot r_{k,\lambda}$, which is either 0 or $r_{k,\lambda} \cdot u_p/\tau z_p$.

$$\begin{aligned}
\mathbf{k}^*_{k,\lambda} &= (0,0,a_{k,\lambda},y_{k,\lambda},0,0,r_{k,\lambda})_{\mathbb{D}^*} \\
&\qquad + (b(p,-1),0,0,0,r_{k,\lambda}\cdot u_p/\tau z_p - c_{k,\lambda}, c_{k,\lambda} + s_{k,\lambda}/z_p, 0)_{\mathbb{V}^*} \\
\mathbf{k}^*_{k,\lambda} &= (0,0,a_{k,\lambda},y_{k,\lambda},0,0,r_{k,\lambda})_{\mathbb{D}^*} \\
&\qquad + (b(p,-1),0,0,b_{k,\lambda},r_{k,\lambda}\cdot u_p/\tau z_p - (c_{k,\lambda} - ab_{k,\lambda}), \\
&\qquad\qquad (c_{k,\lambda} - ab_{k,\lambda}) + s_{k,\lambda}/z_p, 0)_{\mathbb{D}^*} \\
\mathbf{k}^*_{k,\lambda} &= (b(p,-1),a_{k,\lambda},y_{k,\lambda},r_{k,\lambda}\cdot u_p/\tau z_p - \zeta\cdot r_{k,\lambda}, \zeta\cdot r_{k,\lambda} + s_{k,\lambda}/z_p, r_{k,\lambda})_{\mathbb{D}^*}
\end{aligned}$$

If $\zeta = 0$, we are in the previous game. If $\zeta = u_p/\tau z_p$, then $\zeta \cdot r_{k,\lambda} = r_{k,\lambda} \cdot u_p/\tau z_p$ and we are in the current game. All other keys are unchanged and calculated through $\mathbb{D}^*$ directly, without any change. And, $\mathsf{Adv}_{2.k.2.3.p.4} - \mathsf{Adv}_{2.k.2.3.p.5} \leq 2 \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t)$.

Game $\mathbf{G}_{2.k.2.3.p.6}$: In this game, we want to replace $r_{k,\lambda}$ when $t_{k,\lambda} = p$ by a random value in the 7-th column, independently of the value in the 6-th column, so that this 6-th column value can be really random and independent from other values. We will exploit the random $y_{k,\lambda}$ in the 4-th column: We consider a triple $(a \cdot G_2, b \cdot G_2, c \cdot G_2)$, where $c = ab + \zeta \bmod q$ with either $\zeta = 0$ or $\zeta \stackrel{\$}{\leftarrow} \mathbb{Z}_q^*$, which are indistinguishable under the DDH assumption. We choose additional random scalars $\alpha_\lambda, \beta_\lambda \stackrel{\$}{\leftarrow} \mathbb{Z}_q$, to virtually set $b_\lambda = \alpha_\lambda \cdot b + \beta_\lambda$ and $c_\lambda = \alpha_\lambda \cdot c + \beta_\lambda \cdot a$, then $c_\lambda - ab_\lambda = \zeta \cdot \alpha_\lambda$, which are either 0 or independent random values. When we start from random dual orthogonal bases $(\mathbb{U}, \mathbb{U}^*)$ and $(\mathbb{V}, \mathbb{V}^*)$ of size 3 and 7 respectively, one considers the matrices:

$$D = \begin{pmatrix} 1 & 0 \\ -a & 1 \end{pmatrix}_{4,7} \qquad D' = \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}_{4,7} \qquad \mathbb{D} = D \cdot \mathbb{V} \qquad \mathbb{D}^* = D' \cdot \mathbb{V}^*$$

We can calculate all vectors but $\mathbf{d}_7$, which is not in the public key. Through $\mathbb{V}$, we calculate the challenge ciphertext, and the OEncaps-answers, when the 7-th component is non-zero, as the 0 value of the 4-th component does not impact the 7-th during the change of basis.
On the other hand, all the keys can be directly generated in $\mathbb{D}^*$, except $\mathbf{k}_{k,\lambda}$ when $t_{k,\lambda} = p$, for which we use the DDH assumption:

$$\begin{aligned}
\mathbf{k}^*_{k,\lambda} &= (\pi_{k,\lambda}(p,-1), a_{k,\lambda}, 0, 0, \frac{s_{k,\lambda} + r_{k,\lambda}\cdot u_p/\tau}{z_p}, r_{k,\lambda})_{\mathbb{D}^*} \\
&\quad + (0,0,0,b_\lambda,0,0,c_\lambda)_{\mathbb{V}^*} \\
&= (\pi_{k,\lambda}(p,-1), a_{k,\lambda}, 0, 0, \frac{s_{k,\lambda} + r_{k,\lambda}\cdot u_p/\tau}{z_p}, r_{k,\lambda})_{\mathbb{D}^*} \\
&\quad + (0,0,0,b_\lambda,0,0,c_\lambda - ab_\lambda)_{\mathbb{D}^*} \\
&= (\pi_{k,\lambda}(p,-1), a_{k,\lambda}, b_\lambda, 0, \frac{s_{k,\lambda} + r_{k,\lambda}\cdot u_p/\tau}{z_p}, r_{k,\lambda} + \zeta\cdot\alpha_\lambda)_{\mathbb{V}^*}
\end{aligned}$$

When $\zeta = 0$, this is the previous game, with $y_{k,\lambda} = b_\lambda$, when $t_{k,\lambda} = p$. Whereas when $\zeta \stackrel{\$}{\leftarrow} \mathbb{Z}_q^*$, $r'_{k,\lambda} = r_{k,\lambda} + \zeta \cdot \alpha_\lambda$ is independent of $r_{k,\lambda}$, which makes $s'_{k,\lambda} = (s_{k,\lambda} + r_{k,\lambda} \cdot u_p/\tau)/z_p$ independent of $s_{k,\lambda}$ when $r_{k,\lambda} \cdot u_p \neq 0$. Then, $\mathsf{Adv}_{2.k.2.3.p.5} - \mathsf{Adv}_{2.k.2.3.p.6} \leq \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t)$.
In order to keep the same $r_{\ell,\lambda}$ for all the leaves delegated from the same initial key, we also apply this additional vector $(0,0,0,b_\lambda,0,0,c_\lambda)_{\mathbb{V}^*}$. This will also keep the same $y_{\ell,\lambda}$ for all these leaves.

Game $\mathbf{G}_{2.k.2.3.p.7}$: All ciphertexts have exactly the same value in 5-th and 6-th positions. We will thus use the Swap-Ind property to revert the change made in game $\mathbf{G}_{2.k.2.3.p.5}$, with the notable difference we are now working with $r'_{k,\lambda}$ (which has just been randomized) instead of $r_{k,\lambda}$, for keys with $t_{k,\lambda} = p$. We are thus not restoring the initial $s_{k,\lambda}$ but we get a truly random value $s'_{k,\lambda}$. The previous game and this game are indistinguishable under the DDH assumption in $\mathbb{G}_2$: one essentially uses theorem 21. We consider a triple $(a \cdot G_2, b \cdot G_2, c \cdot G_2)$, where $c = ab + \zeta \bmod q$ with either $\zeta = 0$ or $\zeta = u_p/\tau z_p$, which are indistinguishable under the DSDH assumption. When we start from random dual orthogonal bases $(\mathbb{U}, \mathbb{U}^*)$ and $(\mathbb{V}, \mathbb{V}^*)$ of size 3 and 7 respectively, one considers the matrices:

$$D = \begin{pmatrix} 1 & 0 & 0 \\ -a & 1 & 0 \\ a & 0 & 1 \end{pmatrix}_{4,5,6} \qquad D' = \begin{pmatrix} 1 & a & -a \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}_{4,5,6} \qquad \mathbb{D} = D \cdot \mathbb{V} \qquad \mathbb{D}^* = D' \cdot \mathbb{V}^*$$

We can calculate all vectors but $\mathbf{d}_5$ and $\mathbf{d}_6$, which are not in the public key: However, the challenge ciphertext computation through $\mathbb{V}$ is trivial since the 5-th and 6-th component cancel each other out. We can thus simulate them through $\mathbb{V}$. We can revert as above by setting in $\mathbb{V}^*$ the keys from the k-th query so that $t_{k,\lambda} = p$. And, $\mathsf{Adv}_{2.k.2.3.p.6} - \mathsf{Adv}_{2.k.2.3.p.7} \leq 2 \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t)$. We stress that after the swap, we get, for $t_{k,\lambda} = p$

$$\mathbf{k}^*_{k,\lambda} = (\pi_{k,\lambda}(p, -1), a_{k,\lambda}, y_{k,\lambda}, r'_{k,\lambda} u_p/\tau z_p, (s'_{k,\lambda} - r'_{k,\lambda} u_p/\tau)/z_p, r'_{k,\lambda})_{\mathbb{D}^*}$$

where $s'_{k,\lambda}$ is a truly random value independent of $r'_{k,\lambda}$. So we are not back to game $\mathbf{G}_{2.k.2.3.p.4}$, but still with a random value in the 6-th component of the key.

Game $\mathbf{G}_{2.k.2.3.p.8}$: We keep the τz_p value (at the 5-th hidden position) in the ciphertext for the p-th attribute only, and replace all the other values from τz_t to 0

$$\begin{aligned} \mathbf{c}_p &= (\sigma_t(1,t), \omega, 0, \tau z_p, \tau z_p, 0)_{\mathbb{D}} \\ \mathbf{c}_t &= (\sigma_t(1,t), \omega, 0, 0, \tau z_t, u_t)_{\mathbb{D}} && t \neq p \end{aligned}$$

To show this is possible without impacting the other vectors, we use the Index-Ind property from Theorem 23, but in another level of sequence of hybrid games, for $\gamma \in \{1, \ldots, P\} \backslash \{p\}$:

Game $\mathbf{G}_{2.k.2.3.p.8.\gamma}$: We consider

$$\begin{aligned} \mathbf{c}_p &= (\sigma_p(1,p), \omega, 0, \tau z_p, \tau z_p, 0)_{\mathbb{D}} \\ \mathbf{c}_t &= (\sigma_t(1,t), \omega, 0, 0, \tau z_t, u_t)_{\mathbb{D}} && p \neq t < \gamma \\ \mathbf{c}_t &= (\sigma_t(1,t), \omega, 0, \tau z_t, \tau z_t, u_t)_{\mathbb{D}} && p \neq t \geq \gamma \\ \mathbf{k}^*_{\ell,\lambda} &= (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, y_{\ell,\lambda}, r'_{\ell,\lambda}, s^*_{\ell,\lambda}/z_{t_{\ell,\lambda}}, r'_{\ell,\lambda})_{\mathbb{D}^*} && t_{\ell,\lambda} = p \\ \mathbf{k}^*_{\ell,\lambda} &= (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, y_{\ell,\lambda}, 0, s^*_{\ell,\lambda}/z_{t_{\ell,\lambda}}, r_{\ell,\lambda})_{\mathbb{D}^*} && t_{\ell,\lambda} \neq p \end{aligned}$$

When $\gamma = 1$, this is the previous game: $\mathbf{G}_{2.k.2.3.p.8.1} = \mathbf{G}_{2.k.2.3.p.7}$, whereas with $\gamma = P+1$, this is the current game: $\mathbf{G}_{2.k.2.3.p.8.P+1} = \mathbf{G}_{2.k.2.3.p.8}$.
For any $\gamma \in \{1, \ldots, P\} \backslash \{p\}$, we consider a triple $(a \cdot G_1, b \cdot G_1, c \cdot G_1)$, where $c = ab + \zeta \bmod q$ with either $\zeta = 0$ or $\zeta = \tau z_t$, which are indistinguishable under the DSDH assumption. We define the matrices

$$D = \frac{1}{p - \gamma} \times \begin{pmatrix} p & -\gamma & ap \\ -1 & 1 & -a \\ 0 & 0 & p - \gamma \end{pmatrix}_{1,2,5} \qquad D' = \begin{pmatrix} 1 & 1 & 0 \\ \gamma & p & 0 \\ -a & 0 & 1 \end{pmatrix}_{1,2,5}$$

and then $\mathbb{D} = D \cdot \mathbb{V}$, $\mathbb{D}^* = D' \cdot \mathbb{V}^*$: we cannot compute $\mathbf{d}^*_5$, but the components on this vector are all 0 except for $\mathbf{k}^*_{\ell,\lambda}$ with $t_{\ell,\lambda} = p$ which will be defined in $\mathbb{V}^*$, for all ℓ. All other keys are directly defined in $\mathbb{D}^*$, as well as the ciphertexts that are directly defined in $\mathbb{D}$ excepted $\mathbf{c}_\gamma$:

$$\begin{aligned}
\mathbf{c}_\gamma &= (0,0,\omega,0,0,\tau z_\gamma,u_\gamma)_{\mathbb{D}} + (b,0,0,0,c,0,0)_{\mathbb{V}}\\
&= (0,0,\omega,0,0,\tau z_\gamma,u_\gamma)_{\mathbb{D}} + (b,b\gamma,0,0,c-ab,0,0)_{\mathbb{D}}\\
&= (b(1,\gamma),\omega,0,\zeta,\tau z_\gamma,u_\gamma)_{\mathbb{D}}\\
\mathbf{k}^*_{\ell,\lambda} &= (0,0,a_{\ell,\lambda},y_{\ell,\lambda},0,s^*_{\ell,\lambda}/z_p,r_{\ell,\lambda})_{\mathbb{D}^*} + ((p-\gamma)\cdot(\pi,0),0,0,r_{\ell,\lambda},0,0)_{\mathbb{V}^*}\\
&= (0,0,a_{\ell,\lambda},y_{\ell,\lambda},0,s^*_{\ell,\lambda}/z_p,r_{\ell,\lambda})_{\mathbb{D}^*} + (p\pi+apr_{\ell,\lambda},-\pi-ar_{\ell,\lambda},0,0,r_{\ell,\lambda},0,0)_{\mathbb{D}^*}\\
&= ((\pi+ar_{\ell,\lambda})\cdot(p,-1),a_{\ell,\lambda},y_{\ell,\lambda},r_{\ell,\lambda},s^*_{\ell,\lambda}/z_p,r_{\ell,\lambda})_{\mathbb{D}^*}
\end{aligned}$$

which is the hybrid game with $\pi_{\ell,\lambda} = \pi + ar_{\ell,\lambda}$ and the 5-th component of $\mathbf{c}_\gamma$ is ζ, which is either τz_t and thus the game $\mathbf{G}_{2.k.2.3.p.8.\gamma}$ or 0 and thus the game $\mathbf{G}_{2.k.2.3.p.8.\gamma+1}$: hence, the distance between two consecutive games is bounded by $\mathsf{Adv}^{\mathsf{dsdh}}_{\mathbb{G}_1}(t)$.

With all this sequence, we have $\mathsf{Adv}_{2.k.2.3.p.7} - \mathsf{Adv}_{2.k.2.3.p.8} \leq 2P \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t)$.

Game $\mathbf{G}_{2.k.2.3.p.9}$: The previous game (in bases $(\mathbb{U},\mathbb{U}^*,\mathbb{V},\mathbb{V}^*)$) and this game (in bases $(\mathbb{B},\mathbb{B}^*,\mathbb{D},\mathbb{D}^*)$) are perfectly indistinguishable by using a formal change of basis, on hidden vectors, with

$$D = \begin{pmatrix} \frac{u_p}{\tau z_p} \end{pmatrix}_5 \qquad D' = \begin{pmatrix} \frac{\tau z_p}{u_p} \end{pmatrix}_5 \qquad \mathbb{D} = D\cdot\mathbb{V} \qquad \mathbb{D}^* = D'\cdot\mathbb{V}^*$$

The challenge ciphertext and keys that are impacted become:

$$\begin{aligned}
\mathbf{c}_p &= (\sigma_p(1,p),\omega,0,\tau z_p,\tau z_p,0)_{\mathbb{V}}\\
&= (\sigma_p(1,p),\omega,0,u_p,\tau z_p,0)_{\mathbb{D}}\\
\forall \ell, t_{\ell,\lambda}=p,\quad \mathbf{k}^*_{\ell,\lambda} &= (\pi_{\ell,\lambda}(p,-1),a_{\ell,\lambda},y_{\ell,\lambda},r_{\ell,\lambda}\cdot u_p/\tau z_p,s^*_{\ell,\lambda}/z_p,r_{\ell,\lambda})_{\mathbb{D}^*}\\
&= (\pi_{\ell,\lambda}(p,-1),a_{\ell,\lambda},y_{\ell,\lambda},r_{\ell,\lambda},s^*_{\ell,\lambda}/z_p,r_{\ell,\lambda})_{\mathbb{V}^*}
\end{aligned}$$

All the other vectors have a zero in these components (included the $\mathsf{OEncaps}$-ciphertexts). Hence, $\mathsf{Adv}_{2.k.2.3.p.9} = \mathsf{Adv}_{2.k.2.3.p.8}$.

Game $\mathbf{G}_{2.k.2.3.p.10}$: We keep the $r'_{\ell,\lambda}$ value (at the 5-th hidden position) in the keys such that $t_{\ell,\lambda} = p$, and replace back the 0 in all other keys by $r_{\ell,\lambda}$, in order to prepare the possibility to later modify the ciphertexts on this component. To show this is possible without impacting the other vectors, we use the $\mathsf{Index\text{-}Ind}$ property from Theorem 23, but in another level of sequence of hybrid games, for $\gamma \in \{1,\ldots,P\}\backslash\{p\}$:

Game $\mathbf{G}_{2.k.2.3.p.9.\gamma}$: We consider the following hybrid game, where the first satisfied condition on the indices is applied:

$$\mathbf{c}_p = (\sigma_p(1,p),\omega,0,u_p,\tau z_p,0)_{\mathbb{D}} \qquad \mathbf{c}_t = (\sigma_t(1,t),\omega,0,0,\tau z_t,u_t)_{\mathbb{D}} \qquad t\neq p$$

$$\begin{aligned}
\mathbf{k}^*_{\ell,\lambda} &= (\pi_{\ell,\lambda}(t_{\ell,\lambda},-1),a_{\ell,\lambda},y_{\ell,\lambda},r_{\ell,\lambda},s^*_{\ell,\lambda}/z_{t_{\ell,\lambda}},r_{\ell,\lambda})_{\mathbb{D}^*} && t_{\ell,\lambda}=p\\
\mathbf{k}^*_{\ell,\lambda} &= (\pi_{\ell,\lambda}(t_{\ell,\lambda},-1),a_{\ell,\lambda},y_{\ell,\lambda},r_{\ell,\lambda},s^*_{\ell,\lambda}/z_{t_{\ell,\lambda}},r_{\ell,\lambda})_{\mathbb{D}^*} && p\neq t_{\ell,\lambda}<\gamma\\
\mathbf{k}^*_{\ell,\lambda} &= (\pi_{\ell,\lambda}(t_{\ell,\lambda},-1),a_{\ell,\lambda},y_{\ell,\lambda},0,s^*_{\ell,\lambda}/z_{t_{\ell,\lambda}},r_{\ell,\lambda})_{\mathbb{D}^*} && p\neq t_{\ell,\lambda}\geq\gamma
\end{aligned}$$

where $s^*_{\ell,\lambda}$ is either $s'_{\ell,\lambda}$, $s_{\ell,\lambda}$, or 0
When $\gamma = 1$, this is the previous game: $\mathbf{G}_{2.k.2.3.p.9.1} = \mathbf{G}_{2.k.2.3.p.9}$, whereas with $\gamma = P+1$, this is the current game: $\mathbf{G}_{2.k.2.3.p.9.P+1} = \mathbf{G}_{2.k.2.3.p.10}$.
We will gradually replace the 0 values, at the 5-th hidden position, by $r_{\ell,\lambda}$ (when $t_{\ell,\lambda} \neq p$): in this game, we deal with the case $t_{\ell,\lambda} = \gamma$, for all the ℓ-th keys. We consider a triple

$(a \cdot G_2, b \cdot G_2, c \cdot G_2)$, where $c = ab + \zeta \bmod q$ with either $\zeta = 0$ or $\zeta = 1$, which are indistinguishable under the DSDH assumption. We define the matrices

$$D = \begin{pmatrix} -p & -\gamma & 0 \\ 1 & 1 & 0 \\ 0 & a & 1 \end{pmatrix}_{1,2,5} \qquad D' = \frac{1}{\gamma - p} \times \begin{pmatrix} 1 & -1 & a \\ \gamma & -p & ap \\ 0 & 0 & \gamma - p \end{pmatrix}_{1,2,5}$$

and then $\mathbb{D} = D \cdot \mathbb{V}$, $\mathbb{D}^* = D' \cdot \mathbb{V}^*$: we cannot compute $\mathbf{d}_5$, but this component is always 0, except for $\mathbf{c}_p$ which will be defined in the original basis $\mathbb{V}$. The other ciphertexts will be directly generated in $\mathbb{D}$. Similarly, we will define all the keys $\mathbf{k}^*_{\ell,\lambda}$ for $t_{\ell,\lambda} \neq \gamma$ in $\mathbb{D}^*$, but for $t_{\ell,\lambda} = \gamma$, we choose additional random scalars $\beta_{\ell,\lambda} \xleftarrow{\$} \mathbb{Z}_q$, to virtually set $b_{\ell,\lambda} = r_{\ell,\lambda} \cdot b + \beta_{\ell,\lambda}$ and $c_{\ell,\lambda} = r_{\ell,\lambda} \cdot c + \beta_{\ell,\lambda} \cdot a$, then $c_{\ell,\lambda} - ab_{\ell,\lambda} = \zeta \cdot r_{\ell,\lambda}$, which is either 0 or $r_{\ell,\lambda}$. One can set

$$\begin{aligned}
\mathbf{c}_p &= (0,0,\omega,0,0,\tau z_p,0)_{\mathbb{D}} + ((p-\gamma)\sigma(0,1),0,0,u_p,0,0)_{\mathbb{V}} \\
&= (0,0,\omega,0,0,\tau z_p,0)_{\mathbb{D}} + (\sigma(1,p) + au_p(1,p)/(\gamma - p),0,0,u_p,0,0)_{\mathbb{D}} \\
&= ((\sigma + au_p/(\gamma-p))(1,p),\omega,0,u_p,\tau z_p,0)_{\mathbb{D}} \\
&= (\sigma_p(1,p),\omega,0,u_p,\tau z_p,0)_{\mathbb{D}} \\
\mathbf{k}^*_{\ell,\lambda} &= (0,0,a_{\ell,\lambda},y_{\ell,\lambda},0,s^*_{\ell,\lambda}/z_{t_{\ell,\lambda}},r_{\ell,\lambda})_{\mathbb{D}^*} + (b_{\ell,\lambda}\cdot(0,-1),0,0,c_{\ell,\lambda},0,0)_{\mathbb{V}^*} \\
&= (0,0,a_{\ell,\lambda},y_{\ell,\lambda},0,s^*_{\ell,\lambda}/z_{t_{\ell,\lambda}},r_{\ell,\lambda})_{\mathbb{D}^*} + (b_{\ell,\lambda}\cdot(\gamma,-1),0,0,c_{\ell,\lambda} - ab_{\ell,\lambda},0,0)_{\mathbb{D}^*} \\
&= (b_{\ell,\lambda}\cdot(\gamma,-1),a_{\ell,\lambda},y_{\ell,\lambda},\zeta\cdot r_{\ell,\lambda},s^*_{\ell,\lambda}/z_{t_{\ell,\lambda}},r_{\ell,\lambda})_{\mathbb{D}^*}
\end{aligned}$$

when $t_{\ell,\lambda} = \gamma$. This is either the current game $\mathbf{G}_{2.k.2.3.p.9.\gamma}$, if $\zeta = 0$, or the next game $\mathbf{G}_{2.k.2.3.p.9.\gamma+1}$, if $\zeta = 1$.

With all this sequence, we have $\mathsf{Adv}_{2.k.2.3.p.9} - \mathsf{Adv}_{2.k.2.3.p.10} \leq 2P \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t)$.

Game $\mathbf{G}_{2.k.2.3.p.11}$: The previous game and this game are indistinguishable under the DDH assumption in $\mathbb{G}_1$: one essentially uses theorem 21. Given a tuple $(a \cdot G_1, b \cdot G_1, c \cdot G_1)$ in $\mathbb{G}_1$, where $c = ab + \mu \bmod q$ with either $\mu = 0$ or $\mu = u_p$, the 5-th component of the p-th ciphertext. When we start from random dual orthogonal bases $(\mathbb{U}, \mathbb{U}^*)$ and $(\mathbb{V}, \mathbb{V}^*)$ of size 3 and 7 respectively, one considers the matrices:

$$D = \begin{pmatrix} 1 & a & -a \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}_{1,5,7} \qquad D' = \begin{pmatrix} 1 & 0 & 0 \\ -a & 1 & 0 \\ a & 0 & 1 \end{pmatrix}_{1,5,7} \qquad \mathbb{D} = D \cdot \mathbb{V} \qquad \mathbb{D}^* = D' \cdot \mathbb{V}^*$$

We can calculate all vectors but $\mathbf{d}^*_5$ and $\mathbf{d}^*_7$, which are not in the public key. Through $\mathbb{V}$, we calculate the challenge ciphertext for the attribute of the p-th ciphertext

$$\begin{aligned}
\mathbf{c}_p &= (0,0,\omega,0,0,\tau z_p,u_p)_{\mathbb{D}} + (b(1,p),0,0,c,0,-c)_{\mathbb{V}} \\
&= (0,0,\omega,0,0,\tau z_p,u_p)_{\mathbb{D}} + (b(1,p),0,0,c-ab,0,ab-c)_{\mathbb{D}} \\
&= (b(1,p),\omega,0,\mu,\tau z_p,u_p - \mu)_{\mathbb{D}}
\end{aligned}$$

If $\mu = u_p$, we are in the previous game. If $\mu = 0$, then we are in the current game. Then, every other ciphertext is computed directly in $\mathbb{D}$:

$$\forall t \neq p, \mathbf{c}_t = (\sigma_t(1,t),\omega,0,0,\tau z_t,u_t)_{\mathbb{D}}$$

as well as the answers to $\mathsf{OEncaps}$-queries. The keys are calculated through $\mathbb{V}^*$ but are unchanged by the change of basis because the 5-th and 7-th components are exactly the same for every key query ℓ, and thus cancel themselves in the 1st component. We thus have $\mathsf{Adv}_{2.k.2.3.p.10} - \mathsf{Adv}_{2.k.2.3.p.11} \leq 2 \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t)$.

C.3 Proof of Theorem 12 – dKey-IND-Security

Proof. In this security game, the adversary has access to the $\mathsf{OEncaps}$-oracle, but only for distinct key-indistinguishability: all the invalid attributes $t \in \Gamma_{m,i}$ in a $\mathsf{OEncaps}$-query correspond to passive leaves $\lambda \in \mathcal{L}_p$ from the challenge key. We will prove it as usual with a sequence of games:

Game $\mathbf{G}_0$: The first game is the real game where the simulator plays the role of the challenger, with $\mathsf{PK} = \{(\mathbf{b}_1, \mathbf{b}_3, \mathbf{b}_1^*), (\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_1^*, \mathbf{d}_2^*, \mathbf{d}_3^*, \mathbf{d}_7^*)\}$, $\mathsf{SK} = \{\mathbf{d}_7\}$, and $\mathsf{MK} = \{\mathbf{b}_3^*\}$, from random dual orthogonal bases. We note that $\mathbf{d}_7^*$ can be public.

$\mathsf{OKeyGen}(\tilde{\mathcal{T}}_\ell)$ (or $\mathsf{ODelegate}$-queries): The adversary is allowed to issue KeyGen-queries on an access-tree $\tilde{\mathcal{T}}_\ell = (\mathcal{T}_\ell, \mathcal{L}_{\ell,a}, \mathcal{L}_{\ell,p})$ (for the ℓ-th query), for which the simulator chooses a random scalar $a_{\ell,0} \xleftarrow{\$} \mathbb{Z}_q$ and a random $a_{\ell,0}$-labeling $(a_{\ell,\lambda})_\lambda$ of the access-tree $\mathcal{T}_\ell$, and builds the key:

$$\mathbf{k}_{\ell,0}^* = (a_{\ell,0}, 0, 1)_{\mathbb{B}^*} \qquad \mathbf{k}_{\ell,\lambda}^* = (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, 0, 0, r_{\ell,\lambda})_{\mathbb{D}^*}$$

for all the leaves λ, where $t_{\ell,\lambda} = A(\lambda)$, $\pi_{\ell,\lambda} \xleftarrow{\$} \mathbb{Z}_q$ and $r_{\ell,\lambda} \xleftarrow{\$} \mathbb{Z}_q^*$ if $\lambda \in \mathcal{L}_{\ell,a}$, or else $r_{\ell,\lambda} \leftarrow 0$ if $\lambda \in \mathcal{L}_{\ell,p}$. The decryption key dk_ℓ is then $(\mathbf{k}_{\ell,0}^*, (\mathbf{k}_{\ell,\lambda}^*)_\lambda)$;

$\mathsf{OEncaps}(\Gamma_{m,v}, \Gamma_{m,i})$: The adversary is allowed to issue Encaps^*-queries on disjoint unions $\Gamma_m = \Gamma_{m,v} \cup \Gamma_{m,i}$ of sets of attributes, for which the simulator chooses random scalars $\omega_m, \xi_m \xleftarrow{\$} \mathbb{Z}_q$. It then sets $K_m = g_t^{\xi_m}$ and generates the ciphertext $C_m = (\mathbf{c}_{m,0}, (\mathbf{c}_{m,t})_{t \in (\Gamma_{m,v} \cup \Gamma_{m,i})})$ where

$$\mathbf{c}_{m,0} = (\omega_m, 0, \xi_m)_{\mathbb{B}} \qquad \mathbf{c}_{m,t} = (\sigma_{m,t}(t, -1), \omega_m, 0, 0, 0, u_{m,t})_{\mathbb{D}}$$

for all the attributes $t \in \Gamma_{m,v} \cup \Gamma_{m,i}$, $\sigma_{m,t} \xleftarrow{\$} \mathbb{Z}_q$ and $u_{m,t} \xleftarrow{\$} \mathbb{Z}_q^*$ if $t \in \Gamma_{m,i}$ or $u_{m,t} \leftarrow 0$ if $t \in \Gamma_{m,v}$.

$\mathsf{RoAPKeyGen}(\tilde{\mathcal{T}}, \mathcal{L}_a, \mathcal{L}_p)$: On the unique query on an access-tree $\tilde{\mathcal{T}}$ of its choice, with a list $\mathcal{L} = (\mathcal{L}_a \cup \mathcal{L}_p)$ of active and passive leaves, the simulator chooses a random scalar $a_0 \xleftarrow{\$} \mathbb{Z}_q$, and a random a_0-labeling $(a_\lambda)_\lambda$ of the access-tree. It then sets the real key dk_0 as follows, with $r_\lambda \xleftarrow{\$} \mathbb{Z}_q^*$ if $\lambda \in \mathcal{L}_a$, or $r_\lambda \leftarrow 0$ if $\lambda \in \mathcal{L}_p$:

$$\mathbf{k}_0^* = (a_0, 0, 1)_{\mathbb{B}^*} \qquad \mathbf{k}_\lambda^* = (\pi_\lambda(t_\lambda, -1), a_\lambda, 0, 0, 0, r_\lambda)_{\mathbb{D}^*}$$

On the other hand, it sets the all-passive key dk_1 as:

$$\mathbf{k}_0^* = (a_0, 0, 1)_{\mathbb{B}^*} \qquad \mathbf{k}_\lambda^* = (\pi_\lambda(t_\lambda, -1), a_\lambda, 0, 0, 0, 0)_{\mathbb{D}^*}$$

for all λ. According to the real or all-passive ($b \xleftarrow{\$} \{0,1\}$), one outputs dk_b.

From the adversary's guess b' for b, one forwards it as the output β, unless for some $(\Gamma_{m,v}, \Gamma_{m,i})$ asked to the $\mathsf{OEncaps}$-oracle, some active leaf $\lambda \in \mathcal{L}_a$ from the challenge key corresponds to some invalid attribute $t \in \Gamma_{m,i}$, in which case one outputs a random $\beta \xleftarrow{\$} \{0,1\}$. We denote $\mathsf{Adv}_0 = \Pr[\beta = 1 | b = 1] - \Pr[\beta = 1 | b = 0]$.

We stress that in this distinct key-indistinguishability security game, the active keys in the challenge key ($\lambda \in \mathcal{L}_a$ with possibly $r_\lambda \neq 0$) correspond to valid ciphertexts only ($t \in \Gamma_{m,i}$ with $u_{m,t} = 0$, for all queries). But we do not exclude accepting access-trees.

Game $\mathbf{G}_1$: In the second and final game, we set $r_\lambda \leftarrow 0$ for all the leaves in the real key dk_0:

$$\mathbf{k}_0^* = (a_0, 0, 0)_{\mathbb{B}^*} \qquad \mathbf{k}_\lambda^* = (\pi_\lambda(t_\lambda, -1), a_\lambda, 0, 0, 0, 0)_{\mathbb{D}^*}$$

It is then clear than $\mathsf{Adv}_1 = 0$, as all challenge keys are independent from b.

We detail the sub-sequence starting from $\mathbf{G}_{0.p.0}$ to prove the indistinguishability between $\mathbf{G}_0$ and $\mathbf{G}_1$. In the new hybrid sequence $\mathbf{G}_{0.p.*}$, we will modify all the keys associated to the p-th attribute, in an indistinguishable way, using the $\mathsf{Index\text{-}Ind}$ property. It is clear that $\mathbf{G}_{0.1.0} = \mathbf{G}_0$, whereas $\mathbf{G}_{0.P+1.0} = \mathbf{G}_1$, and $\mathbf{G}_{0.p.4} = \mathbf{G}_{0.p+1.0}$.

$$\mathbf{c}_0 = (\omega \quad 0 \quad \xi) \qquad\qquad \mathbf{k}^*_{\ell,0} = (a_{\ell,0} \quad 0 \quad 1)$$

$\mathbf{G}_{0.p.0}$ Hybrid game for $\mathbf{G}_0$, with $1 \leq p \leq P+1$, such that $u_{m,p} = 0$ for all m

$$\begin{array}{lrl}
 & \mathbf{c}_{m,t} = & (\ \sigma_{m,t}(1,t) \quad \omega_m \mid 0 \quad 0 \quad 0 \quad u_{m,t}\ ) \\
t_\lambda < p & \mathbf{k}^*_\lambda = & (\ \pi_\lambda(t_\lambda,-1) \quad a_\lambda \mid 0 \quad 0 \quad 0 \quad 0\ ) \\
t_\lambda \geq p & \mathbf{k}^*_\lambda = & (\ \pi_\lambda(t_\lambda,-1) \quad a_\lambda \mid 0 \quad 0 \quad 0 \quad r_\lambda\ )
\end{array}$$

$\mathbf{G}_{0.p.1}$ Formal basis change, on $(\mathbb{D}, \mathbb{D}^*)_{6,7}$, to duplicate $u_{m,t}$ in the 6-th column

$$\begin{array}{lrl}
 & \mathbf{c}_{m,t} = & (\ \sigma_{m,t}(1,t) \quad \omega_m \mid 0 \quad 0 \quad u_{m,t} \quad u_{m,t}\ ) \\
t_\lambda < p & \mathbf{k}^*_\lambda = & (\ \pi_\lambda(t_\lambda,-1) \quad a_\lambda \mid 0 \quad 0 \quad 0 \quad 0\ ) \\
t_\lambda \geq p & \mathbf{k}^*_\lambda = & (\ \pi_\lambda(t_\lambda,-1) \quad a_\lambda \mid 0 \quad 0 \quad 0 \quad r_\lambda\ )
\end{array}$$

$\mathbf{G}_{0.p.2}$ Swap-Ind Property, on $(\mathbb{D}, \mathbb{D}^*)_{1,6,7}$, to swap r_λ, for $t_\lambda = p$, in the 6-th column

$$\begin{array}{lrl}
 & \mathbf{c}_{m,t} = & (\ \sigma_{m,t}(1,t) \quad \omega_m \mid 0 \quad 0 \quad u_{m,t} \quad u_{m,t}\ ) \\
t_\lambda < p & \mathbf{k}^*_\lambda = & (\ \pi_\lambda(t_\lambda,-1) \quad a_\lambda \mid 0 \quad 0 \quad 0 \quad 0\ ) \\
t_\lambda = p & \mathbf{k}^*_\lambda = & (\ \pi_\lambda(p,-1) \quad a_\lambda \mid 0 \quad 0 \quad r_\lambda \quad 0\ ) \\
t_\lambda > p & \mathbf{k}^*_\lambda = & (\ \pi_\lambda(t_\lambda,-1) \quad a_\lambda \mid 0 \quad 0 \quad 0 \quad r_\lambda\ )
\end{array}$$

$\mathbf{G}_{0.p.3}$ Index-Ind Property, on $(\mathbb{D}, \mathbb{D}^*)_{1,2,6}$, between $u_{m,t}$ and 0, for $t \neq p$

$$\begin{array}{lrl}
 & \mathbf{c}_{m,p} = & (\ \sigma_{m,t}(1,t) \quad \omega_m \mid 0 \quad 0 \quad 0 \quad 0\ ) \\
t \neq p & \mathbf{c}_{m,t} = & (\ \sigma_{m,t}(1,t) \quad \omega_m \mid 0 \quad 0 \quad 0 \quad u_{m,t}\ ) \\
t_\lambda < p & \mathbf{k}^*_\lambda = & (\ \pi_\lambda(t_\lambda,-1) \quad a_\lambda \mid 0 \quad 0 \quad 0 \quad 0\ ) \\
t_\lambda = p & \mathbf{k}^*_\lambda = & (\ \pi_\lambda(p,-1) \quad a_\lambda \mid 0 \quad 0 \quad r_\lambda \quad 0\ ) \\
t_\lambda > p & \mathbf{k}^*_\lambda = & (\ \pi_\lambda(t_\lambda,-1) \quad a_\lambda \mid 0 \quad 0 \quad 0 \quad r_\lambda\ )
\end{array}$$

$\mathbf{G}_{0.p.4}$ SubSpace-Ind Property, on $(\mathbb{D}^*, \mathbb{D})_{1,6}$, between u_p and 0

$$\begin{array}{lrl}
 & \mathbf{c}_{m,t} = & (\ \sigma_{m,t}(1,t) \quad \omega_m \mid 0 \quad 0 \quad 0 \quad u_{m,t}\ ) \\
t_\lambda < p & \mathbf{k}^*_\lambda = & (\ \pi_\lambda(t_\lambda,-1) \quad a_\lambda \mid 0 \quad 0 \quad 0 \quad 0\ ) \\
t_\lambda = p & \mathbf{k}^*_\lambda = & (\ \pi_\lambda(p,-1) \quad a_\lambda \mid 0 \quad 0 \quad 0 \quad 0\ ) \\
t_\lambda > p & \mathbf{k}^*_\lambda = & (\ \pi_\lambda(t_\lambda,-1) \quad a_\lambda \mid 0 \quad 0 \quad 0 \quad r_\lambda\ )
\end{array}$$

Fig. 15: Sub-sequence of games for Distinct Key-Indistinguishability

Game $\mathbf{G}_{0.p.0}$: One defines the hybrid game for p :

$$\begin{array}{rl}
 & \mathbf{c}_{m,t} = (\sigma_{m,t}(1,t), \omega_m, 0, 0, 0, u_{m,t})_{\mathbb{D}} \\
t_\lambda < p & \mathbf{k}^*_\lambda = (\pi_\lambda(t_\lambda,-1), a_\lambda, 0, 0, 0, 0)_{\mathbb{D}^*} \\
t_\lambda \geq p & \mathbf{k}^*_\lambda = (\pi_\lambda(t_\lambda,-1), a_\lambda, 0, 0, 0, r_\lambda)_{\mathbb{D}^*}
\end{array}$$

Game $\mathbf{G}_{0.p.1}$: In this game, we duplicate every $u_{m,t}$ into the 5-th column of the ciphertext. To this aim, one defines the matrices

$$D = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}_{6,7} \qquad D' = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}_{6,7} \qquad \mathbb{D}^* = D' \cdot \mathbb{V}^* \qquad \mathbb{D} = D \cdot \mathbb{V}$$

which only modifies $\mathbf{d}_7$, which is secret, and $\mathbf{d}^*_6$, which is hidden, so the change is indistinguishable for the adversary. One can compute the keys and ciphertexts as follows, for all leaves λ, and for each of each query m of the adversary:

$$\begin{array}{rl}
 & \mathbf{c}_{m,t} = (\sigma_{m,t}(1,t), \omega_m, 0, 0, 0, u_{m,t})_{\mathbb{V}} \\
 & \phantom{\mathbf{c}_{m,t}} = (\sigma_{m,t}(1,t), \omega_m, 0, 0, u_{m,t}, u_{m,t})_{\mathbb{D}} \\
t_\lambda < p & \mathbf{k}^*_\lambda = (\pi_\lambda(t_\lambda,-1), a_\lambda, 0, 0, 0, 0)_{\mathbb{V}^*} \\
 & \phantom{\mathbf{k}^*_\lambda} = (\pi_\lambda(t_\lambda,-1), a_\lambda, 0, 0, 0, 0)_{\mathbb{D}^*} \\
t_\lambda \geq p & \mathbf{k}^*_\lambda = (\pi_\lambda(t_\lambda,-1), a_\lambda, 0, 0, 0, r_\lambda)_{\mathbb{V}^*} \\
 & \phantom{\mathbf{k}^*_\lambda} = (\pi_\lambda(t_\lambda,-1), a_\lambda, 0, 0, 0, r_\lambda)_{\mathbb{D}^*}
\end{array}$$

Hence, the perfect indistinguishability between the two games: $\mathsf{Adv}_{0.p.1} = \mathsf{Adv}_{0.p.0}$.

Game $\mathbf{G}_{0.p.2}$: The previous game and this game are indistinguishable under the DSDH assumption in $\mathbb{G}_2$: one essentially uses theorem 21. Given a tuple $(a \cdot G_2, b \cdot G_2, c \cdot G_2)$ in $\mathbb{G}_2$, where $c = ab + \mu \bmod q$ with either $\mu = 0$ or $\mu = 1$, the 7-th component of the leaf λ of the challenge key, with $t_\lambda = p$. When we start from random dual orthogonal bases $(\mathbb{U}, \mathbb{U}^*)$ and $(\mathbb{V}, \mathbb{V}^*)$ of size 3 and 7 respectively, one considers the matrices:

$$D = \begin{pmatrix} 1 & 0 & 0 \\ a & 1 & 0 \\ -a & 0 & 1 \end{pmatrix}_{2,6,7} \qquad D' = \begin{pmatrix} 1 & -a & a \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}_{2,6,7} \qquad \mathbb{D} = D \cdot \mathbb{V} \qquad \mathbb{D}^* = D' \cdot \mathbb{V}^*$$

We can calculate all vectors but $\mathbf{d}_6$ and $\mathbf{d}_7$, which are not in the public key. Through $\mathbb{V}$, we calculate the challenge key for the attribute of the p-th ciphertext
We choose additional random scalars $\beta_\lambda \stackrel{\$}{\leftarrow} \mathbb{Z}_q$, to virtually set $b_\lambda = r_\lambda \cdot b + \beta_\lambda$ and $c_\lambda = r_\lambda \cdot c + \beta_\lambda \cdot a$, then $c_\lambda - ab_\lambda = \mu \cdot r_\lambda$, which is either 0 or r_λ.

$$\begin{aligned} t_\lambda = p \quad \mathbf{k}^*_\lambda &= (0,0,a_\lambda,0,0,0,r_\lambda)_{\mathbb{D}^*} + (b_\lambda(t_\lambda,-1),0,0,0,c_\lambda,-c_\lambda)_{\mathbb{V}^*} \\ &= (0,0,a_\lambda,0,0,0,r_\lambda)_{\mathbb{D}^*} + (b_\lambda(t_\lambda,-1),0,0,0,c_\lambda - ab_\lambda, -c_\lambda + ab_\lambda)_{\mathbb{D}^*} \\ &= (b_\lambda(t_\lambda,-1),a_\lambda,0,0,\mu \cdot r_\lambda, r_\lambda - \mu \cdot r_\lambda)_{\mathbb{D}^*} \end{aligned}$$

If $\mu = 0$, we are in the previous game. If $\mu = 1$, then we are in the current game. Then, every other key is computed directly in $\mathbb{D}^*$:

$$\begin{aligned} t_\lambda < p \quad \mathbf{k}^*_\lambda &= (\pi_\lambda(t_\lambda,-1),a_\lambda,0,0,0,0)_{\mathbb{D}^*} \\ t_\lambda > p \quad \mathbf{k}^*_\lambda &= (\pi_\lambda(t_\lambda,-1),a_\lambda,0,0,0,r_\lambda)_{\mathbb{D}^*} \end{aligned}$$

as well as the answers to $\mathsf{OKeyGen}$-queries.
The ciphertexts are calculated through $\mathbb{V}$ but are unchanged by the change of basis because the 6-th and 7-th components are exactly the same for every ciphertext query m, and thus cancel themselves in the 2nd component. We thus have $\mathsf{Adv}_{0.p.1} - \mathsf{Adv}_{0.p.2} \leq 2 \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t)$.

Game $\mathbf{G}_{0.p.3}$: We keep the $u_{m,p}$ value (at the 6-th hidden position) in the ciphertexts, and replace it in all other ciphertexts by 0. To show this is possible without impacting the other vectors, we use the $\mathsf{Index\text{-}Ind}$ property from Theorem 23, but in another level of sequence of hybrid games, for $\gamma \in \{1, \dots, P\} \backslash \{p\}$:

Game $\mathbf{G}_{0.p.2.\gamma}$: We consider the following hybrid game, where the first satisfied condition on the indices is applied:

$$\begin{aligned} \mathbf{c}_{m,p} &= (\sigma_{m,p}(1,p),\omega_m,0,0,u_{m,p},u_{m,p})_{\mathbb{D}} & \\ \mathbf{c}_{m,t} &= (\sigma_{m,t}(1,t),\omega_m,0,0,0,u_{m,t})_{\mathbb{D}} & p \neq t < \gamma \\ \mathbf{c}_{m,t} &= (\sigma_{m,t}(1,t),\omega_m,0,0,u_{m,t},u_{m,t})_{\mathbb{D}} & p \neq t \geq \gamma \end{aligned}$$

Keys are unchanged throughout the hybrid game

$$\begin{aligned} \mathbf{k}^*_\lambda &= (\pi_\lambda(t_\lambda,-1),a_\lambda,0,0,0,0)_{\mathbb{D}^*} & t_\lambda < p \\ \mathbf{k}^*_\lambda &= (\pi_\lambda(t_\lambda,-1),a_\lambda,0,0,r_\lambda,0)_{\mathbb{D}^*} & t_\lambda = p \\ \mathbf{k}^*_\lambda &= (\pi_\lambda(t_\lambda,-1),a_\lambda,0,0,0,r_\lambda)_{\mathbb{D}^*} & t_\lambda > p \end{aligned}$$

When $\gamma = 1$, this is the previous game: $\mathbf{G}_{0.p.2.1} = \mathbf{G}_{0.p.2}$, whereas with $\gamma = P + 1$, this is the current game: $\mathbf{G}_{0.p.2.P+1} = \mathbf{G}_{0.p.3}$.
We will gradually replace the $u_{m,t}$ values, at the 6-th hidden position, by 0 (when $t \neq p$): in this game, we deal with the case $t = \gamma$, for the m-th ciphertext query. We consider a

triple $(a \cdot G_1, b \cdot G_1, c \cdot G_1)$, where $c = ab + \zeta \bmod q$ with either $\zeta = 0$ or $\zeta = 1$, which are indistinguishable under the DSDH assumption. We define the matrices

$$D = \frac{1}{p-\gamma} \times \begin{pmatrix} p & -\gamma & ap \\ -1 & 1 & -a \\ 0 & 0 & p-\gamma \end{pmatrix}_{1,2,6} \qquad D' = \begin{pmatrix} 1 & 1 & 0 \\ \gamma & p & 0 \\ -a & 0 & 1 \end{pmatrix}_{1,2,6}$$

and then $\mathbb{D} = D \cdot \mathbb{V}$, $\mathbb{D}^* = D' \cdot \mathbb{V}^*$: we cannot compute $\mathbf{d}_6^*$, but this component is always 0, except for $\mathbf{k}_\lambda^*$ with $t_\lambda = p$ which will be defined in the original basis $\mathbb{V}^*$.

$$\begin{aligned} \mathbf{k}_\lambda^* &= (0,0,a_\lambda,0,0,0,0)_{\mathbb{D}^*} + ((p-\gamma)\cdot(\pi,0),0,0,0,r_\lambda,0)_{\mathbb{V}^*} \\ &= (0,0,a_\lambda,0,0,0,0)_{\mathbb{D}^*} + (p\pi + apr_\lambda, -\pi - ar_\lambda, 0,0,0,r_\lambda,0)_{\mathbb{D}^*} \\ &= ((\pi + ar_\lambda)\cdot(p,-1), a_\lambda, 0,0,r_\lambda,0)_{\mathbb{D}^*} \end{aligned}$$

The other keys will be directly generated in $\mathbb{D}^*$, since they have no 6-th component. Similarly, we will define all the ciphertexts $\mathbf{c}_{m,t}$ for $t \neq \gamma$ in $\mathbb{D}^*$, but for $t = \gamma$, we choose additional random scalars $\beta_{m,t} \xleftarrow{\$} \mathbb{Z}_q$, to virtually set $b_{m,t} = u_{m,t} \cdot b + \beta_{m,t}$ and $c_{m,t} = u_{m,t} \cdot c + \beta_{m,t} \cdot a$, then $c_{m,t} - ab_{m,t} = \zeta \cdot u_{m,t}$, which is either 0 or $u_{m,t}$. One can set

$$\begin{aligned} \mathbf{c}_{m,t} &= (0,0,\omega_m,0,0,0,u_{m,t})_{\mathbb{D}} + (b_{m,t},0,0,0,0,c_{m,t},0)_{\mathbb{V}} \\ &= (0,0,\omega_m,0,0,0,u_{m,t})_{\mathbb{D}} + (b_{m,t}(1,\gamma),0,0,0,c_{m,t} - ab_{m,t},0)_{\mathbb{D}} \\ &= (b_{m,t}(1,\gamma),\omega_m,0,0,\zeta\cdot u_{m,t},u_{m,t})_{\mathbb{D}} \end{aligned}$$

when $t = \gamma \neq p$. This is either the current game $\mathbf{G}_{0.p.2.\gamma}$, if $\zeta = 1$, or the next game $\mathbf{G}_{0.p.2.\gamma+1}$, if $\zeta = 0$.
We remind that $u_{m,p} = 0$ because $r_\lambda \neq 0$. If $r_\lambda = 0$, then we would have skipped directly to the hybrid $p+1$ game.
With all this sequence, we have $\mathsf{Adv}_{0.p.2} - \mathsf{Adv}_{0.p.3} \leq 2P \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t)$.

Game $\mathbf{G}_{0.p.4}$: In this final game for p, we can finally cancel out r_λ in each key with $t_\lambda = p$ because it corresponds to a coordinate where all other values (in keys and ciphertexts) are 0. We consider a triple $(a \cdot G_2, b \cdot G_2, c \cdot G_2)$, where $c = ab + \alpha \mod q$, with either $\alpha = 0$ or $\alpha = r_\lambda$. One defines the matrices

$$D = \begin{pmatrix} 1 & 0 \\ a & 1 \end{pmatrix}_{1,6} \qquad D' = \begin{pmatrix} 1 & -a \\ 0 & 1 \end{pmatrix}_{1,6} \qquad \mathbb{D} = D \cdot \mathbb{V} \qquad \mathbb{D}^* = D' \cdot \mathbb{V}^*$$

Note that we can compute all the basis vectors excepted $\mathbf{d}_6$, but all the ciphertexts have a 0 components in 6-th position. So one can set all the values honestly in $\mathbb{D}$ and $\mathbb{D}^*$, except for

$$\begin{aligned} \mathbf{k}_\lambda &= (0,0,a_\lambda,0,0,0,0)_{\mathbb{D}} + (b(p,-1),0,0,0,c,0)_{\mathbb{V}} \\ &= (0,0,a_\lambda,0,0,0,0)_{\mathbb{D}} + (b(p,-1),0,0,0,c-ab,0)_{\mathbb{D}} \\ &= (b(1,p),a_\lambda,0,0,\alpha,0)_{\mathbb{D}} \end{aligned}$$

When $\alpha = 0$, this is exactly the current game, with $\pi_\lambda = b$, whereas $\alpha = r_\lambda$, this is the previous game. Then, $\mathsf{Adv}_{0.p.3} - \mathsf{Adv}_{0.p.4} \leq 2 \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t)$.

In total, this sequence of games, for a given p, satisfies Then,

$$\begin{aligned} \mathsf{Adv}_{\mathbf{G}_{0.p.4}} - \mathsf{Adv}_{\mathbf{G}_{0.p.0}} &\leq 2 \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t) + 2P \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t) + 2 \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t) \\ &\leq 4 \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t) + 2P \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t) \leq (2P+4) \cdot \mathsf{Adv}^{\mathsf{sxdh}}(t) \end{aligned}$$

In the last game, the adversary has zero advantage. Indeed, whether $b = 0$ or $b = 1$, the distributions of dk_0 and dk_1 are perfectly identical, with all-passive leaves.

C.4 Proof of Theorem 13 – dAtt-IND-Security

Proof. We start with the distinct variant, where all the invalid attributes in the challenge ciphertext do not correspond to any active leaf in the obtained keys. Our proof will proceed by games.

Game $\mathbf{G}_0$: This is the real security game, where the simulator honestly emulates the challenger, with $\mathsf{PK} = \{(\mathbf{b}_1, \mathbf{b}_3, \mathbf{b}_1^*), (\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_7, \mathbf{d}_1^*, \mathbf{d}_2^*, \mathbf{d}_3^*)\}$ and $\mathsf{MK} = \{\mathbf{b}_3^*, \mathbf{d}_7^*\}$, from random dual orthogonal bases. The public parameters PK are provided to the adversary. Since $\mathbf{d}_7$ is public (empty SK), there is no need to provide access to an encryption oracle.

$\mathsf{OKeyGen}(\tilde{\mathcal{T}}_\ell)$ (or $\mathsf{ODelegate}$-queries): The adversary is allowed to issue KeyGen-queries on an access-tree $\tilde{\mathcal{T}}_\ell = (\mathcal{T}_\ell, \mathcal{L}_{\ell,a}, \mathcal{L}_{\ell,p})$ (for the ℓ-th query), for which the simulator chooses a random scalar $a_{\ell,0} \xleftarrow{\$} \mathbb{Z}_q$ and a random $a_{\ell,0}$-labeling $(a_{\ell,\lambda})_\lambda$ of the access-tree $\mathcal{T}_\ell$, and builds the key:

$$\mathbf{k}_{\ell,0}^* = (a_{\ell,0}, 0, 1)_{\mathbb{B}^*} \qquad \mathbf{k}_{\ell,\lambda}^* = (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, 0, 0, r_{\ell,\lambda})_{\mathbb{D}^*}$$

for all the leaves λ, where $t_{\ell,\lambda} = A(\lambda)$, $\pi_{\ell,\lambda} \xleftarrow{\$} \mathbb{Z}_q$ and $r_{\ell,\lambda} \xleftarrow{\$} \mathbb{Z}_q^*$ if λ is an active leaf, or $r_{\ell,\lambda} \leftarrow 0$ otherwise. The decryption key is $\mathsf{dk}_\ell = (\mathbf{k}_{\ell,0}^*, (\mathbf{k}_{\ell,\lambda}^*)_\lambda)$;

$\mathsf{RoAVEncaps}(\Gamma_v, \Gamma_i)$: The challenge ciphertext is built on a set of attributes $\Gamma_v \cup \Gamma_i$, with random scalars $\omega, \xi \xleftarrow{\$} \mathbb{Z}_q$ to set $K = g_t^\xi$. Then, the simulator generates the ciphertext $C_0 = (\mathbf{c}_0, (\mathbf{c}_t)_t)$, for all the attributes $t \in \Gamma_v \cup \Gamma_i$, with $\sigma_t \xleftarrow{\$} \mathbb{Z}_q$, and where $u_t \xleftarrow{\$} \mathbb{Z}_q^*$ if $t \in \Gamma_i$, or $u_t = 0$ if $t \in \Gamma_v$:

$$\mathbf{c}_0 = (\omega, 0, \xi)_{\mathbb{B}} \qquad \mathbf{c}_t = (\sigma_t(1, t), \omega, 0, 0, 0, u_t)_{\mathbb{D}}$$

On the other hand, it computes $C_1 = (\mathbf{c}_0, (\mathbf{c}_t)_t)$ for all $t \in \Gamma_v \cup \Gamma_i$ as:

$$\mathbf{c}_0 = (\omega, 0, \xi)_{\mathbb{B}} \qquad \mathbf{c}_t = (\sigma_t(1, t), \omega, 0, 0, 0, 0)_{\mathbb{D}}$$

According to the real or all-valid game (bit $b \xleftarrow{\$} \{0, 1\}$), one outputs (K, C_b).

From the adversary's guess b' for b, if for some $\tilde{\mathcal{T}}_\ell = (\mathcal{T}_\ell, \mathcal{L}_{\ell,a}, \mathcal{L}_{\ell,p})$, there is some active leaf $\lambda \in \mathcal{L}_{\ell,a}$ such that $t_\lambda = A(\lambda) \in \Gamma_i$, then $\beta \xleftarrow{\$} \{0, 1\}$, otherwise $\beta = b'$. We denote $\mathsf{Adv}_0 = \Pr[\beta = 1 | b = 1] - \Pr[\beta = 1 | b = 0]$.

We stress that in this distinct attribute-indistinguishability security game, the invalid attributes in the challenge ciphertext ($t \in \Gamma_i$ with possibly $u_t \neq 0$) correspond to passive leaves only ($\lambda \in \mathcal{L}_{\ell,p}$ with $r_{\ell,\lambda} = 0$, for all queries). But we do not exclude accepting access-trees.

Game $\mathbf{G}_1$: The second and final game simply corresponds to the situation where $u_t = 0$ in C_0, clearly leading to $\mathsf{Adv}_1 = 0$.

Using the indexing technique, we can show this game is indistinguishable the previous game. But we need to describe a sub-sequence of games (see Figure 16) for proving the gap from the above $\mathbf{G}_0$ to $\mathbf{G}_1$, with the sequence $\mathbf{G}_{0.p.*}$, that will modify the p-th ciphertext in the challenge ciphertext, for $p \in \{1, \ldots, P+1\}$, where $\mathbf{G}_0 = \mathbf{G}_{0.1.0}$, and $\mathbf{G}_1 = \mathbf{G}_{0.P+1.0}$. In these games, we describe how we generate the keys and the real encapsulation C_0. C_1 will be easily simulated in an honest way.

Game $\mathbf{G}_{0.p.0}$: One thus chooses random scalars and defines the hybrid game for some p, where the first components of the ciphertext are all-valid, and the last ones are real:

$$\begin{array}{lll}
\mathbf{k}_{\ell,0}^* = (a_{\ell,0}, 0, 1)_{\mathbb{B}^*} & \mathbf{k}_{\ell,\lambda}^* = (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, 0, 0, r_{\ell,\lambda})_{\mathbb{D}^*} & \\
\mathbf{c}_0 = (\omega, 0, \xi)_{\mathbb{B}} & \mathbf{c}_t = (\sigma_t(1, t), \omega, 0, 0, 0, 0)_{\mathbb{D}} & \text{if } t < p \\
\mathbf{c}_0 = (\omega, 0, \xi)_{\mathbb{B}} & \mathbf{c}_t = (\sigma_t(1, t), \omega, 0, 0, 0, u_t)_{\mathbb{D}} & \text{if } t \geq p
\end{array}$$

$$\mathbf{c}_0 = \begin{pmatrix} \omega & 0 & \xi \end{pmatrix} \qquad \mathbf{k}^*_{\ell,0} = \begin{pmatrix} a_{\ell,0} & 0 & 1 \end{pmatrix}$$

$\mathbf{G}_{0.p.0}$ Hybrid game for $\mathbf{G}_0$ and $\mathbf{G}_1$, with $1 \leq p \leq P+1$

$$\begin{array}{lrl} & \mathbf{k}^*_{\ell,\lambda} = & (\; \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) \;\; a_{\ell,\lambda} \mid 0 \;\; 0 \;\; 0 \;\; r_{\ell,\lambda} \;) \\ t < p & \mathbf{c}_t = & (\; \sigma_t(1,t) \;\; \omega \mid 0 \;\; 0 \;\; 0 \;\; 0 \;) \\ t \geq p & \mathbf{c}_t = & (\; \sigma_t(1,t) \;\; \omega \mid 0 \;\; 0 \;\; 0 \;\; u_t \;) \end{array}$$

$\mathbf{G}_{0.p.1}$ Formal basis change, on $(\mathbb{D},\mathbb{D}^*)_{6,7}$, to duplicate $r_{\ell,\lambda}$ in the 6-th column

$$\begin{array}{lrl} & \mathbf{k}^*_{\ell,\lambda} = & (\; \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) \;\; a_{\ell,\lambda} \mid 0 \;\; 0 \;\; r_{\ell,\lambda} \;\; r_{\ell,\lambda} \;) \\ t < p & \mathbf{c}_t = & (\; \sigma_t(1,t) \;\; \omega \mid 0 \;\; 0 \;\; 0 \;\; 0 \;) \\ t \geq p & \mathbf{c}_t = & (\; \sigma_t(1,t) \;\; \omega \mid 0 \;\; 0 \;\; 0 \;\; u_t \;) \end{array}$$

$\mathbf{G}_{0.p.2}$ Swap-Ind Property, on $(\mathbb{D},\mathbb{D}^*)_{1,6,7}$, to swap u_p alone in the 6-th column

$$\begin{array}{lrl} & \mathbf{k}^*_{\ell,\lambda} = & (\; \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) \;\; a_{\ell,\lambda} \mid 0 \;\; 0 \;\; r_{\ell,\lambda} \;\; r_{\ell,\lambda} \;) \\ t < p & \mathbf{c}_t = & (\; \sigma_t(1,t) \;\; \omega \mid 0 \;\; 0 \;\; 0 \;\; 0 \;) \\ & \mathbf{c}_p = & (\; \sigma_p(1,p) \;\; \omega \mid 0 \;\; 0 \;\; u_p \;\; 0 \;) \\ t > p & \mathbf{c}_t = & (\; \sigma_t(1,t) \;\; \omega \mid 0 \;\; 0 \;\; 0 \;\; u_t \;) \end{array}$$

$\mathbf{G}_{0.p.3}$ Index-Ind Property, on $(\mathbb{D},\mathbb{D}^*)_{1,2,6}$, between $r_{\ell,\lambda}$ and 0, for $t_{\ell,\lambda} \neq p$

$$\begin{array}{lrl} t_{\ell,\lambda} = p & \mathbf{k}^*_{\ell,\lambda} = & (\; \pi_{\ell,\lambda}(p,-1) \;\; a_{\ell,\lambda} \mid 0 \;\; 0 \;\; 0 \;\; 0 \;) \\ t_{\ell,\lambda} \neq p & \mathbf{k}^*_{\ell,\lambda} = & (\; \pi_{\ell,\lambda}(p,-1) \;\; a_{\ell,\lambda} \mid 0 \;\; 0 \;\; 0 \;\; r_{\ell,\lambda} \;) \\ t < p & \mathbf{c}_t = & (\; \sigma_t(1,t) \;\; \omega \mid 0 \;\; 0 \;\; 0 \;\; 0 \;) \\ & \mathbf{c}_p = & (\; \sigma_p(1,p) \;\; \omega \mid 0 \;\; 0 \;\; u_p \;\; 0 \;) \\ t > p & \mathbf{c}_t = & (\; \sigma_t(1,t) \;\; \omega \mid 0 \;\; 0 \;\; 0 \;\; u_t \;) \end{array}$$

$\mathbf{G}_{0.p.4}$ SubSpace-Ind Property, on $(\mathbb{D},\mathbb{D}^*)_{1,6}$, between u_p and 0

$$\begin{array}{lrl} & \mathbf{k}^*_{\ell,\lambda} = & (\; \pi_{\ell,\lambda}(p,-1) \;\; a_{\ell,\lambda} \mid 0 \;\; 0 \;\; 0 \;\; r_{\ell,\lambda} \;) \\ t < p & \mathbf{c}_t = & (\; \sigma_t(1,t) \;\; \omega \mid 0 \;\; 0 \;\; 0 \;\; 0 \;) \\ & \mathbf{c}_p = & (\; \sigma_p(1,p) \;\; \omega \mid 0 \;\; 0 \;\; 0 \;\; 0 \;) \\ t > p & \mathbf{c}_t = & (\; \sigma_t(1,t) \;\; \omega \mid 0 \;\; 0 \;\; 0 \;\; u_t \;) \end{array}$$

Fig. 16: Sub-sequence of games for Distinct Attribute-Indistinguishability

Of course, the values $r_{\ell,\lambda}$ and u_t are random in $\mathbb{Z}_q^*$ or 0 according to $\mathcal{L}_{\ell,a}/\mathcal{L}_{\ell,p}$ and Γ_i/Γ_v. In particular, if $u_p = 0$, we can directly go to $\mathbf{G}_{0.p.4}$, as there is no change from this game. The following sequence only makes sense when $u_p \neq 0$, but then necessarily $r_{\ell,\lambda} = 0$ for all the pairs (ℓ,λ) such that $t_{\ell,\lambda} = p$. We thus assume this restriction in this sequence: $u_p \neq 0$ and $r_{\ell,\lambda} = 0$ for all (ℓ,λ) such that $t_{\ell,\lambda} = p$.

Game $\mathbf{G}_{0.p.1}$: One defines the matrices

$$D = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}_{6,7} \qquad D' = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}_{6,7} \qquad \mathbb{D} = D \cdot \mathbb{V} \qquad \mathbb{D}^* = D' \cdot \mathbb{V}^*$$

which modifies the hidden and secret vectors $\mathbf{d}_6$ and $\mathbf{d}^*_7$, and so are not in the view of the adversary:

$$\begin{aligned} \mathbf{k}^*_{\ell,\lambda} &= (\pi_{\ell,\lambda}(t_{\ell,\lambda},-1), a_{\ell,\lambda}, 0, 0, 0, r_{\ell,\lambda})_{\mathbb{V}^*} \\ &= (\pi_{\ell,\lambda}(t_{\ell,\lambda},-1), a_{\ell,\lambda}, 0, 0, r_{\ell,\lambda}, r_{\ell,\lambda})_{\mathbb{D}^*} \\ \mathbf{c}_t &= (\sigma_t(1,t), \omega, 0, 0, 0, 0)_{\mathbb{V}} = (\sigma_t(1,t), \omega, 0, 0, 0, 0)_{\mathbb{D}} && \text{if } t < p \\ \mathbf{c}_t &= (\sigma_t(1,t), \omega, 0, 0, 0, u_t)_{\mathbb{V}} = (\sigma_t(1,t), \omega, 0, 0, 0, u_t)_{\mathbb{D}} && \text{if } t \geq p \end{aligned}$$

We thus have $\mathsf{Adv}_{0.p.1} = \mathsf{Adv}_{0.p.0}$.

Game $\mathbf{G}_{0.p.2}$: We use the Swap-Ind-property on $(\mathbb{D},\mathbb{D}^*)_{1,6,7}$: Indeed, we can consider a triple $(a \cdot G_1, b \cdot G_1, c \cdot G_1)$, where $c = ab + \theta \bmod q$ with either $\theta = 0$ or $\theta = u_p$. We define the matrices

$$D = \begin{pmatrix} 1 & a & -a \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}_{1,6,7} \qquad D' = \begin{pmatrix} 1 & 0 & 0 \\ -a & 1 & 0 \\ a & 0 & 1 \end{pmatrix}_{1,6,7} \qquad \mathbb{D} = D \cdot \mathbb{V} \qquad \mathbb{D}^* = D' \cdot \mathbb{V}^*$$

Note that we can compute all the basis vectors excepted $\mathbf{d}_6^*, \mathbf{d}_7^*$, but we define the keys on the original basis $\mathbb{V}^*$:

$$\begin{aligned}
\mathbf{k}^*_{\ell,\lambda} &= (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, 0, r_{\ell,\lambda}, r_{\ell,\lambda})_{\mathbb{V}^*} \\
&= (\pi_{\ell,\lambda} \cdot t_{\ell,\lambda} + ar_{\ell,\lambda} - ar_{\ell,\lambda}, -\pi_{\ell,\lambda}, a_{\ell,\lambda}, 0, 0, r_{\ell,\lambda}, r_{\ell,\lambda})_{\mathbb{D}^*} \\
&= (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, 0, r_{\ell,\lambda}, r_{\ell,\lambda})_{\mathbb{D}^*} \\
\mathbf{c}_t &= (\sigma_t(1,t), \omega, 0, 0, 0, 0)_{\mathbb{D}} && \text{if } t < p \\
\mathbf{c}_p &= (\sigma(1,p), \omega, 0, 0, 0, u_p)_{\mathbb{D}} + (b(1,p), 0, 0, 0, c, -c)_{\mathbb{V}} \\
&= (\sigma(1,p), \omega, 0, 0, 0, u_p)_{\mathbb{D}} + (b(1,p), 0, 0, 0, c - ab, -c + ab)_{\mathbb{D}} \\
&= ((\sigma + b)(1,p), \omega, 0, 0, \theta, u_p - \theta)_{\mathbb{D}} \\
\mathbf{c}_t &= (\sigma_t(1,t), \omega, 0, 0, 0, u_t)_{\mathbb{D}} && \text{if } t > p
\end{aligned}$$

With $\theta = 0$, this is as in the previous game, where $\sigma_p = \sigma + b$. When $\theta = u_p$, this is the current game: $\mathsf{Adv}_{0.p.1} - \mathsf{Adv}_{0.p.2} \leq 2 \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t)$.

Game $\mathbf{G}_{0.p.3}$: We make all the $r_{\ell,\lambda}$ values (at the 6-th hidden position) in the keys to be 0, excepted for $t_{\ell,\lambda} = p$. The case $t_{\ell,\lambda} = p$ is already $r_{\ell,\lambda} = 0$, by assumption in this sequence, as $u_p \neq 0$. For that, we iteratively replace all the values by zero, using $\mathsf{Index\text{-}Ind}$-property, in another level of sequence of hybrid games, for $\gamma \in \{1, \dots, P\} \backslash \{p\}$:

Game $\mathbf{G}_{0.p.2.\gamma}$: We consider

$$\begin{aligned}
\mathbf{k}^*_{\ell,\lambda} &= (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, 0, 0, 0)_{\mathbb{D}^*} && \text{if } t_{\ell,\lambda} = p \\
\mathbf{k}^*_{\ell,\lambda} &= (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, 0, 0, r_{\ell,\lambda})_{\mathbb{D}^*} && \text{if } p \neq t_{\ell,\lambda} < \gamma \\
\mathbf{k}^*_{\ell,\lambda} &= (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, 0, r_{\ell,\lambda}, r_{\ell,\lambda})_{\mathbb{D}^*} && \text{if } p \neq t_{\ell,\lambda} \geq \gamma \\
\mathbf{c}_t &= (\sigma_t(1,t), \omega, 0, 0, 0, 0)_{\mathbb{D}} && \text{if } t < p \\
\mathbf{c}_p &= (\sigma_p(1,p), \omega, 0, 0, u_p, 0)_{\mathbb{D}} \\
\mathbf{c}_t &= (\sigma_t(1,t), \omega, 0, 0, 0, u_t)_{\mathbb{D}} && \text{if } t > p
\end{aligned}$$

When $\gamma = 1$, this is the previous game: $\mathbf{G}_{0.p.2.1} = \mathbf{G}_{0.p.2}$, whereas with $\gamma = P + 1$, this is the current game: $\mathbf{G}_{0.p.2.P+1} = \mathbf{G}_{0.p.3}$.
We can consider a triple $(a \cdot G_2, b \cdot G_2, c \cdot G_2)$, where $c = ab + \zeta \bmod q$ with either $\zeta = 0$ or $\zeta = 1$. We define the matrices

$$D' = \frac{1}{\gamma - p} \times \begin{pmatrix} 1 & -1 & a \\ \gamma & -p & ap \\ 0 & 0 & \gamma - p \end{pmatrix}_{1,2,6} \qquad D = \begin{pmatrix} -p & -\gamma & 0 \\ 1 & 1 & 0 \\ 0 & a & 1 \end{pmatrix}_{1,2,6}$$

and then $\mathbb{D} = D \cdot \mathbb{V}$, $\mathbb{D}^* = D' \cdot \mathbb{V}^*$: we cannot compute $\mathbf{d}_6$, but this component is always 0, excepted for $\mathbf{c}_p$ we will define in the original basis $\mathbb{V}$. One chooses additional random scalars $\beta_{\ell,\lambda} \xleftarrow{\$} \mathbb{Z}_q$, for all (ℓ, λ) such that $t_{\ell,\lambda} = \gamma$, to virtually set $b_{\ell,\lambda} = -r_{\ell,\lambda} \cdot b + \beta_{\ell,\lambda}$

and $c_{\ell,\lambda} = -r_{\ell,\lambda} \cdot c + \beta_{\ell,\lambda} \cdot a$, $c_{\ell,\lambda} - ab_{\ell,\lambda} = -r_{\ell,\lambda} \cdot \zeta$. One can set

$$\begin{aligned}
\mathbf{k}^*_{\ell,\lambda} &= (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, 0, 0, 0)_{\mathbb{D}^*} && \text{if } t_{\ell,\lambda} = p \\
\mathbf{k}^*_{\ell,\lambda} &= (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, 0, 0, r_{\ell,\lambda})_{\mathbb{D}^*} && \text{if } p \neq t_{\ell,\lambda} < \gamma \\
\mathbf{k}^*_{\ell,\lambda} &= (b_{\ell,\lambda}(0, -1), a_{\ell,\lambda}, 0, 0, r_{\ell,\lambda} + c_{\ell,\lambda}, r_{\ell,\lambda})_{\mathbb{V}^*} && \text{if } t_{\ell,\lambda} = \gamma \\
&= (b_{\ell,\lambda}(\gamma, -1), a_{\ell,\lambda}, 0, 0, r_{\ell,\lambda} + c_{\ell,\lambda} - ab_{\ell,\lambda}, r_{\ell,\lambda})_{\mathbb{V}^*} \\
&= (b_{\ell,\lambda}(\gamma, -1), a_{\ell,\lambda}, 0, 0, r_{\ell,\lambda} \cdot (1 - \zeta), r_{\ell,\lambda})_{\mathbb{V}^*} \\
\mathbf{k}^*_{\ell,\lambda} &= (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, 0, r_{\ell,\lambda}, r_{\ell,\lambda})_{\mathbb{D}^*} && \text{if } p \neq t_{\ell,\lambda} > \gamma \\
\mathbf{c}_t &= (\sigma_t(1, t), \omega, 0, 0, 0, 0)_{\mathbb{D}} && \text{if } t < p \\
\mathbf{c}_p &= ((p - \gamma)\sigma(0, 1), \omega, 0, 0, u_p, 0)_{\mathbb{V}} \\
&= (\sigma(1, p) + au_p(1, p)/(\gamma - p), \omega, 0, 0, u_p, 0)_{\mathbb{D}} \\
&= ((\sigma + au_p/(\gamma - p))(1, p), \omega, 0, 0, u_p, 0)_{\mathbb{D}} \\
&= (\sigma_p(1, p), \omega, 0, 0, u_p, 0)_{\mathbb{D}} \\
\mathbf{c}_t &= (\sigma_t(1, t), \omega, 0, 0, 0, u_t)_{\mathbb{D}} && \text{if } t > p
\end{aligned}$$

When $\zeta = 0$, this $\mathbf{G}_{0.p.2.\gamma}$, when $\zeta = 1$, this is $\mathbf{G}_{0.p.2.\gamma+1}$.
As a consequence, $\mathsf{Adv}_{0.p.2} - \mathsf{Adv}_{0.p.3} \leq 2P \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t)$.

Game $\mathbf{G}_{0.p.4}$: One can easily conclude by removing u_p in the ciphertext $\mathbf{c}_p$, as it corresponds to a coordinate where all the other values (in the keys and the ciphertext) are 0. To this aim, we can consider a triple $(a \cdot G_1, b \cdot G_1, c \cdot G_1)$, where $c = ab + \alpha \bmod q$ with either $\alpha = 0$ or $\alpha = u_p$. One defines the matrices

$$D = \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}_{1,6} \qquad D' = \begin{pmatrix} 1 & 0 \\ -a & 1 \end{pmatrix}_{1,6} \qquad \mathbb{D} = D \cdot \mathbb{V} \qquad \mathbb{D}^* = D' \cdot \mathbb{V}^*$$

Note that we can compute all the basis vectors excepted $\mathbf{d}^*_6$, which has only 0 components in the keys. So one can set all the values honestly in $\mathbb{D}$ and $\mathbb{D}^*$, excepted

$$\begin{aligned}
\mathbf{c}_p &= (b(1, p), \omega, 0, 0, c, 0)_{\mathbb{V}} = (b(1, p), \omega, 0, 0, c - ab, 0)_{\mathbb{D}} \\
&= (b(1, p), \omega, 0, 0, \alpha, 0)_{\mathbb{D}}
\end{aligned}$$

When $\alpha = 0$, this is exactly the current game, with $\sigma_p = b$, whereas for $\alpha = u_p$, this is the previous game. Then, $\mathsf{Adv}_{0.p.3} - \mathsf{Adv}_{0.p.4} \leq 2 \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t)$.

In total, this sequence of games, for a given p, satisfies Then,

$$\begin{aligned}
\mathsf{Adv}_{\mathbf{G}_{0.p.4}} - \mathsf{Adv}_{\mathbf{G}_{0.p.0}} &\leq 2 \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t) + 2P \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t) + 2 \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t) \\
&\leq 4 \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t) + 2P \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t) \leq (4 + 2P) \cdot \mathsf{Adv}^{\mathsf{sxdh}}(t)
\end{aligned}$$

C.5 Proof of Theorem 14 – Att-IND-Security

Proof. We now prove the attribute-indistinguishability, where there are no restrictions between active leaves in the keys and invalid attributes in the challenge ciphertext, but just that the access-trees of the obtained keys reject the attribute-set of the challenge ciphertext, even in the all-valid case. Our proof will proceed by games. Not that we also assume active keys correspond to independent leaves with respect to the set of attributes $\Gamma = \Gamma_v \cup \Gamma_i$ in the challenge ciphertext.

Game $\mathbf{G}_0$: This is the real security game, where the simulator honestly emulates the challenger, with $\mathsf{PK} = \{(\mathbf{b}_1, \mathbf{b}_3, \mathbf{b}^*_1), (\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_7, \mathbf{d}^*_1, \mathbf{d}^*_2, \mathbf{d}^*_3)\}$ and $\mathsf{MK} = \{\mathbf{b}^*_3, \mathbf{d}^*_7\}$, from random dual orthogonal bases. The public parameters PK are provided to the adversary. Since $\mathbf{d}_7$ is public (empty SK), there is no need to provide access to an encryption oracle.

$\mathsf{OKeyGen}(\tilde{\mathcal{T}}_\ell)$ (or $\mathsf{ODelegate}$-queries): The adversary is allowed to issue KeyGen-queries on an access-tree $\tilde{\mathcal{T}}_\ell = (\mathcal{T}_\ell, \mathcal{L}_{\ell,a}, \mathcal{L}_{\ell,p})$ (for the ℓ-th query), for which the simulator chooses a random scalar $a_{\ell,0} \xleftarrow{\$} \mathbb{Z}_q$ and a random $a_{\ell,0}$-labeling $(a_{\ell,\lambda})_\lambda$ of the access-tree $\mathcal{T}_\ell$, and builds the key:

$$\mathbf{k}^*_{\ell,0} = (a_{\ell,0}, 0, 1)_{\mathbb{B}^*} \qquad \mathbf{k}^*_{\ell,\lambda} = (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, 0, 0, r_{\ell,\lambda})_{\mathbb{D}^*}$$

for all the leaves λ, where $t_{\ell,\lambda} = A(\lambda)$, $\pi_{\ell,\lambda} \xleftarrow{\$} \mathbb{Z}_q$ and $r_{\ell,\lambda} \xleftarrow{\$} \mathbb{Z}_q^*$ if λ is an active leaf, or $r_{\ell,\lambda} \leftarrow 0$ otherwise. The decryption key is $\mathsf{dk}_\ell = (\mathbf{k}^*_{\ell,0}, (\mathbf{k}^*_{\ell,\lambda})_\lambda)$;

$\mathsf{RoAVEncaps}(\Gamma_v, \Gamma_i)$: The challenge ciphertext is built on a set of attributes $\Gamma_v \cup \Gamma_i$, with random scalars $\omega, \xi \xleftarrow{\$} \mathbb{Z}_q$ to set $K = g_t^\xi$. Then, the simulator generates the ciphertext $C_1 = (\mathbf{c}_0, (\mathbf{c}_t)_t)$, for all the attributes $t \in \Gamma_v \cup \Gamma_i$, with $\sigma_t \xleftarrow{\$} \mathbb{Z}_q$:

$$\mathbf{c}_0 = (\omega, 0, \xi)_{\mathbb{B}} \qquad \mathbf{c}_t = (\sigma_t(1, t), \omega, 0, 0, 0, 0)_{\mathbb{D}}$$

On the other hand, it computes $C_0 = (\mathbf{c}_0, (\mathbf{c}_t + (0, 0, 0, 0, 0, 0, u_t)_{\mathbb{D}})_t)$, where $u_t \xleftarrow{\$} \mathbb{Z}_q^*$ if $t \in \Gamma_i$, or $u_t = 0$ if $t \in \Gamma_v$. According to the real or all-valid game (bit $b \xleftarrow{\$} \{0,1\}$), one outputs (K, C_b).

From the adversary's guess b' for b, if for some $\tilde{\mathcal{T}}_\ell = (\mathcal{T}_\ell, \mathcal{L}_{\ell,a}, \mathcal{L}_{\ell,p})$, for which tree a key has been obtained, $\tilde{\mathcal{T}}_\ell(\Gamma_v \cup \Gamma_i, \emptyset) = 1$ then $\beta \xleftarrow{\$} \{0,1\}$, otherwise $\beta = b'$. We denote $\mathsf{Adv}_0 = \Pr[\beta = 1|b = 1] - \Pr[\beta = 1|b = 0]$.

We now proceed with exactly the same sequence as in the IND-security proof of the KP-ABE in the appendix B.3, except the RoREncaps-challenge is instead a RoAVEncaps-challenge, where we require $\tilde{\mathcal{T}}_\ell(\Gamma_v \cup \Gamma_i, 0) = 0$ for all the obtained keys. For the same reason, the OEncaps-queries on pairs $(\Gamma_{m,v}, \Gamma_{m,i})$, with $\Gamma_{m,i} \neq \emptyset$ can be simulated. Indeed, as above, everything on the 7-th component can be done independently, knowing both $\mathbf{d}_7$ and $\mathbf{d}^*_7$, as these vectors will be known to the simulator, almost all the time, excepted in some specific gaps. In theses cases, we will have to make sure how to simulate the OEncaps ciphertexts.

As in that proof, the idea of the sequence is to introduce an additional labeling $(s_{\ell,0}, (s_{\ell,\lambda})_\lambda)$ in the hidden components of each key, with a random $s_{\ell,0}$, as the trees are rejecting. We are thus able to go as in $\mathbf{G}_3$, from Figure 10, where each label is masked by a random z_t for each attribute t. The following sequence is described on Figure 17.

Game $\mathbf{G}_1$: This is as $\mathbf{G}_1$, with a random τ in the challenge ciphertext.

Game $\mathbf{G}_2$: This is as $\mathbf{G}_2$, with random z_t in the challenge ciphertext.

Game $\mathbf{G}_3$: This is as $\mathbf{G}_3$, with an additional independent $s_{\ell,0}$-labeling $(s_{\ell,\lambda})$ for each access-tree $\mathcal{T}_\ell$ and a random $r_{\ell,0}$ to define

$$\mathbf{k}^*_{\ell,0} = (a_{\ell,0}, r_{\ell,0}, 1)_{\mathbb{B}^*} \qquad \mathbf{k}^*_{\ell,\lambda} = (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, 0, s_{\ell,\lambda}/z_{t_{k,\lambda}}, r_{\ell,\lambda})_{\mathbb{D}^*}$$

We stress that all these steps are not impacted by the values u_t in the 7-th component of the challenge ciphertext:

$$\mathbf{c}_0 = (\omega, 0, \xi)_{\mathbb{B}} \qquad \mathbf{c}_t = (\sigma_t(1, t), \omega, \tau, 0, \tau z_t, (1 - b) \cdot u_t)_{\mathbb{D}}$$

where b is the random bit of the challenger: when $b = 0$, the ciphertext is in the real case, whereas for $b = 1$, one gets an all-valid ciphertext.

Game $\mathbf{G}_4$: We remove all u_t from the RoAVEncaps challenge query, in the case $b = 1$:

$$\mathbf{c}_0 = (\omega, 0, \xi)_{\mathbb{B}} \qquad \mathbf{c}_t = (\sigma_t(1, t), \omega, \tau, 0, \tau z_t, 0)_{\mathbb{D}}$$
$$\mathbf{k}^*_{\ell,0} = (a_{\ell,0}, r_{\ell,0}, 1)_{\mathbb{B}^*} \qquad \mathbf{k}^*_{\ell,\lambda} = (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, 0, s'_{\ell,\lambda}/z_{t_{k,\lambda}}, r_{\ell,\lambda})_{\mathbb{D}^*}$$

$\mathbf{G}_0$ Real Att-IND-Security game

$$\begin{array}{rlcccl}
\mathbf{c}_0 = (& \omega & 0 & \xi &) & \mathbf{c}_t = (\ldots| \; 0 \quad 0 \quad 0 \quad (1-b)\cdot u_t \;) \\
\mathbf{k}^*_{\ell,0} = (& a_{\ell,0} & 0 & 1 &) & \mathbf{k}^*_{\ell,\lambda} = (\ldots| \; 0 \quad 0 \quad 0 \quad r_{\ell,\lambda} \;)
\end{array}$$

$\mathbf{G}_1$ SubSpace-Ind Property, on $(\mathbb{B},\mathbb{B}^*)_{1,2}$ and $(\mathbb{D},\mathbb{D}^*)_{3,4}$, between 0 and $\tau \xleftarrow{\$} \mathbb{Z}_q$

$$\begin{array}{rlcccl}
\mathbf{c}_0 = (& \omega & \tau & \xi &) & \mathbf{c}_t = (\ldots| \; \tau \quad 0 \quad 0 \quad (1-b)\cdot u_t \;) \\
\mathbf{k}^*_{\ell,0} = (& a_{\ell,0} & 0 & 1 &) & \mathbf{k}^*_{\ell,\lambda} = (\ldots| \; 0 \quad 0 \quad 0 \quad r_{\ell,\lambda} \;)
\end{array}$$

$\mathbf{G}_2$ SubSpace-Ind Property, on $(\mathbb{D},\mathbb{D}^*)_{(1,2),6}$, between 0 and τz_t

$$\begin{array}{rlcccl}
\mathbf{c}_0 = (& \omega & \tau & \xi &) & \mathbf{c}_t = (\ldots| \; \tau \quad 0 \quad \tau z_t \quad (1-b)\cdot u_t \;) \\
\mathbf{k}^*_{\ell,0} = (& a_{\ell,0} & 0 & 1 &) & \mathbf{k}^*_{\ell,\lambda} = (\ldots| \; 0 \quad 0 \quad 0 \quad r_{\ell,\lambda} \;)
\end{array}$$

$\mathbf{G}_3$ Additional random-labeling as in the IND-security proof. See Figure 11

$$\begin{array}{rlcccl}
\mathbf{c}_0 = (& \omega & \tau & \xi &) & \mathbf{c}_t = (\ldots| \; \tau \quad 0 \quad \tau z_t \quad (1-b)\cdot u_t \;) \\
\mathbf{k}^*_{\ell,0} = (& a_{\ell,0} & r_{\ell,0} & 1 &) & \mathbf{k}^*_{\ell,\lambda} = (\ldots| \; 0 \quad 0 \quad s_{\ell,\lambda}/z_{t_{\ell,\lambda}} \quad r_{\ell,\lambda} \;)
\end{array}$$

$\mathbf{G}_4$ Index-Ind property to suppress u_t, when $b=0$. See Figure 18

$$\begin{array}{rlcccl}
\mathbf{c}_0 = (& \omega & \tau & \xi &) & \mathbf{c}_t = (\ldots| \; \tau \quad 0 \quad \tau z_t \quad 0 \;) \\
\mathbf{k}^*_{\ell,0} = (& a_{\ell,0} & r_{\ell,0} & 1 &) & \mathbf{k}^*_{\ell,\lambda} = (\ldots| \; 0 \quad 0 \quad s'_{\ell,\lambda}/z_{t_{\ell,\lambda}} \quad r_{\ell,\lambda} \;)
\end{array}$$

$\mathbf{G}_5$ Limitation of independent active leaves

$$\begin{array}{rlcccl}
\mathbf{c}_0 = (& \omega & \tau & \xi &) & \mathbf{c}_t = (\ldots| \; \tau \quad 0 \quad \tau z_t \quad 0 \;) \\
\mathbf{k}^*_{\ell,0} = (& a_{\ell,0} & r_{\ell,0} & 1 &) & \mathbf{k}^*_{\ell,\lambda} = (\ldots| \; 0 \quad 0 \quad s_{\ell,\lambda}/z_{t_{\ell,\lambda}} \quad r_{\ell,\lambda} \;)
\end{array}$$

Fig. 17: Global sequence of games for the Att-IND-security proof of our SA-KP-ABE

$\mathbf{G}_{3.p.0}$ Hybrid game for $\mathbf{G}_3$ and $\mathbf{G}_4$, with $1 \leq p \leq P+1$

$$\begin{array}{llcccccccl}
t < p & \mathbf{c}_t = (& \sigma_t(1,t) & \omega & | & \tau & 0 & \tau z_t & 0 &) \\
t \geq p & \mathbf{c}_t = (& \sigma_t(1,t) & \omega & | & \tau & 0 & \tau z_t & u_t &) \\
 & \mathbf{k}^*_{\ell,0} = (& a_{\ell,0} & r_{\ell,0} & 1 \;) & & & & & \\
 & \mathbf{k}^*_{\ell,\lambda} = (& \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) & a_{\ell,\lambda} & | & 0 & 0 & s_{\ell,\lambda}/z_{t_{\ell,\lambda}} & r_{\ell,\lambda} &)
\end{array}$$

$\mathbf{G}_{3.p.1}$ Formal basis change, on $(\mathbb{D},\mathbb{D}^*)_{5,7}$, to duplicate $r_{\ell,\lambda}$ in the 5-th column

$$\begin{array}{llcccccccl}
 & \mathbf{c}_p = (& \sigma_p(1,p) & \omega & | & \tau & 0 & \tau z_p & u_p &) \\
 & \mathbf{k}^*_{\ell,\lambda} = (& \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) & a_{\ell,\lambda} & | & 0 & r_{\ell,\lambda} & s_{\ell,\lambda}/z_{t_{\ell,\lambda}} & r_{\ell,\lambda} &)
\end{array}$$

$\mathbf{G}_{3.p.2}$ Swap-Ind Property, on $(\mathbb{D},\mathbb{D}^*)_{1,5,7}$, to swap u_p alone in the 5-th column

$$\begin{array}{llcccccccl}
 & \mathbf{c}_p = (& \sigma_p(1,p) & \omega & | & \tau & u_p & \tau z_p & 0 &) \\
t \neq p & \mathbf{c}_t = (& \sigma_t(1,t) & \omega & | & \tau & 0 & \tau z_t & u_t &)
\end{array}$$

$\mathbf{G}_{3.p.3}$ Index-Ind Property, on $(\mathbb{D}^*,\mathbb{D})_{1,2,5}$, between $r_{\ell,\lambda}$ and 0, for $t_{\ell,\lambda} \neq p$

$$\begin{array}{llcccccccl}
 & \mathbf{c}_p = (& \sigma_p(1,p) & \omega & | & \tau & u_p & \tau z_p & 0 &) \\
t_{\ell,\lambda} \neq p & \mathbf{k}^*_{\ell,\lambda} = (& \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) & a_{\ell,\lambda} & | & 0 & 0 & s_{\ell,\lambda}/z_{t_{\ell,\lambda}} & r_{\ell,\lambda} &) \\
t_{\ell,\lambda} = p & \mathbf{k}^*_{\ell,\lambda} = (& \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) & a_{\ell,\lambda} & | & 0 & r_{\ell,\lambda} & s_{\ell,\lambda}/z_{t_{\ell,\lambda}} & r_{\ell,\lambda} &)
\end{array}$$

$\mathbf{G}_{3.p.4}$ SubSpace-Ind Property, on $(\mathbb{D},\mathbb{D}^*)_{6,5}$, between u_p and 0

$$\begin{array}{llcccccccl}
 & \mathbf{c}_p = (& \sigma_p(1,p) & \omega & | & \tau & 0 & \tau z_p & 0 &) \\
t_{\ell,\lambda} \neq p & \mathbf{k}^*_{\ell,\lambda} = (& \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) & a_{\ell,\lambda} & | & 0 & 0 & s_{\ell,\lambda}/z_{t_{\ell,\lambda}} & r_{\ell,\lambda} &) \\
t_{\ell,\lambda} = p & \mathbf{k}^*_{\ell,\lambda} = (& \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) & a_{\ell,\lambda} & | & 0 & r_{\ell,\lambda} & s'_{\ell,\lambda}/z_{t_{\ell,\lambda}} & r_{\ell,\lambda} &)
\end{array}$$

$\mathbf{G}_{3.p.5}$ SubSpace-Ind Property, on $(\mathbb{D}^*,\mathbb{D})_{6,5}$, between $r_{\ell,\lambda}$ and 0, for $t_{\ell,\lambda} = p$

$$\begin{array}{llcccccccl}
t \leq p & \mathbf{c}_t = (& \sigma_t(1,t) & \omega & | & \tau & 0 & \tau z_t & 0 &) \\
t > p & \mathbf{c}_t = (& \sigma_t(1,t) & \omega & | & \tau & 0 & \tau z_t & u_t &) \\
t_{\ell,\lambda} \neq p & \mathbf{k}^*_{\ell,\lambda} = (& \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) & a_{\ell,\lambda} & | & 0 & 0 & s_{\ell,\lambda}/z_{t_{\ell,\lambda}} & r_{\ell,\lambda} &) \\
t_{\ell,\lambda} = p & \mathbf{k}^*_{\ell,\lambda} = (& \pi_{\ell,\lambda}(t_{\ell,\lambda},-1) & a_{\ell,\lambda} & | & 0 & 0 & s'_{\ell,\lambda}/z_{t_{\ell,\lambda}} & r_{\ell,\lambda} &)
\end{array}$$

Fig. 18: Hybrid game on p for the Att-IND-security proof of our SA-KP-ABE, when $b=0$

where $s'_{\ell,\lambda}$ is either the label $s_{\ell,\lambda}$ or an independent random value when $u_{t_{k,\lambda}} \cdot r_{k,\lambda} \neq 0$, in the case $b = 0$. And nothing is changed when $b = 1$. To this aim, we use a different sequence $\mathbf{G}_{3.p.*}$ presented in the Figure 18, when $b = 1$ only, for $p \in \{1, \dots, P\}$, that will modify the p-th ciphertext in the challenge ciphertext, where $\mathbf{G}_3 = \mathbf{G}_{3.1.0}$, and $\mathbf{G}_4 = \mathbf{G}_{3.P+1.0}$.

Game $\mathbf{G}_{3.p.0}$ One thus chooses random scalars and defines the hybrid game for some p, where the first components of the ciphertext are all-valid, and the last ones are real:

$$\begin{aligned}
\mathbf{k}^*_{\ell,0} &= (a_{\ell,0}, r_{\ell,0}, 1)_{\mathbb{B}^*} & \mathbf{k}^*_{\ell,\lambda} &= (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, 0, s_{\ell,\lambda}/z_{t_{\ell,\lambda}}, r_{\ell,\lambda})_{\mathbb{D}^*} & \\
\mathbf{c}_0 &= (\omega, 0, \xi)_{\mathbb{B}} & \mathbf{c}_t &= (\sigma_t(1,t), \omega, \tau, 0, \tau z_t, 0)_{\mathbb{D}} & \text{if } t < p \\
\mathbf{c}_0 &= (\omega, 0, \xi)_{\mathbb{B}} & \mathbf{c}_t &= (\sigma_t(1,t), \omega, \tau, 0, \tau z_t, u_t)_{\mathbb{D}} & \text{if } t \geq p
\end{aligned}$$

Of course, the values $r_{\ell,\lambda}$ and u_t are random in $\mathbb{Z}_q^*$ or 0 according to $\mathcal{L}_{\ell,a}/\mathcal{L}_{\ell,p}$ and Γ_i/Γ_v. In particular, if $u_p = 0$, we can directly go to $\mathbf{G}_{3.p.5}$, as there is no change from this game. But there is no need to know it in advance, and so we can follow this sequence in any case and set u_p in the ciphertext at the challenge-time.

Game $\mathbf{G}_{3.p.1}$ One defines the matrices

$$D = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}_{5,7} \qquad D' = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}_{5,7} \qquad \mathbb{D} = D \cdot \mathbb{V} \qquad \mathbb{D}^* = D' \cdot \mathbb{V}^*$$

which modifies the hidden and secret vectors $\mathbf{d}_6$ and $\mathbf{d}^*_7$, and so are not in the view of the adversary:

$$\begin{aligned}
\mathbf{k}^*_{\ell,\lambda} &= (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, 0, s_{\ell,\lambda}/z_{t_{\ell,\lambda}}, r_{\ell,\lambda})_{\mathbb{V}^*} \\
&= (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, r_{\ell,\lambda}, s_{\ell,\lambda}/z_{t_{\ell,\lambda}}, r_{\ell,\lambda})_{\mathbb{D}^*} \\
\mathbf{c}_t &= (\sigma_t(1,t), \omega, \tau, 0, \tau z_t, 0)_{\mathbb{V}} = (\sigma_t(1,t), \omega, \tau, 0, \tau z_t, 0)_{\mathbb{D}} && \text{if } t < p \\
\mathbf{c}_t &= (\sigma_t(1,t), \omega, \tau, 0, \tau z_t, u_t)_{\mathbb{V}} = (\sigma_t(1,t), \omega, \tau, 0, \tau z_t, u_t)_{\mathbb{D}} && \text{if } t \geq p
\end{aligned}$$

We thus have $\mathsf{Adv}_{3.p.1} = \mathsf{Adv}_{3.p.0}$.

Game $\mathbf{G}_{3.p.2}$ We use the $\mathsf{Swap\text{-}Ind}$-property on $(\mathbb{D}, \mathbb{D}^*)_{1,5,7}$: Indeed, we can consider a triple $(a \cdot G_1, b \cdot G_1, c \cdot G_1)$, where $c = ab + \theta \bmod q$ with either $\theta = 0$ or $\theta = u_p$. We define the matrices

$$D = \begin{pmatrix} 1 & a & -a \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}_{1,5,7} \qquad D' = \begin{pmatrix} 1 & 0 & 0 \\ -a & 1 & 0 \\ a & 0 & 1 \end{pmatrix}_{1,5,7} \qquad \mathbb{D} = D \cdot \mathbb{V} \qquad \mathbb{D}^* = D' \cdot \mathbb{V}^*$$

Note that we can compute all the basis vectors excepted $\mathbf{d}^*_5, \mathbf{d}^*_7$, but we define the keys on the original basis $\mathbb{V}^*$:

$$\begin{aligned}
\mathbf{k}^*_{\ell,\lambda} &= (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, r_{\ell,\lambda}, s_{\ell,\lambda}/z_{t_{\ell,\lambda}}, r_{\ell,\lambda})_{\mathbb{V}^*} \\
&= (\pi_{\ell,\lambda} \cdot t_{\ell,\lambda} + ar_{\ell,\lambda} - ar_{\ell,\lambda}, -\pi_{\ell,\lambda}, a_{\ell,\lambda}, 0, r_{\ell,\lambda}, s_{\ell,\lambda}/z_{t_{\ell,\lambda}}, r_{\ell,\lambda})_{\mathbb{D}^*} \\
&= (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, r_{\ell,\lambda}, s_{\ell,\lambda}/z_{t_{\ell,\lambda}}, r_{\ell,\lambda})_{\mathbb{D}^*} \\
\mathbf{c}_t &= (\sigma_t(1,t), \omega, \tau, 0, \tau z_t, 0)_{\mathbb{D}} && \text{if } t < p \\
\mathbf{c}_p &= (\sigma(1,p), \omega, \tau, 0, \tau z_p, u_p)_{\mathbb{D}} + (b(1,p), 0, 0, c, 0, -c)_{\mathbb{V}} \\
&= (\sigma(1,p), \omega, \tau, 0, \tau z_p, u_p)_{\mathbb{D}} + (b(1,p), 0, 0, c - ab, 0, -c + ab)_{\mathbb{D}} \\
&= ((\sigma + b)(1,p), \omega, \tau, \theta, \tau z_p, u_p - \theta)_{\mathbb{D}} \\
\mathbf{c}_t &= (\sigma_t(1,t), \omega, \tau, 0, \tau z_t, u_t)_{\mathbb{D}} && \text{if } t > p
\end{aligned}$$

With $\theta = 0$, this is as in the previous game, where $\sigma_p = \sigma + b$. When $\theta = u_p$, this is the current game: $\mathsf{Adv}_{3.p.1} - \mathsf{Adv}_{3.p.2} \leq 2 \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t)$.

Game $\mathbf{G}_{3.p.3}$ We make all the $r_{\ell,\lambda}$ values (at the 5-th hidden position) in the keys to be 0, excepted when $t_{\ell,\lambda} = p$. For that, we iteratively replace all the values by zero, using Index-Ind-property, in another level of sequence of hybrid games, for $\gamma \in \{1, \dots, P\}\backslash\{p\}$:

Game $\mathbf{G}_{3.p.2.\gamma}$: We consider

$$\begin{aligned}
\mathbf{k}^*_{\ell,\lambda} &= (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, r_{\ell,\lambda}, s_{\ell,\lambda}/z_{t_{\ell,\lambda}}, r_{\ell,\lambda})_{\mathbb{D}^*} && \text{if } t_{\ell,\lambda} = p\\
\mathbf{k}^*_{\ell,\lambda} &= (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, 0, s_{\ell,\lambda}/z_{t_{\ell,\lambda}}, r_{\ell,\lambda})_{\mathbb{D}^*} && \text{if } p \neq t_{\ell,\lambda} < \gamma\\
\mathbf{k}^*_{\ell,\lambda} &= (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, r_{\ell,\lambda}, s_{\ell,\lambda}/z_{t_{\ell,\lambda}}, r_{\ell,\lambda})_{\mathbb{D}^*} && \text{if } p \neq t_{\ell,\lambda} \geq \gamma\\
\mathbf{c}_t &= (\sigma_t(1, t), \omega, \tau, 0, \tau z_t, 0)_{\mathbb{D}} && \text{if } t < p\\
\mathbf{c}_p &= (\sigma_p(1, p), \omega, \tau, u_p, \tau z_p, 0)_{\mathbb{D}}\\
\mathbf{c}_t &= (\sigma_t(1, t), \omega, \tau, 0, \tau z_t, u_t)_{\mathbb{D}} && \text{if } t > p
\end{aligned}$$

When $\gamma = 1$, this is the previous game: $\mathbf{G}_{3.p.2.1} = \mathbf{G}_{3.p.2}$, whereas with $\gamma = P + 1$, this is the current game: $\mathbf{G}_{3.p.2.P+1} = \mathbf{G}_{3.p.3}$.
We can consider a triple $(a \cdot G_2, b \cdot G_2, c \cdot G_2)$, where $c = ab + \zeta \bmod q$ with either $\zeta = 0$ or $\zeta = 1$. We define the matrices

$$D' = \frac{1}{\gamma - p} \times \begin{pmatrix} 1 & -1 & a\\ \gamma & -p & ap\\ 0 & 0 & \gamma - p \end{pmatrix}_{1,2,5} \qquad D = \begin{pmatrix} -p & -\gamma & 0\\ 1 & 1 & 0\\ 0 & a & 1 \end{pmatrix}_{1,2,5}$$

and then $\mathbb{D} = D \cdot \mathbb{V}$, $\mathbb{D}^* = D' \cdot \mathbb{V}^*$: we cannot compute $\mathbf{d}_5$, but this component is always 0, excepted for $\mathbf{c}_p$ that we will define in the original basis $\mathbb{V}$. One chooses additional random scalars $\beta_{\ell,\lambda} \xleftarrow{\$} \mathbb{Z}_q$, for all (ℓ, λ) such that $t_{\ell,\lambda} = \gamma$, to virtually set $b_{\ell,\lambda} = -r_{\ell,\lambda} \cdot b + \beta_{\ell,\lambda}$ and $c_{\ell,\lambda} = -r_{\ell,\lambda} \cdot c + \beta_{\ell,\lambda} \cdot a$, $c_{\ell,\lambda} - ab_{\ell,\lambda} = -r_{\ell,\lambda} \cdot \zeta$. One can set

$$\begin{aligned}
\mathbf{k}^*_{\ell,\lambda} &= (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, r_{\ell,\lambda}, s_{\ell,\lambda}/z_{t_{\ell,\lambda}}, r_{\ell,\lambda})_{\mathbb{D}^*} && \text{if } t_{\ell,\lambda} = p\\
\mathbf{k}^*_{\ell,\lambda} &= (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, 0, s_{\ell,\lambda}/z_{t_{\ell,\lambda}}, r_{\ell,\lambda})_{\mathbb{D}^*} && \text{if } p \neq t_{\ell,\lambda} < \gamma\\
\mathbf{k}^*_{\ell,\lambda} &= (b_{\ell,\lambda}(0, -1), a_{\ell,\lambda}, 0, r_{\ell,\lambda} + c_{\ell,\lambda}, s_{\ell,\lambda}/z_{t_{\ell,\lambda}}, r_{\ell,\lambda})_{\mathbb{V}^*} && \text{if } t_{\ell,\lambda} = \gamma\\
&= (b_{\ell,\lambda}(\gamma, -1), a_{\ell,\lambda}, 0, r_{\ell,\lambda} + c_{\ell,\lambda} - ab_{\ell,\lambda}, s_{\ell,\lambda}/z_{t_{\ell,\lambda}}, r_{\ell,\lambda})_{\mathbb{D}^*}\\
&= (b_{\ell,\lambda}(\gamma, -1), a_{\ell,\lambda}, 0, r_{\ell,\lambda} \cdot (1 - \zeta), s_{\ell,\lambda}/z_{t_{\ell,\lambda}}, r_{\ell,\lambda})_{\mathbb{D}^*}\\
\mathbf{k}^*_{\ell,\lambda} &= (\pi_{\ell,\lambda}(t_{\ell,\lambda}, -1), a_{\ell,\lambda}, 0, r_{\ell,\lambda}, s_{\ell,\lambda}/z_{t_{\ell,\lambda}}, r_{\ell,\lambda})_{\mathbb{D}^*} && \text{if } p \neq t_{\ell,\lambda} > \gamma\\
\mathbf{c}_t &= (\sigma_t(1, t), \omega, \tau, 0, \tau z_t, 0)_{\mathbb{D}} && \text{if } t < p\\
\mathbf{c}_p &= ((p - \gamma)\sigma(0, 1), \omega, \tau, u_p, \tau z_p, 0)_{\mathbb{V}}\\
&= (\sigma(1, p) + au_p(1, p)/(\gamma - p), \omega, \tau, u_p, \tau z_p, 0)_{\mathbb{D}}\\
&= ((\sigma + au_p/(\gamma - p))(1, p), \omega, \tau, u_p, \tau z_p, 0)_{\mathbb{D}}\\
&= (\sigma_p(1, p), \omega, \tau, u_p, \tau z_p, 0)_{\mathbb{D}}\\
\mathbf{c}_t &= (\sigma_t(1, t), \omega, \tau, 0, \tau z_t, u_t)_{\mathbb{D}} && \text{if } t > p
\end{aligned}$$

When $\zeta = 0$, this $\mathbf{G}_{3.p.2.\gamma}$, when $\zeta = 1$, this is $\mathbf{G}_{3.p.2.\gamma+1}$.

Game $\mathbf{G}_{3.p.4}$ We use the SubSpace-Ind-property on $(\mathbb{D}, \mathbb{D}^*)_{6,5}$: Indeed, we can consider a triple $(a \cdot G_1, b \cdot G_1, c \cdot G_1)$, where $c = ab + \theta \bmod q$ with either $\theta = 0$ or $\theta = u_p$. We define the matrices

$$D = \begin{pmatrix} 1 & 0\\ a & 1 \end{pmatrix}_{5,6} \qquad D' = \begin{pmatrix} 1 & -a\\ 0 & 1 \end{pmatrix}_{5,6} \qquad \mathbb{D} = D \cdot \mathbb{V} \qquad \mathbb{D}^* = D' \cdot \mathbb{V}^*$$

Note that we can compute all the basis vectors excepted $\mathbf{d}_5^*$ that is not public, and not used excepted for the keys with $t_{\ell,\lambda} = p$, which will be defined in the original basis $\mathbb{V}^*$:

$$\begin{aligned}
\mathbf{k}_{\ell,\lambda}^* &= (\pi_{\ell,\lambda}(p,-1), a_{\ell,\lambda}, 0, r_{\ell,\lambda}, s_{\ell,\lambda}/z_p, r_{\ell,\lambda})_{\mathbb{V}^*} \\
&= (\pi_{\ell,\lambda}(p,-1), a_{\ell,\lambda}, 0, r_{\ell,\lambda}, s_{\ell,\lambda}/z_p + ar_{\ell,\lambda}, r_{\ell,\lambda})_{\mathbb{D}^*} \\
&= (\pi_{\ell,\lambda}(p,-1), a_{\ell,\lambda}, 0, r_{\ell,\lambda}, s'_{\ell,\lambda}/z_p, r_{\ell,\lambda})_{\mathbb{D}^*} \\
\mathbf{c}_t &= (\sigma_t(1,t), \omega, b, 0, bz_t, 0)_{\mathbb{D}} && \text{if } t < p \\
\mathbf{c}_p &= (\sigma_p(1,p), \omega, b, c, bz_p, 0)_{\mathbb{V}} = (\sigma_p(1,p), \omega, b, c - ab, bz_p, 0)_{\mathbb{D}} \\
&= (\sigma_p(1,p), \omega, b, \theta, bz_p, 0)_{\mathbb{D}} \\
\mathbf{c}_t &= (\sigma_t(1,t), \omega, b, 0, bz_t, u_t)_{\mathbb{D}} && \text{if } t > p
\end{aligned}$$

When $\theta = 0$, this is this game, whereas when $\theta = u_p$, this is the previous game, with $\tau = b$ and $s'_{\ell,\lambda} = s_{\ell,\lambda} + az_p r_{\ell,\lambda}$ a new random and independent value for each active leaf associated to the attribute p.

Game $\mathbf{G}_{3.p.5}$ We use the SubSpace-Ind-property on $(\mathbb{D}^*, \mathbb{D})_{6,5}$: Indeed, we can consider a triple $(a \cdot G_2, b \cdot G_2, c \cdot G_2)$, where $c = ab + \zeta \bmod q$ with either $\zeta = 0$ or $\zeta = 1$. We define the matrices

$$D' = \begin{pmatrix} 1 & 0 \\ a & 1 \end{pmatrix}_{5,6} \qquad D = \begin{pmatrix} 1 & -a \\ 0 & 1 \end{pmatrix}_{5,6} \qquad \mathbb{D} = D \cdot \mathbb{V} \qquad \mathbb{D}^* = D' \cdot \mathbb{V}^*$$

Note that we can compute all the basis vectors excepted $\mathbf{d}_5$ that is not public, and not used in the ciphertext. All the vectors can be computed in the new bases, excepted the keys for $t_{\ell,\lambda} = p$, for which one chooses additional random scalars $\beta_{\ell,\lambda} \stackrel{\$}{\leftarrow} \mathbb{Z}_q$, to virtually set $b_{\ell,\lambda} = r_{\ell,\lambda} \cdot b + \beta_{\ell,\lambda}$ and $c_{\ell,\lambda} = r_{\ell,\lambda} \cdot c + \beta_{\ell,\lambda} \cdot a$, $c_{\ell,\lambda} - ab_{\ell,\lambda} = r_{\ell,\lambda} \cdot \zeta$.

$$\begin{aligned}
\mathbf{k}_{\ell,\lambda}^* &= (\pi_{\ell,\lambda}(p,-1), a_{\ell,\lambda}, 0, c_{\ell,\lambda}, b_{\ell,\lambda}, r_{\ell,\lambda})_{\mathbb{V}^*} \\
&= (\pi_{\ell,\lambda}(p,-1), a_{\ell,\lambda}, 0, c_{\ell,\lambda} - ab_{\ell,\lambda}, b_{\ell,\lambda}, r_{\ell,\lambda})_{\mathbb{D}^*} \\
&= (\pi_{\ell,\lambda}(p,-1), a_{\ell,\lambda}, 0, \zeta \cdot r_{\ell,\lambda}, b_{\ell,\lambda}, r_{\ell,\lambda})_{\mathbb{D}^*}
\end{aligned}$$

When $\zeta = 0$, this is this game, whereas when $\zeta = 1$, this is the previous game, with $s'_{\ell,\lambda} = z_p \cdot b_{\ell,\lambda}$, a truly random and independent value for each active leaf associated to the attribute p.

Game $\mathbf{G}_5$: Under the assumption of independent active leaves with respect to the set of attributes $\Gamma = \Gamma_v \cup \Gamma_i$ in the challenge ciphertext, the random values $s'_{\ell,\lambda}$ are indistinguishable from real labels $s'_{\ell,\lambda}$. Indeed, labels that correspond to leaves that are associated to attributes not in Γ are unknown, as the masks z_t are not revealed. This shows that the advantage of the adversary in this last game is 0.