Attribute-Based Signatures with Advanced Delegation, and Tracing

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

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

Abstract. Attribute-based cryptography allows fine-grained control on the use of the private key. In particular, attribute-based signature (ABS) specifies the capabilities of the signer, which can only sign messages associated to a policy that is authorized by his set of attributes. Furthermore, we can expect signature to not leak any information about the identity of the signer. ABS is a useful tool for identity-preserving authentication process which requires granular access-control, and can furthermore be enhanced with additional properties, for example delegation where users are able to manage a set of keys derived from their original one.

In this paper, we address delegation of signing keys. Our first delegation works for any subset of the original attributes, which is the intuitive approach of delegation. Furthermore, we also provide another kind of delegation where the delegator can choose a policy at delegation time to produce keys that can sign any message under this specific policy. This last approach to delegation is a direct application of a new version of the *indexing* technique, which was first introduced by Okamoto and Takashima in order to prove adaptive security in ABS and its counterpart for encryption, ABE. On top of that, we prove that our scheme is compatible with a well studied feature of ABS, traceability, by using an approach based on Linearly-Homomorphic signatures. All our schemes also guarantee the anonymity of the real signer.

The unforgeability of our schemes is proven using the SXDH assumption, and our constructions use the Dual Pairing Vector Spaces (DPVS) framework developed by Okamoto and Takashima, which has been widely used for all kind of attribute and functional cryptography mechanisms.

1 Introduction

Digital signatures have key applications into emerging technologies : E-signatures, smart contracts on blockchain, and authentication to online services. All of them require some kind of signature from the user, that must then be verified by another party.

Furthermore, some applications are reliant on the anonymity granted to its participants, like e-vote and anonymous auctions. It can be because of mandatory data privacy regulations (US Data Privacy Laws, Europe's GDPR), or because it provides a concurrential advantage, as users are concerned about their privacy. Hence, these applications should only enforce verifications that are strictly necessary, like verifying the legitimacy of a signer to sign a contract or access a service, without revealing the identity of the signer.

As mentioned in CT-RSA 2021 [DGK+21], blockchain applications like cryptocurrencies must also ensure accountability and identity management to follow regulatory requirements (*Know Your Customer/Anti-Money Laundering*), as well as ensure public verifiability to keep their users' trust. Therefore, suitable primitives should allow to reconcile these regulations and principles with anonymity. This can be achieved through tracing, which also prevents abuse and makes signers accountable. This crosses out fully anonymous primitives like ring signatures [RST01].

One could use solutions based on group signatures [Cv91], where a designated tracer can remove the anonymity of a signer, but they unfortunately lack expressivity. Indeed, they can only express membership of a group of potential signers, and not more complex policies, based on boolean expressions. This is not enough for the many applications we have mentioned, where we need to verify precise but varied statements and conditionals.

Attribute-based signatures (ABS), introduced in [MPR11], combine anonymity, expressivity and traceability. It allows users to sign a message for a policy that can be validated with a set of attributes. ABS are very appropriate for situations where high-granularity of the policy is required: management of rights for members of a department, industrial contracts, financial operations, and blockchain operations.

The expressivity and granularity of ABS can be pushed even further with delegation, where users can create new sub-keys with restricted rights from their original keys. This technique grants users the capacity to manage any type of rights on a case-by-case basis, without the need to refer to an authority for approval.

Direct applications of delegation can be found in project management and team management. Since delegated keys can be tailored by the delegator depending on its needs, they can include restricted timeframes of validity, scope perimeters, and authorizations of specific sets of actions. This can be useful to integrate temporary members in an existing team, or allow cooperation between members of two different teams in a secure way. Another application is the management of many devices by a single user. In this scenario, a user is in possession of many devices, and he wishes to configure them following common security practices like the least privilege principle. Thanks to delegation, this can be done dynamically on a device-per-device basis, without referring to an authority.

These possibilities make delegation a promising feature for ABS, as it furthers the granularity needed to manage modern applications (both from a security and a functionality standpoint), while maintaining the anonymity and traceability required by different regulations.

Related Work. Attribute-Based Signature was introduced by Maji *et al.* [MPR11], as the signature version of Attribute-Based Encryption (ABE) [GPSW06]. They define what one can expect as unforgeability for ABS: one is unable to produce a convincing signature for any policy he wouldn't satisfy. Unforgeability can be *adaptive*, where the adversary can choose the challenge policy at the moment it outputs the forgery, or *selective*, a weaker version where the adversary must choose the challenge policy at the beginning of the game. They also introduce privacy (or anonymity) for ABS, where any verifier does not learn anything on the identity nor the attributes of the signer when seeing a signature, except that the signature is valid or not, with respect to the claimed policy.

Our constructions are based on Okamoto and Takashima's [OT11] original work in the Dual-Pairing Vector Space (DPVS) framework, which is still the basis for their recent work for signatures [DOT19]. The DPVS allows to prove adaptive unforgeability in ABS and adaptive security in its ABE counterpart [OT11,OT12]. Most of their previous works are based on the DLIN assumption or variants. Along with Attrapadung *et al.*, they focused on the expressivity of the policy [SKAH18].

A common feature for ABS is tracing [DZL14,Gha15] where a dedicated tracing authority can remove the anonymity of signatures for accountability. The most common approach for traceability in the litterature is the one proposed by Ghadafi *et al.* [EGK14], where a designated tracer with a secret tracing key can produce a proof of identity of the signer, which is verifiable by any third party with a public verification key. It relies on attaching an encryption of the identity of the signer to signatures, along with a NIZK proof that the identity is the one used to sign. The tracer simply own the secret key to decrypt any identity when tracing a signature, and can produce NIZK proof of the identity to any verifying third-party to prove tracing was done correctly.

Another functionality which has not received much attention in ABS is the delegation functionality. A work from [LMY14] proposes proxy signature with a warrant for ABS, which works as a delegation of pre-set policies. Their work is in a restricted setting, where policies are a subset cover of attributes, and their construction is selectively secure and doesn't consider anonymity.

A recent line of work from Manulis *et al.* [DGM18,GM19] explores a Hierarchical ABS with a focus on the management of intermediate authorities, where the delegation keeps track of the delegation path containing all the authorities that participated in the creation of someone's key. While this allows delegation via intermediate authorities, the tracing of the delegation path makes the size of keys and signatures linear in the number of attributes *and* the length of the delegation path. This additional cost cannot be avoided as their delegation and tracing are intermingled. Since delegation cannot be separated from tracing in their construction, this also means that delegation cannot be done without the NIZK that is used for tracing.

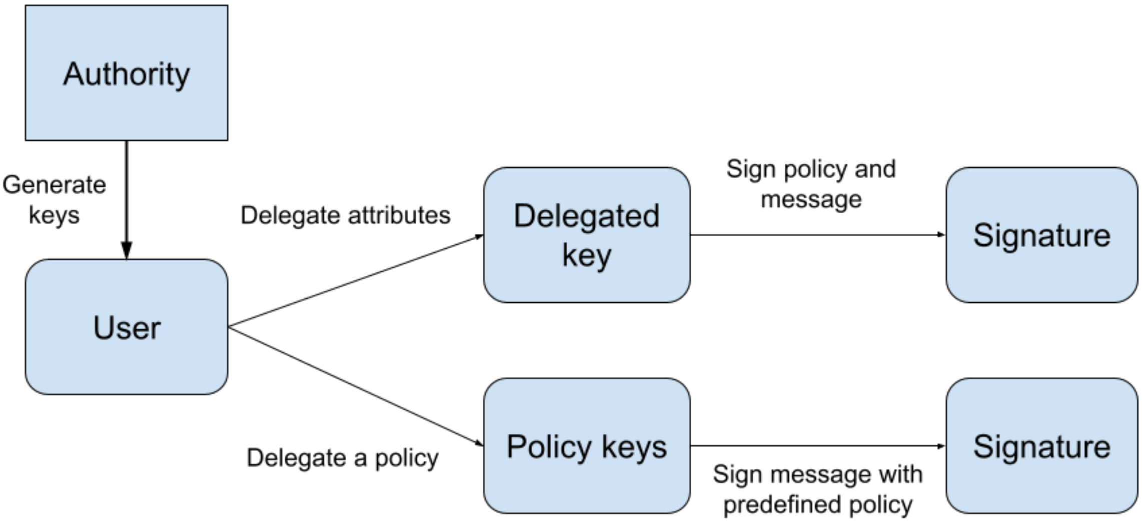

Fig. 1. Two types of delegation : attributes and policies.

Contributions. In this paper, we propose and prove an ABS construction with two different types of delegation: delegation of attributes and delegation of policies. A depiction of these functionalities is given in Figure 1. This construction is existentially unforgeable, as well as perfectly anonymous, under the following two standard assumptions: the SXDH assumption in the standard model, and the collision-resistance of some hash functions.

We also present a new version of indexing for DPVS, that builds on the one introduced by [OT12]. We then show as an application that this new version of indexing can be used to separate the commitment between message and policy with a hash function in our signature scheme, which is the core component of our delegation of policies.

Finally, we present a construction for ABS with traceability, which is compatible with our first construction with delegation. It is existentially unforgeable under the SXDH assumption in the standard model, and the collision-resistance of some hash functions. It is also computationally anonymous under these same assumptions, and the perfect zero-knowledge of the Square Diffie-Hellman problem [HPP20] in the Random Oracle Model (ROM). The traceability stands in the ROM, and relies on the security of a Linearly-Homomorphic signature scheme, the simulation-extractability of some non-interactive zero-knowledge proof (NIZK) and the soundness of some other NIZK. We exploit the scheme from [HPP20], whose security is proven in the generic bilinear group model.

The keys and signatures of both our schemes are linear in the number of attributes involved, with performances comparable to [OT11]. See Figure 2.

2 Preliminaries

Our constructions will heavily use the Dual Pairing Vector Spaces (DPVS), proposed for efficient schemes with adaptive security [LOS$^+$10,OT12], in the same vein as Dual System Encryption (DSE) [Wat09,LW10], in either prime-order groups under the DLIN assumption or pairings on composite-order elliptic curves, and thereafter on the SXDH assumption in a pairing-friendly

Feature	[OT11]	[GM19]	Ours
Unforge. assumpt.	DLIN	q-type, SXDH, GGM	SXDH
Delegation	×	✓	✓
Traceability	×	Delegation path	Original delegator
Trace. assumpt.	×	×	GGM, ROM
Signature size	$\mathbb{G}_2 : 9t + 11$	$\mathbb{G}_1 : 6(2\ell - 1)t + 24$ $\mathbb{G}_2 : 4(2.5\ell - 1)t + 17$	$\mathbb{G}_2 : 10t + 14$

Fig. 2. Comparison with Related Work. t is the number of attributes used in a signature, and ℓ is the height of the delegation hierarchy.

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 $\mathbb{G}_t$, where G_1 (respectively G_2) is a generator of $\mathbb{G}_1$ (respectively $\mathbb{G}_2$), and all the groups are of prime order q. We will use additive notation for $\mathbb{G}_1$ and $\mathbb{G}_2$, and multiplicative notation in $\mathbb{G}_t$.

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*

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

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. Eventually, we will make the following more general Symmetric eXternal Diffie-Hellman (SXDH) Assumption which makes the DDH assumptions in both $\mathbb{G}_1$ and $\mathbb{G}_2$. Then, we define $\mathsf{Adv}^{\mathsf{sxdh}}(T) = \max\{\mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(T), \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(T)\}$.

2.1 Dual Pairing Vector Spaces

We will use the framework from [DGP22] which uses a lower number of bases than the original framework from Okamoto and Takashima [OT11]. One could also consider the more recent framework from Datta *et al.* [DOT19] for a lower number of bases, but it also implies a higher number of specific sub-problems for the proof, which makes it less modular for the security proof.

To define Dual Pairing Vector Spaces (DPVS) under the SXDH assumption, we consider the additional law between an element $X \in \mathbb{G}_1^n$ and $Y \in \mathbb{G}_2^n$: $X \times Y \overset{\text{def}}{=} \prod_i e(X_i, Y_i)$. If $X = (X_i)_i = \vec{x} \cdot G_1 \in \mathbb{G}_1^n$ and $Y = (Y_i)_i = \vec{y} \cdot G_2 \in \mathbb{G}_2^n$: $(\vec{x} \cdot G_1) \times (\vec{y} \cdot G_2) = X \times Y = \prod_i e(X_i, Y_i) = g_t^{\langle \vec{x}, \vec{y} \rangle}$, 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 $\mathbf{b}_i = \vec{b}_i \cdot G_1$. This allows us to note $(a_1, \dots, a_n)_{\mathbb{B}} = \sum_i a_i \cdot \mathbf{b}_i$.

Such a basis $\mathcal{B}$ is equivalent to a random invertible matrix $B \overset{\$}{\leftarrow} \mathrm{GL}_n(\mathbb{Z}_q)$, 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 $i, j \in \{1, \dots, n\}$: $\mathbb{B}$ and $\mathbb{B}^*$ are called *Dual Orthogonal Bases*. A pairing-friendly setting with such dual orthogonal bases $\mathbb{B}$ and $\mathbb{B}^*$ of size n is called a *Dual Pairing Vector Space*.

2.2 Change of Basis

The security games will heavily rely on indistinguishable change of basis. We recap the indistinguishable modifications on *random* dual orthogonal bases $\mathbb{B}$ and $\mathbb{B}^*$, under the DDH assumption in $\mathbb{G}_1$ (can also be applied in $\mathbb{G}_2$), proven in [DGP22]. We illustrate these theorems in the Figure 6, in the Supplementary Materials.

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 $(y_1,y_2,\dots,y_n)_{\mathbb{B}^*}$, for chosen $y_2,\dots,y_n \in \mathbb{Z}_q$, but unknown random $y_1 \stackrel{\$}{\leftarrow} \mathbb{Z}_q$, one cannot distinguish the vectors $(x_1,x_2',x_3,\dots,x_n)_{\mathbb{B}}$ and $(x_1,x_2,x_3,\dots,x_n)_{\mathbb{B}}$, for chosen $x_2',x_2,\dots,x_n \in \mathbb{Z}_q$, but unknown random $x_1 \stackrel{\$}{\leftarrow} \mathbb{Z}_q$.

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 $(y_1,y_1,y_3,\dots,y_n)_{\mathbb{B}^*}$, for chosen $y_1,y_3,\dots,y_n \in \mathbb{Z}_q$, one cannot distinguish the vectors $(x_1,0,x_3,x_4,\dots,x_n)_{\mathbb{B}}$ and $(0,x_1,x_3,x_4,\dots,x_n)_{\mathbb{B}}$, for chosen $x_1,x_4,\dots,x_n \in \mathbb{Z}_q$, but unknown random $x_3 \stackrel{\$}{\leftarrow} \mathbb{Z}_q$.

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 $(\pi\cdot(t,-1),y_3,\dots,y_n)_{\mathbb{B}^*}$, for chosen $y_3,\dots,y_n \in \mathbb{Z}_q$, but unknown random $\pi \stackrel{\$}{\leftarrow} \mathbb{Z}_q$, and for any chosen $p \neq t \in \mathbb{Z}_q$, 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 $\sigma \stackrel{\$}{\leftarrow} \mathbb{Z}_q$.

We also present a new version of Index-Ind in dimension 3 (instead of 2), with a sketch of the proof in Figure 7 in the Appendix, and the full proof is presented in Appendix C.2.

Theorem 2 (Index-Ind Property). *In $(\mathbb{B},\mathbb{B}^*)$ of dimension 6, from the view of $(\mathbf{b}_1,\mathbf{b}_2,\mathbf{b}_3,\mathbf{b}_1^*,\mathbf{b}_2^*,\mathbf{b}_3^*,\mathbf{b}_4^*)$, and any vector $\mathbf{u}=(\pi\cdot(x+\rho x',-1,-\rho),\beta,0,0)_{\mathbb{B}}$, for chosen $x,x',\beta \in \mathbb{Z}_q$, but unknown random $\pi,\rho \stackrel{\$}{\leftarrow} \mathbb{Z}_q$, and for any chosen $(y,y') \neq (x,x') \in \mathbb{Z}_q^2$, one cannot distinguish the vectors $\mathbf{v}_0^*=(\sigma\cdot(1,y,y'),0,0,0)_{\mathbb{B}^*}$ and $\mathbf{v}_1^*=(\sigma\cdot(1,y,y'),\alpha,0,0)_{\mathbb{B}^*}$, for chosen $\alpha \in \mathbb{Z}_q$, but unknown random $\sigma \stackrel{\$}{\leftarrow} \mathbb{Z}_q$, with an advantage better than $4\times\mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t)+2\times\mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t)$.*

An important application of this theorem for our construction is that we can now have independent commitment for the message and for the policy, contrary to [OT11]. In their construction, the message m and the policy $\mathcal{T}$ are commited together inside a single call of a hash function $\mathcal{H}$, which can then be used as part of an indexing in dimension 2, where $\mathcal{H}(m,\mathcal{T})$ plays the role of p in the 2-Dimension property. A direct application of the 3-Dimensional Indexing allows to decorrelate $\mathcal{H}(m,\mathcal{T})$ into $\mathcal{H}(m)$ and $\mathcal{H}(\mathcal{T})$, which can then be used to build the new delegation functionalities that we present in our construction.

2.3 Attribute-Based Signature

Standard ABS Definition. Attribute-Based Signatures (ABS) have been formalized in [MPR11], with attributes in the keys and policies in the signatures:

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

$\mathsf{KeyGen}(\mathsf{MK},\mathsf{id},\Gamma)$**.** For a master secret key MK, an identity id and a list of attributes Γ, the algorithm outputs a private key $\mathsf{SK}_{\mathsf{id},\Gamma}$ specific to the user id and the set of attributes Γ;

$\mathsf{Sig}(\mathsf{SK}_{\mathsf{id},\Gamma},m,\mathcal{T})$**.** For a private key $\mathsf{SK}_{\mathsf{id},\Gamma}$, on a set of attributes Γ, a message m and a policy $\mathcal{T}$ satisfied by Γ, the algorithm outputs a signature σ;

$\mathsf{Verif}(\mathsf{PK},m,\mathcal{T},\sigma)$**.** Given the public parameters PK, a signature σ for a message m under a policy $\mathcal{T}$, the algorithm outputs 1 for accept or 0 for reject.

For correctness, the Verif algorithm should output 1 with overwhelming probability on $(\sigma,m,\mathcal{T})$ if σ has been generated on m and $\mathcal{T}$, with a private key $\mathsf{SK}_{\mathsf{id},\Gamma}$ that has been generated from the KeyGen algorithm associated on a set Γ that satisfies $\mathcal{T}$. We will note $\mathcal{T}(\Gamma)=1$ when $\mathcal{T}$ is satisfied by Γ, and $\mathcal{T}(\Gamma)=0$ otherwise.

ABS with Delegation. We now consider two different kinds of delegation. The first one is to delegate a subset of attributes from a key, which is the usual approach to delegation where keys are a set of attributes. The other one is to choose a policy which can be validated by the delegator key, and create a new delegated key that is commited to this policy. The new delegated key can sign any new message under the commited policy. We will prove that both these delegations are fully compatible with anonymity.

For ABS with delegation, in addition to the initial definition of an ABS, we also consider delegation algorithms, with an additional signing algorithm:

$\mathsf{Delegate\text{-}Attributes}(\mathsf{SK}_{\mathsf{id},\Gamma}, \bar{\mathsf{id}}, \Gamma')$. From $\mathsf{SK}_{\mathsf{id},\Gamma}$, and for a subset $\Gamma' \subset \Gamma$, one can derive a signing key $\mathsf{SK}_{\bar{\mathsf{id}},\Gamma'}$ for a user $\bar{\mathsf{id}}$.

$\mathsf{Delegate\text{-}Policy}(\mathsf{SK}_{\mathsf{id},\Gamma}, \bar{\mathsf{id}}, \mathcal{T})$. For a private key $\mathsf{SK}_{\mathsf{id},\Gamma}$ on a set of attributes Γ and a policy $\mathcal{T}$ satisfied by Γ, the algorithm outputs a policy key $\mathsf{SK}_{\bar{\mathsf{id}},\mathcal{T}}$;

$\mathsf{DelegateSig}(\mathsf{SK}_{\mathsf{id},\mathcal{T}}, m)$. For a delegated key $\mathsf{SK}_{\mathsf{id},\mathcal{T}}$ on a policy $\mathcal{T}$ and a message m, the algorithm outputs a signature σ;

The delegated keys from the first algorithm can be used in a similar way as a fresh key. For this reason, when we refer to a key in the following, it can be from either KeyGen or Delegate-Attributes without distinction, except if specified otherwise. It thus allows hierarchical delegation of attributes. On the other hand, policy delegation provides different keys, hence another signing algorithm, which is why we'll refer to them as *policy keys*.

For the correctness, we add that the Verif algorithm should output 1 with overwhelming probability on $(\sigma, m, \mathcal{T})$ even if σ has been generated on m with a policy key $\mathsf{SK}_{\bar{\mathsf{id}},\mathcal{T}}$. In both cases, we note $\bar{\mathsf{id}}$ the new full identity associated to the keys, which could be formed for example by concatenation: $\bar{\mathsf{id}} = \mathsf{id}||\mathsf{id}'$, for some id', and even possibly a longer chain, as only delegated attributes under the exact same chain might be combined as a new key.

2.4 Security Model

As for any signature scheme, with ABS, one should not be able to produce a valid signature under a policy $\mathcal{T}$ if one does not own the appropriate attributes to fulfill it. However, we also account for the nature of delegated keys: One should not be able to produce a valid signature under a policy $\mathcal{T}$ if one does not own the appropriate attributes or the delegated key for unforgeability.

In particular, for the delegation of attributes, we consider that the adversary will have access to delegated keys via the ODelegateAttributes oracle, which he can use on keys he previously queried for via either OKeyGen or ODelegateAttributes. Regarding the policy delegation, we add two additional oracles: ODelegatePolicy to generate a policy key from a previous OKeyGen or ODelegateAttributes query, and ODelegateSig to obtain a signature from a policy key ODelegatePolicy that has already been queried. None of the keys are actually revealed to the adversary, unless he queries specifically for them via OGet, to model the real information learnt by the adversary. Indeed, some keys can be generated, but only as source of delegations, and only delegated keys will be known to the adversary, in case the adversary is just a delegatee.

Definition 3 (Existential Unforgeability). *EUF for ABS with delegation is defined by the following game between the adversary and a challenger:*

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

Oracles: *The following oracles can be called in any order and any number of times:*

$\mathsf{OKeyGen}(\mathsf{id}, \Gamma)$: *to model* KeyGen*-queries for any identity* id *and any set of attributes Γ of its choice, and gets back the index k of the key;*

$\mathsf{ODelegateAttributes}(k, \bar{\mathsf{id}}, \Gamma')$: *to model* Delegate-Attributes*-queries for identity* $\bar{\mathsf{id}}$ *and any subset of attributes $\Gamma' \subset \Gamma$, for the k-indexed generated key from Γ. It generates the signing key but only outputs the index k' of the new key;*

$\mathsf{ODelegatePolicy}(k, \bar{\mathsf{id}}, \mathcal{T})$: *to model* Delegate-Policy*-queries for identity* $\bar{\mathsf{id}}$ *and any policy $\mathcal{T}$, from the k-indexed generated key for Γ so that $\mathcal{T}(\Gamma) = 1$. It generates the new policy key but only outputs the index k' of the new policy key;*

$\mathsf{OGet}(k)$: *the adversary obtains the k-indexed key generated by one of the above oracles;*

$\mathsf{OSig}(\mathsf{id}, m, \mathcal{T}, k)$: *to model* Sig*-queries under any policy $\mathcal{T}$ of its choice for a message m, for the identity* id*, and a key index k. It generates and outputs the signature;*

$\mathsf{ODelegateSig}(\bar{\mathsf{id}}, m, \mathcal{T}, k)$: *to model* DelegateSig*-queries for any message m, for identity $\bar{\mathsf{id}}$, policy $\mathcal{T}$, and key index k. It generates and outputs the signature.*

Finalize(b'): *The adversary outputs a forgery $(m', \mathcal{T}', \sigma')$. If for some attribute set Γ corresponding to a key asked to the* OGet *oracle, $\mathcal{T}'(\Gamma) = 1$, or if the adversary queried* OSig *or* ODelegateSig *on $(m', \mathcal{T}')$, or if the adversary queries* ODelegatePolicy *on $\mathcal{T}'$, one outputs* 0*. Otherwise one outputs* $\mathsf{Verif}(\mathsf{PK}, m', \mathcal{T}', \sigma')$.

The advantage $\mathsf{Adv}^{\mathsf{del\text{-}euf}}(\mathcal{A})$ of an adversary $\mathcal{A}$ in this game is defined as the probability to output 1.

As usual, the Finalize-step excludes trivial attacks, where the adversary owns a key able to generate an acceptable signature or just forwards a query asked to the signing oracle.

Another security notion that should also be satisfied by an ABS scheme, even more so with delegation, is that a signature generated for a given policy should be independent of the user, and signatures generated by fresh keys or delegated keys should be indistinguishable. We refer to this property as anonymity, as in [MPR11]. Our definition requires to examine six different distributions, to take into account the possibility of delegation:

Definition 4 (Anonymity). *An ABS with delegation scheme is said anonymous if, for any* $(\mathsf{PK}, \mathsf{MK}) \xleftarrow{\$} \mathsf{Setup}$*, any message m, any identities* $\mathsf{id}_0, \mathsf{id}_1$*, any attribute sets Γ_0, Γ_1, any signing keys* $\mathsf{SK}_0 \xleftarrow{\$} \mathsf{KeyGen}(\mathsf{MK}, \mathsf{id}_0, \Gamma_0)$, $\mathsf{SK}_1 \xleftarrow{\$} \mathsf{KeyGen}(\mathsf{MK}, \mathsf{id}_1, \Gamma_1)$*, any delegated keys* $\mathsf{SK}'_0 \xleftarrow{\$} \mathsf{Delegate\text{-}Attributes}(\mathsf{SK}_0, \mathsf{id}'_0, \Gamma'_0)$, $\mathsf{SK}'_1 \xleftarrow{\$} \mathsf{Delegate\text{-}Attributes}(\mathsf{SK}_1, \mathsf{id}'_1, \Gamma'_1)$*, for $\Gamma'_0 \subset \Gamma_0$ and $\Gamma'_1 \subset \Gamma_1$, any policy keys* $\tilde{\mathsf{SK}}'_0 \xleftarrow{\$} \mathsf{Delegate\text{-}Policy}(\mathsf{SK}_0, \mathcal{T})$, $\tilde{\mathsf{SK}}'_1 \xleftarrow{\$} \mathsf{Delegate\text{-}Policy}(\mathsf{SK}_1, \mathcal{T})$*, for any policy $\mathcal{T}$ satisfied by both Γ'_0 and Γ'_1, the six distributions of the signatures generated by* $\mathsf{Sig}(\mathsf{SK}_0, m, \mathcal{T})$, $\mathsf{Sig}(\mathsf{SK}'_0, m, \mathcal{T})$, $\mathsf{DelegateSig}(\tilde{\mathsf{SK}}'_0, m)$, $\mathsf{Sig}(\mathsf{SK}_1, m, \mathcal{T})$, $\mathsf{Sig}(\mathsf{SK}'_1, m, \mathcal{T})$, $\mathsf{DelegateSig}(\tilde{\mathsf{SK}}'_1, m)$ *are indistinguishable.*

Indistinguishability can be perfect, statistical or computational, which leads to perfect, statistical or computational anonymity.

Whereas perfect anonymity excludes traceability, computational anonymity may allow the existence of a trapdoor leading to traceability. We will propose both in the following.

2.5 Policies and Access-Trees

We use the same approach as [GPSW06] by defining a policy on attributes in $\mathcal{U}$: we will consider a policy as an access-tree $\mathcal{T}$ 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). Nevertheless, access-trees with only AND and OR gates are as expressive as access-trees with threshold gates.

Access-trees have a similar structure to boolean expressions, which are commonly used in applicative security. This makes the access-tree approach easily compatible with existing security infrastructures that already rely on such expressions for their policy.

Definition of access-trees. We only recall the important notations of access-trees, and refer the reader to [DGP22] for a full definition of access-trees. We also introduce the additional notion of dual trees, that will be used to prove the correctness and anonymity our signatures.

An access-tree $\mathcal{T}$ is a rooted labeled tree from the root ρ, with internal nodes associated to AND and OR gates and leaves associated to attributes. 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 $\mathsf{children}(\nu)$.

Satisfying an access-tree. 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. We then denote $\mathcal{T}(\Gamma) = 1$ when the access-tree $\mathcal{T}$ is satisfied by the set of attributes Γ.

Evaluation Pruned Trees. We consider an access-tree $\mathcal{T}$ with leaves $\mathcal{L}$ and a set Γ of attributes so that $\mathcal{T}(\Gamma) = 1$. A Γ-evaluation tree $\mathcal{T}' \subset \mathcal{T}$ is a pruned version of $\mathcal{T}$, where one children only is kept for 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. $\mathsf{EPT}(\mathcal{T}, \Gamma)$ is non-empty if and only if $\mathcal{T}(\Gamma) = 1$.

Labelings of a Tree. We define the labeling of a tree, which can be seen as a linear secret sharing among the leaves of the tree.

Definition 5 (Random y-Labeling). *A random y-labeling Λ_y of an access-tree $\mathcal{T}$, for any $y \in \mathbb{Z}_p$, is the probabilistic algorithm $\Lambda_y(\mathcal{T})$ that sets $a_\rho \leftarrow y$ for the root, and then in a top-down manner starting from the root, set a_ν for each internal node ν : if ν is an AND-node 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 ν is an OR-gate, each children is associated to the value a_ν.*

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 $\mathcal{T}$. Random labelings have several properties: the sum of a y-labeling and a random z-labeling is a random $(y+z)$-labeling of $\mathcal{T}$. And multiplying all the labels of a y-labeling by a constant c leads to a cy-labeling. Furthermore, because of the recursive definition of labelings, one can see that, given a labeling (a_λ) on $\mathcal{T}$, we can extract a a_ν-labeling for any subtree of $\mathcal{T}$, rooted at node ν, which coincides with $\mathcal{T}$ on all values a_λ for leaves of the subtree.

Evaluation of a Labeled Tree. As noted above, labels on leaves are a linear secret sharing of the root that allows reconstruction of the secret if and only if the policy is satisfied: for a set Γ that satisfies $\mathcal{T}$ 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.

Dual-Trees. For our construction, we will use another type of tree, called the *dual-tree* $\mathcal{T}^*$ of $\mathcal{T}$: this is the exact same tree as $\mathcal{T}$, except that all OR gates in $\mathcal{T}$ become AND gates in $\mathcal{T}^*$, and conversely all AND gates in $\mathcal{T}$ become OR gates in $\mathcal{T}^*$. We note that the structure of $\mathcal{T}$ and $\mathcal{T}^*$ is identical, in particular all leaves are present on both trees, thus we will abuse notations and consider $\mathcal{L}_\mathcal{T} = \mathcal{L}_{\mathcal{T}^*}$ when there is no ambiguity.

Dual-trees will be crucial in our constructions. They will allow the signer to share enough information to the verifier for the verification of the signature, to prove correctness. At the same time, it prevents revealing anything about the validity of the access-tree other than the signer could sign it with its attributes, to ensure anonymity. This is formalized in the two next propositions, proven in Appendix C.1.

Proposition 6. *If $(a_\lambda)_\lambda$ is an a_0-labeling of $\mathcal{T}$, and $(b_\lambda)_\lambda$ is a b_0-labeling of its dual tree $\mathcal{T}^*$, then $\sum_{\lambda \in \mathcal{L}} a_\lambda b_\lambda = a_0 b_0$.*

This stems from the fact that there is always an OR-gate (from either $\mathcal{T}$ or $\mathcal{T}'$) which creates a common factor when recursively evaluating the product at each node on both trees (Illustration on Figure 3).

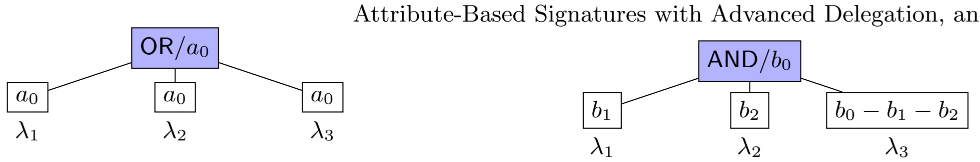

Fig. 3. A trivial access-tree with an OR-gate (left) and its dual tree with an AND-gate (right). Each tree has a random labeling for a_0 or b_0. One can see that $\sum_{\lambda \in \mathcal{L}} a_\lambda b_\lambda = a_0 b_1 + a_0 b_2 + a_0(b_0 - b_1 - b_2) = a_0 b_0$. This generalizes to any number of children for a gate, and to any access-tree by recursion from bottom to top.

Proposition 7. *Let $\mathcal{T}$ be an access-tree and Γ a set of attributes so that $\mathcal{T}(\Gamma) = 1$. Then, for any Evaluation Pruned Tree $\mathcal{T}' \in \mathsf{EPT}(\mathcal{T}, \Gamma)$, there is a 1-labeling $(b_\lambda)_\lambda$ of the dual $\mathcal{T}^*$ which verifies: $b_\lambda = 1$ for all $\lambda \in \mathcal{L}_{\mathcal{T}'}$ and $b_\lambda = 0$ for all $\lambda \notin \mathcal{L}_{\mathcal{T}'}$.*

If a tree $\mathcal{T}$ is satisfied by a set of attributes, we can associate the value 1 to the leaves of the satisfied attributes in the tree (which defines the pruned tree $\mathcal{T}' \subset \mathcal{T}$), and this association is effectively a 1-labeling of the dual-tree $\mathcal{T}'^* \subset \mathcal{T}^*$, but also a 1-labeling of the dual-tree $\mathcal{T}^*$. As this is a 1-labeling of $\mathcal{T}^*$, but specific to $\mathcal{T}'$, we can then randomize it for anonymity, with a 0-labeling of $\mathcal{T}^*$, through the linearity of labelings on $\mathcal{T}^*$, while maintaining correctness (Illustration on Figure 4).

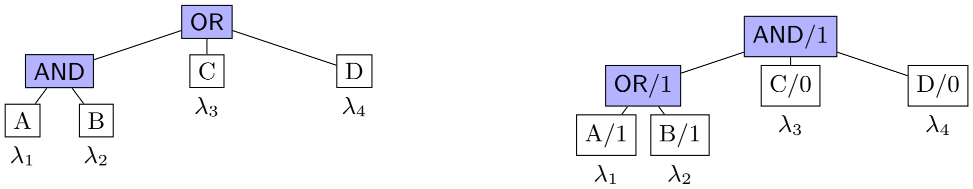

Fig. 4. An access-tree fulfilled by the set $\{A, B\}$ (left). One can extract a 1-labeling from its dual-tree (right) which has values 1 on leaves $\{A, B\}$ and 0 on all other leaves.

3 ABS with Attribute and Policy Delegation

In this section, we describe a Delegatable ABS scheme with perfect anonymity, and an unbounded universe of attributes. The basic idea is to derive the scheme from a KP-ABE, where signatures can be seen as a decryption key associated to a policy, and the verification algorithm tries to decrypt a ciphertext on a set of attributes. If decryption works, the signature is valid, otherwise the signature is invalid. To enable delegations of attributes and policies, we separate the commitment to the message and the policy in the signing process, which is a critical difference from the [OT13] construction.

3.1 Description of our ABS Scheme

In our ABS with delegation, users can delegate subset of attributes with Delegate-Attributes, or pre-signed policies with Delegate-Policy. Then, they can sign using Sig with keys generated from either KeyGen or Delegate-Attributes, using any attributes they want. Alternatively, they can sign using DelegateSig with keys generated from Delegate-Policy, on a pre-chosen access-tree included in the keys.

To simplify reading, we also detail the distribution of vectors of each element, which will be enough to make the security proof afterwards. 0^k denotes k zero components in a vector.

Setup(1^κ)**.** The algorithm chooses three random dual orthogonal bases, in a pairing-friendly setting $\mathcal{PG} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, e, G_1, G_2, q)$:

$$\mathbb{B} = (\mathbf{b}_1, \dots, \mathbf{b}_4) \qquad \mathbb{D} = (\mathbf{d}_1, \dots, \mathbf{d}_{10}) \qquad \mathbb{H} = (\mathbf{h}_1, \dots, \mathbf{h}_8)$$
$$\mathbb{B}^* = (\mathbf{b}_1^*, \dots, \mathbf{b}_4^*) \qquad \mathbb{D}^* = (\mathbf{d}_1^*, \dots, \mathbf{d}_{10}^*) \qquad \mathbb{H}^* = (\mathbf{h}_1^*, \dots, \mathbf{h}_8^*).$$

It picks two full-domain hash functions $\mathcal{H}$ and $\mathcal{H}'$ onto $\mathbb{Z}_q$. It sets the public parameters $\mathsf{PK} = \{\mathcal{PG}, \mathcal{H}, \mathcal{H}', (\mathbf{b}_1, \mathbf{b}_3), (\mathbf{b}_2^*), (\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_5), (\mathbf{d}_1^*, \mathbf{d}_2^*, \mathbf{d}_3^*, \mathbf{d}_4^*), (\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3, \mathbf{h}_5), (\mathbf{h}_4^*)\}$, and master secret key $\mathsf{MK} = \{(\mathbf{b}_1^*), (\mathbf{h}_1^*, \mathbf{h}_2^*, \mathbf{h}_3^*)\}$.

$\mathsf{KeyGen}(\mathsf{MK},\mathsf{id},\varGamma)$**.** A random scalar $\delta_{\mathsf{id}} \xleftarrow{\$} \mathbb{Z}_q^*$ is associated to id, to define

$$\mathbf{k}_0^* = (\delta_{\mathsf{id}}, \phi_0, 0^2)_{\mathbb{B}^*} \qquad \mathbf{k}_t^* = (\delta_{\mathsf{id}}, \pi_t(1,t), \phi_t, 0^6)_{\mathbb{D}^*}$$
$$\mathbf{r}_1^* = (\delta_{\mathsf{id}}, 0, 0, \psi_1, 0^4)_{\mathbb{H}^*} \qquad \mathbf{r}_2^* = (0, \delta_{\mathsf{id}}, 0, \psi_2, 0^4)_{\mathbb{H}^*} \quad \mathbf{r}_3^* = (0, 0, \delta_{\mathsf{id}}, \psi_3, 0^4)_{\mathbb{H}^*}$$

for all attributes $t \in \varGamma$, with $\phi_0, \psi_1, \psi_2, \psi_3, (\phi_t)_t, (\pi_t)_t \xleftarrow{\$} \mathbb{Z}_q^*$ for each t. The signing key $\mathsf{SK}_{\mathsf{id},\varGamma}$ is then set $(\mathbf{k}_0^*, (\mathbf{k}_t^*)_{t\in\varGamma}, \mathbf{r}_1^*, \mathbf{r}_2^*, \mathbf{r}_3^*)$. It can be computed later for new attributes t, for id, with extra $\mathbf{k}_t^*$, using the same δ_{id}, specific to user id.

$\mathsf{Delegate\text{-}Attributes}(\mathsf{SK}_{\mathsf{id},\varGamma}, \bar{\mathsf{id}}, \varGamma')$**.** Pick random $\alpha_{\bar{\mathsf{id}}}, \phi_0', (\phi_t')_t, \psi_1', \psi_2', \psi_3' \xleftarrow{\$} \mathbb{Z}_q$ for $t \in \varGamma' \subset \varGamma$, and compute

$$\bar{\mathbf{k}}_0^* = \alpha_{\bar{\mathsf{id}}} \cdot \mathbf{k}_0^* + \phi_0' \cdot \mathbf{b}_2^* \qquad \bar{\mathbf{k}}_t^* = \alpha_{\bar{\mathsf{id}}} \cdot \mathbf{k}_t^* + \phi_t' \cdot \mathbf{d}_4^*$$
$$\bar{\mathbf{r}}_1^* = \alpha_{\bar{\mathsf{id}}} \cdot \mathbf{r}_1^* + \psi_1' \cdot \mathbf{h}_4^* \qquad \bar{\mathbf{r}}_2^* = \alpha_{\bar{\mathsf{id}}} \cdot \mathbf{r}_2^* + \psi_2' \cdot \mathbf{h}_4^* \qquad \bar{\mathbf{r}}_3^* = \alpha_{\bar{\mathsf{id}}} \cdot \mathbf{r}_3^* + \psi_3' \cdot \mathbf{h}_4^*$$

The delegated signing key $\mathsf{SK}_{\bar{\mathsf{id}},\varGamma'}$ is then set as $(\bar{\mathbf{k}}_0^*, (\bar{\mathbf{k}}_t^*)_{t\in\varGamma'}, \bar{\mathbf{r}}_1^*, \bar{\mathbf{r}}_2^*, \bar{\mathbf{r}}_3^*)$. More delegations can be provided with additional $\bar{\mathbf{k}}_t^*$ for $\bar{\mathsf{id}}$ from id using the same $\alpha_{\bar{\mathsf{id}}}$, specific to the attribute-delegation from id to $\bar{\mathsf{id}}$.

This results in the vectors, with $\alpha_{\bar{\mathsf{id}}}, \phi_0, \phi_0', (\phi_t)_t, (\phi_t')_t, (\pi_t)_t, \psi_1, \psi_1', \psi_2, \psi_2', \psi_3, \psi_3' \xleftarrow{\$} \mathbb{Z}_q$

$$\bar{\mathbf{k}}_0^* = (\alpha_{\bar{\mathsf{id}}}\delta_{\mathsf{id}}, \alpha_{\bar{\mathsf{id}}}\phi_0 + \phi_0', 0^2)_{\mathbb{B}^*} \qquad \bar{\mathbf{k}}_t^* = (\alpha_{\bar{\mathsf{id}}}\delta_{\mathsf{id}}, \alpha_{\bar{\mathsf{id}}}\pi_t(1,t), \alpha_{\bar{\mathsf{id}}}\phi_t + \phi_t', 0^6)_{\mathbb{D}^*}$$
$$\bar{\mathbf{r}}_1^* = (\alpha_{\bar{\mathsf{id}}}\delta_{\mathsf{id}}, 0, 0, \alpha_{\bar{\mathsf{id}}}\psi_1 + \psi_1', 0^4)_{\mathbb{H}^*}$$
$$\bar{\mathbf{r}}_2^* = (0, \alpha_{\bar{\mathsf{id}}}\delta_{\mathsf{id}}, 0, \alpha_{\bar{\mathsf{id}}}\psi_2 + \psi_2', 0^4)_{\mathbb{H}^*} \qquad \bar{\mathbf{r}}_3^* = (0, 0, \alpha_{\bar{\mathsf{id}}}\delta_{\mathsf{id}}, \alpha_{\bar{\mathsf{id}}}\psi_3 + \psi_3', 0^4)_{\mathbb{H}^*}$$

which follow the same distributions as, with $\delta_{\bar{\mathsf{id}}}, (\pi_t)_t, \phi_0, (\phi_t)_t, \psi_1, \psi_2, \psi_3 \xleftarrow{\$} \mathbb{Z}_q^*$

$$\bar{\mathbf{k}}_0^* = (\delta_{\bar{\mathsf{id}}}, \phi_0, 0^2)_{\mathbb{B}^*} \qquad \bar{\mathbf{k}}_t^* = (\delta_{\bar{\mathsf{id}}}, \pi_t(1,t), \phi_t, 0^6)_{\mathbb{D}^*} \qquad \forall t \in \varGamma$$
$$\bar{\mathbf{r}}_1^* = (\delta_{\bar{\mathsf{id}}}, 0, 0, \psi_1, 0^4)_{\mathbb{H}^*}$$
$$\bar{\mathbf{r}}_2^* = (0, \delta_{\bar{\mathsf{id}}}, 0, \psi_2, 0^4)_{\mathbb{H}^*} \qquad \bar{\mathbf{r}}_3^* = (0, 0, \delta_{\bar{\mathsf{id}}}, \psi_3, 0^4)_{\mathbb{H}^*}$$

$\mathsf{Sig}(\mathsf{SK}_{\mathsf{id},\varGamma}, m, \mathcal{T})$**.** Let $\mathcal{T}' \in \mathsf{EPT}(\mathcal{T}, \varGamma)$ be an Evaluation Pruned Tree, scalars $\nu, \xi, \zeta, (\omega_\lambda)_\lambda, (q_\lambda)_\lambda \xleftarrow{\$} \mathbb{Z}_q^*$, and $(\alpha_\lambda)_\lambda$ the 1-labeling of the dual $\mathcal{T}^*$ specific to $\mathcal{T}'$ (see Proposition 7), where $\alpha_\lambda = 1$ if $\lambda \in \mathcal{L}_{\mathcal{T}'}$, else $\alpha_\lambda = 0$. This is possible as $\mathcal{T}(\varGamma) = 1$. Compute $(\beta_\lambda)_\lambda$ to be a random 0-labeling of the dual $\mathcal{T}^*$ associated to $\mathcal{T}$.

Eventually, set, for $H = \mathcal{H}(\mathcal{T}), H' = \mathcal{H}'(m)$:

$$U^* = \xi\mathbf{k}_0^* + \zeta\mathbf{b}_2^* \qquad S_\lambda^* = \alpha_\lambda\xi \cdot \mathbf{k}_{t_\lambda}^* + \beta_\lambda \cdot \mathbf{d}_1^* + \omega_\lambda \cdot (\mathbf{d}_2^* + t_\lambda \cdot \mathbf{d}_3^*) + q_\lambda \cdot \mathbf{d}_4^*$$
$$V^* = \xi(\mathbf{r}_1^* + H \cdot \mathbf{r}_2^* + H' \cdot \mathbf{r}_3^*) + \nu \cdot \mathbf{h}_4^*$$

for all the leaves λ, where t_λ is the associated attribute of λ. The signature is thus $\sigma = (U^*, V^*, (S_\lambda^*)_\lambda)$.

This results in the vectors, with $\delta_{\mathsf{id}}, \nu, \xi, \zeta, (\omega_\lambda)_\lambda, (q_\lambda)_\lambda, \phi_0, \psi_1, \psi_2, \psi_3, (\phi_{t_\lambda})_\lambda, (\pi_{t_\lambda})_{t_\lambda} \xleftarrow{\$} \mathbb{Z}_q^*$

$$U^* = (\xi\delta_{\mathsf{id}}, \xi\phi_0 + \zeta, 0^2)_{\mathbb{B}^*}$$
$$S_\lambda^* = (\alpha_\lambda\xi\delta_{\mathsf{id}} + \beta_\lambda, (\alpha_\lambda\xi\pi_{t_\lambda} + \omega_\lambda)(1, t_\lambda), \alpha_\lambda\xi\phi_{t_\lambda} + q_\lambda, 0^6)_{\mathbb{D}^*}$$
$$V^* = (\xi\delta_{\mathsf{id}} \cdot (1, H, H'), \xi(\psi_1 + \psi_2 H + \psi_3 H') + \nu, 0^4)_{\mathbb{H}^*}$$

which follow the same distributions as, with $\delta, \nu, \zeta, (\omega_\lambda)_\lambda, (q_\lambda)_\lambda \xleftarrow{\$} \mathbb{Z}_q^*$, and $(\beta_\lambda)_\lambda$ a random 0-labeling of $\mathcal{T}^*$,

$$U^* = (\delta, \zeta, 0^2)_{\mathbb{B}^*} \qquad S_\lambda^* = (\alpha_\lambda\delta + \beta_\lambda, \omega_\lambda(1, t_\lambda), q_\lambda, 0^6)_{\mathbb{D}^*}$$
$$V^* = (\delta \cdot (1, H, H'), \nu, 0^4)_{\mathbb{H}^*}$$

$\mathsf{Delegate\text{-}Policy}(\mathsf{SK}_{\mathsf{id},\mathit{\Gamma}}, \bar{\mathsf{id}}, \mathcal{T})$**.** Let $\mathcal{T}' \in \mathsf{EPT}(\mathcal{T}, \mathit{\Gamma})$ be an Evaluation Pruned Tree, scalars $\nu, \xi, \zeta, \psi_3', (\omega_\lambda)_\lambda, (q_\lambda)_\lambda$ $\mathbb{Z}_q^*$, and $(\alpha_\lambda)_\lambda$ the 1-labeling of $\mathcal{T}^*$ specific to $\mathcal{T}'$ (see Proposition 7), where $\alpha_\lambda = 1$ if $\lambda \in \mathcal{L}_{\mathcal{T}'}$, else $\alpha_\lambda = 0$. This is possible as $\mathcal{T}(\mathit{\Gamma}) = 1$. Then, compute $(\beta_\lambda)_\lambda$ to be a random 0-labeling of the dual $\mathcal{T}^*$ associated to $\mathcal{T}$.

Eventually, set for $H = \mathcal{H}(\mathcal{T})$:

$$U^* = \xi \mathbf{k}_0^* + \zeta \mathbf{b}_2^* \qquad S_\lambda^* = \alpha_\lambda \xi \cdot \mathbf{k}_{t_\lambda}^* + \beta_\lambda \mathbf{d}_1^* + \omega_\lambda(\mathbf{d}_2^* + t_\lambda \cdot \mathbf{d}_3^*) + q_\lambda \cdot \mathbf{d}_4^*$$

$$\mathbf{r}_3'^* = \xi \mathbf{r}_3^* + \psi_3' \mathbf{h}_4^* \qquad V^* = \xi(\mathbf{r}_1^* + H \cdot \mathbf{r}_2^*) + \nu \cdot \mathbf{h}_4^*$$

for all the leaves λ, where t_λ is the associated attribute of λ. The delegated key is thus $\mathsf{SK}_{\bar{\mathsf{id}},\mathcal{T}} = (U^*, V^*, \mathbf{r}_3'^*, (S_\lambda^*)_\lambda)$.

This results in the vectors, with $\delta_{\mathsf{id}}, \nu, \xi, \zeta, (\omega_\lambda)_\lambda, (q_\lambda)_\lambda, \phi_0, \psi_1, \psi_2, (\phi_{t_\lambda})_\lambda$, $(\pi_{t_\lambda})_{t_\lambda} \xleftarrow{\$} \mathbb{Z}_q^*$

$$\begin{aligned} U^* &= (\xi\delta_{\mathsf{id}}, \xi\phi_0 + \zeta, 0^2)_{\mathbb{B}^*} \\ S_\lambda^* &= (\alpha_\lambda\xi\delta_{\mathsf{id}} + \beta_\lambda, (\alpha_\lambda\xi\pi_{t_\lambda} + \omega_\lambda)(1, t_\lambda), \alpha_\lambda\xi\phi_{t_\lambda} + q_\lambda, 0^6)_{\mathbb{D}^*} \\ V^* &= (\xi\delta_{\mathsf{id}} \cdot (1, H, 0), \xi(\psi_1 + \psi_2 H) + \nu, 0^4)_{\mathbb{H}^*} \\ \mathbf{r}_3'^* &= (0, 0, \xi\delta_{\mathsf{id}}, \xi\psi_3 + \psi_3', 0^4)_{\mathbb{H}^*} \end{aligned}$$

which follow the same distributions as, with $\delta_{\mathsf{id}}, \nu, \zeta, (\omega_\lambda)_\lambda, (q_\lambda)_\lambda, \psi_3' \xleftarrow{\$} \mathbb{Z}_q^*$, and $(\beta_\lambda)_\lambda$ a random 0-labeling of $\mathcal{T}^*$

$$U_i^* = (\delta_{\mathsf{id}}, \zeta, 0^2)_{\mathbb{B}^*} \qquad S_\lambda^* = (\alpha_\lambda\delta_{\mathsf{id}} + \beta_\lambda, \omega_\lambda(1, t_\lambda), q_\lambda, 0^6)_{\mathbb{D}^*}$$

$$V^* = (\delta_{\mathsf{id}} \cdot (1, H, 0), \nu, 0^4)_{\mathbb{H}^*} \qquad \mathbf{r}_3'^* = (\delta_{\mathsf{id}} \cdot (0, 0, 1), \psi_3', 0^4)_{\mathbb{H}^*}$$

$\mathsf{DelegateSig}(\mathsf{SK}_{\mathsf{id},\mathcal{T}}, m)$**.** Let $\mathcal{T}' \in \mathsf{EPT}(\mathcal{T}, \mathit{\Gamma})$ be an Evaluation Pruned Tree, $\nu, \xi, \zeta, (\omega_\lambda')_\lambda, (q_\lambda')_\lambda \xleftarrow{\$} \mathbb{Z}_q^*$, $(\beta_\lambda')_\lambda$ a random 0-labeling of $\mathcal{T}^*$. Set, for $H' = \mathcal{H}'(m)$:

$$U'^* = \xi U^* + \zeta \mathbf{b}_2^* \qquad S_\lambda'^* = \xi \cdot S_\lambda^* + \beta_\lambda' \mathbf{d}_1^* + \omega_\lambda'(\mathbf{d}_2^* + t_\lambda \cdot \mathbf{d}_3^*) + q_\lambda' \cdot \mathbf{d}_4^*$$

$$V'^* = \xi(V^* + H' \cdot \mathbf{r}_3'^*) + \nu \cdot \mathbf{h}_4^*$$

for all the leaves λ, where t_λ is the associated attribute of λ. The signature is thus $\sigma = (U'^*, V'^*, (S_\lambda'^*)_\lambda)$.

This results in the vectors, with $\delta_{\mathsf{id}}, \nu, \nu', \xi, \zeta, \zeta', (\omega_\lambda)_\lambda, (\omega_\lambda')_\lambda, (q_\lambda)_\lambda, (q_\lambda')_\lambda \xleftarrow{\$} \mathbb{Z}_q^*$, and $(\alpha_\lambda)_\lambda$ the 1-labeling of $\mathcal{T}^*$,

$$\begin{aligned} U'^* &= (\xi\delta_{\mathsf{id}}, \xi\zeta + \zeta', 0^2)_{\mathbb{B}^*} \\ S_\lambda'^* &= (\xi(\alpha_\lambda\delta_{\mathsf{id}} + \beta_\lambda) + \beta_\lambda', (\alpha_\lambda\xi\omega_\lambda + \omega_\lambda')(1, t_\lambda), \alpha_\lambda\xi q_\lambda + q_\lambda', 0^6)_{\mathbb{D}^*} \\ V'^* &= (\xi\delta_{\mathsf{id}} \cdot (1, H, H'), \xi\nu + \nu', 0^4)_{\mathbb{H}^*} \end{aligned}$$

which follow the same distributions as, with $\delta_{\mathsf{id}}, \nu, \zeta, (\omega_\lambda)_\lambda, (q_\lambda)_\lambda \xleftarrow{\$} \mathbb{Z}_q^*$, $(\beta_\lambda)_\lambda$ a random 0-labeling of $\mathcal{T}^*$, and $H = \mathcal{H}(\mathcal{T})$

$$U'^* = (\delta_{\mathsf{id}}, \zeta, 0^2)_{\mathbb{B}^*} \qquad S_\lambda'^* = (\alpha_\lambda\delta_{\mathsf{id}} + \beta_\lambda, \omega_\lambda(1, t_\lambda), q_\lambda, 0^6)_{\mathbb{D}^*}$$

$$V'^* = (\delta_{\mathsf{id}} \cdot (1, H, H'), \nu, 0^4)_{\mathbb{H}^*}$$

$\mathsf{Verif}(\mathsf{PK}, m, \mathcal{T}, \sigma)$**.** Let $\kappa, \kappa_0, (\kappa_\lambda)_\lambda, s, s_0, \theta, \theta', (\theta_\lambda)_\lambda \xleftarrow{\$} \mathbb{Z}_q$. Let $(s_\lambda)_\lambda$ be a random s_0-labeling of $\mathcal{T}$, then set, for $\bar{H} = \mathcal{H}(\mathcal{T}), \bar{H}' = \mathcal{H}'(m)$:

$$u = (-s_0 - s, 0, \kappa_0, 0)_{\mathbb{B}} \qquad c_\lambda = (s_\lambda, \theta_\lambda(t_\lambda, -1), 0, \kappa_\lambda, 0^5)_{\mathbb{D}}$$

$$v = (s + \theta\bar{H} + \theta'\bar{H}', -\theta, -\theta', 0, \kappa, 0^3)_{\mathbb{H}}$$

If $e(\mathbf{b}_1, U^*) \neq 1_{\mathbb{G}_t} \wedge e(u, U^*) \cdot e(v, V^*) \cdot \prod e(c_\lambda, S_\lambda^*) = 1_{\mathbb{G}_t}$, accept and output 1, else reject and output 0.

One can note that, as usual with Dual Pairing Vectors Spaces, some basis vectors are kept hidden to real-life players, as they will only be used in the security proofs: $(\mathbf{b}_2, \mathbf{b}_4)$, $(\mathbf{b}_3^*, \mathbf{b}_4^*)$, $(\mathbf{d}_4, \mathbf{d}_6, \mathbf{d}_7, \mathbf{d}_8, \mathbf{d}_9, \mathbf{d}_{10})$, $(\mathbf{d}_5^*, \mathbf{d}_6^*, \mathbf{d}_7^*, \mathbf{d}_8^*, \mathbf{d}_9^*, \mathbf{d}_{10}^*)$, $(\mathbf{h}_4, \mathbf{h}_6, \mathbf{h}_7, \mathbf{h}_8)$, and $(\mathbf{h}_5^*, \mathbf{h}_6^*, \mathbf{h}_7^*, \mathbf{h}_8^*)$.

Correctness. Let $(U^*, V^*, (S_\lambda^*)_\lambda)$ be a signature generated by Sig with a key $\mathsf{SK}_{\mathsf{id},\Gamma}$, or by $\mathsf{DelegateSig}$ with a key $\mathsf{SK}_{\mathsf{id},\mathcal{T}}$, for an access-tree $\mathcal{T}$ and attributes Γ so that $\mathcal{T}(\Gamma) = 1$. Let $(u, v, (c_\lambda)_\lambda)$ the verification vectors generated by Verif for the same access-tree. We note $\mathcal{T}' \in \mathsf{EPT}(\mathcal{T}, \Gamma)$ the Evaluation Pruned Tree used during signature.

We remind from Proposition 6 that $\sum_{\lambda \in \mathcal{L}} s_\lambda \alpha_\lambda = s_0$ and $\sum_{\lambda \in \mathcal{L}} s_\lambda \beta_\lambda = 0$, as we have labelings of $\mathcal{T}$ and $\mathcal{T}^*$. We have $\sum_{\lambda \in \mathcal{L}} s_\lambda (\alpha_\lambda \delta_{\mathsf{id}} + \beta_\lambda) = s_0(\delta_{\mathsf{id}} + 0) = s_0 \delta_{\mathsf{id}}$. The first check $e(\mathbf{b}_1, U^*) = g_t^{\delta_{\mathsf{id}}} \neq 1_{\mathbb{G}_t}$ is to make sure $\delta_{\mathsf{id}} \neq 0$, and thus that $\xi \neq 0$ during the signing process. For the second verification:

$$\prod e(c_\lambda, S_\lambda^*) = g_t^{\sum_{\lambda \in \mathcal{L}} s_\lambda (\alpha_\lambda \delta_{\mathsf{id}} + \beta_\lambda)} = g_t^{s_0 \delta_{\mathsf{id}}} \tag{1}$$

$$e(u, U^*) \cdot e(v, V^*) = g_t^{\delta_{\mathsf{id}} \cdot (-s_0 - s)} \cdot g_t^{\delta_{\mathsf{id}} s} = g_t^{-\delta_{\mathsf{id}} s_0} \tag{2}$$

This leads to an accept if the signature was properly generated, with the same t_λ in the S_λ^*'s and c_λ's, such that the vectors $(1, t_\lambda)$ and $(t_\lambda, -1)$ are orthogonal in equation (1), and $(H, H') = (\bar{H}, \bar{H})$ so that $(1, H)$ and $(\bar{H}, -1)$, as well as $(1, H')$ and $(\bar{H}', -1)$ are orthogonal, to guarantee that for random θ and θ', $(1, H, H')$ and $(\theta\bar{H} + \theta'\bar{H}', -\theta, -\theta')$ are orthogonal in equation (2).

Issuing new attributes to an existing key. We propose a way for the Central Authority, as well as any delegator, to compute new attributes for an existing signing key without having to recompute the full key.

The main insight of our technique is that all keys possess a random δ_{id} (which can be of the form $\alpha_{\mathsf{id}}\delta_{\mathsf{id}}$ after any number of delegation), specific to the current id considered, that binds every part of the key together. Thus, we use a Pseudo-Random Function on the entry id to generate these δ_{id}.

The key of the PRF used by each agent (the Central Authority, and any delegator) shall be unique and secret, as the knowledge of δ_{id} is enough to generate new attributes arbitrarily for any existing signing key.

3.2 Security Results

About the above ABS with delegation, one can claim the unforgeability and the perfect anonymity, as defined in Section 2.4.

Theorem 8 (Existential Unforgeability). *The* ABS *scheme with delegation described in Section 3.1 is existentially unforgeable under the collision-resistance of the hash functions* $\mathcal{H}, \mathcal{H}'$ *and the* SXDH *assumption.*

Theorem 9 (Perfect Anonymity). *The* ABS *scheme with delegation described in Section 3.1 is perfectly anonymous.*

4 Sketches of the Security Proofs

We provide here some sketches of the proofs. Full proofs are detailed in Appendix C.

4.1 Perfect Anonymity

Proof. Let us define an alternative signing algorithm AltSig, that uses the master secret key instead of an individual signing key. We will first show that this alternative signature algorithm produces signatures indistinguishable from the ones created by an ABS without delegation. Then, we show that the distribution of signatures generated by delegated keys, whether via delegation of attributes or delegation of policies, is the same as the distribution of signatures made by keys in an ABS without delegation. This will show that the distribution of all the possible signatures made with our scheme is the same.

Let us begin with AltSig:

$\mathsf{AltSig}(\mathsf{MK}, m, \mathcal{T})$**.** With random scalars $\delta', \zeta, \nu, (q_\lambda)_\lambda, (\gamma_\lambda)_\lambda \xleftarrow{\$} \mathbb{Z}_q$, and $(\beta'_\lambda)_\lambda$ a random δ'-labeling of $\mathcal{T}^*$, set, for $H = \mathcal{H}(\mathcal{T})$ and $H' = \mathcal{H}'(m)$:

$$U^* = (\delta', \zeta, 0^2)_{\mathbb{B}^*} \qquad S^*_\lambda = (\beta'_\lambda, \gamma_\lambda(1, t_\lambda), q_\lambda, 0^6)_{\mathbb{D}^*} \qquad V^* = (\delta' \cdot (1, H, H'), \nu, 0^4)_{\mathbb{H}^*}$$

We claim that this is the same distribution as a real signature generated by a signing key $\mathsf{SK}_{\mathsf{id},\Gamma}$ from the KeyGen algorithm (which is detailed in the supplementary materials in $\mathbf{G}_0$), except for two elements that we now discuss.

First, the random $(\gamma_\lambda)_\lambda$ from the second component of S^*_λ follows the same random uniform distribution as $(\alpha_\lambda \xi \pi_\lambda + \omega_\lambda)_\lambda$.

Second, the random δ'-labeling $(\beta'_\lambda)_\lambda$ of $\mathcal{T}^*$ replaces $(\alpha_\lambda \delta' + \beta_\lambda)_\lambda$, where (α_λ) is the 1-labeling of $\mathcal{T}^*$ specific to the Evaluation Pruned Tree associated to Γ, and (β_λ) a random 0-labeling of $\mathcal{T}^*$. As already noted, from the linearity of the labelings on $\mathcal{T}^*$, the linear combination is a random $1 \cdot \delta' + 0$-labeling of $\mathcal{T}^*$, as $(\beta'_\lambda)_\lambda$ is. We stress that adding a random 0-labeling of $\mathcal{T}^*$, which only depends on the policy $\mathcal{T}$, completely hides the initial labeling that was specific to Γ, and thus to the verifier.

Finally, we prove that the distribution of signatures made using delegation is the same as the one from AltSig. This is shown in $\mathbf{G}_0$, as we instantiate the first game for our EUF proof.

As an additional note for the incoming EUF proof, we note that, should it be necessary, AltSig would also be able to return the value $\mathbf{r}^*_3 = \delta' \cdot \mathbf{h}^*_3$, which will be useful to simulate answer to $\mathsf{ODelegatePolicy}$ queries, and output $\mathbf{r}^*_3$.

4.2 Existential Unforgeability

Proof. The proof is done in two parts. First, we show that if the adversary is only allowed access to oracles $\mathsf{OKeyGen}$ and OSig, then the scheme is EUF. This is effectively considering an adversary without delegation. Then, we show that we can simulate all the other oracles ($\mathsf{ODelegateAttributes}, \mathsf{ODelegatePolicy}, \mathsf{ODelegateSig}$) using only the $\mathsf{OKeyGen}$ and OSig oracles.

Proof without Delegation. For this proof, thanks to the (perfect) indistinguishability of the Sig and AltSig outputs from the Anonymity of the scheme, we will first replace the simulation of the signing oracle by the AltSig procedure.

Then, we will use the index id for all the KeyGen queries/answers, and we assume the number of KeyGen queries bounded by K. We use the index i for all the Sig queries/answers, and we assume the number of Sig queries bounded by S. We will also use t to denote the attributes, and we assume the number of attributes involved in a security game bounded by T.

The verification done by the challenger (on the candidate forgery output by the adversary) uses a pair $(m, \mathcal{T})$ that is different from any pair that appeared in the signing queries, hence with $\bar{H} = \mathcal{H}(\mathcal{T})$ and $\bar{H}' = \mathcal{H}'(m)$, but $(\bar{H}, \bar{H}') \neq (H_i, H'_i)$ for all i, under the collision-resistance of $\mathcal{H}$ and $\mathcal{H}'$, as for any pair $(m_i, \mathcal{T}_i)$ at least $m \neq m_i$ or $\mathcal{T} \neq \mathcal{T}_i$. Then, the proof follows the sequence of games presented on Figure 5, to show, as proven in Appendix C.3, that

$$\begin{aligned}\mathsf{Adv}_0 - \mathsf{Adv}_5 \leq\ & (6KT + 2S + 2) \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t) + (4T^2K + 6K + 4S) \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t) \\ & + S/q + \mathsf{Adv}^{\mathsf{coll}}_{\mathcal{H}}(t) + \mathsf{Adv}^{\mathsf{coll}}_{\mathcal{H}'}(t).\end{aligned}$$

We now deal with the final game $\mathbf{G}_5$ and consider a signature $(U^*, V^*, (S^*_\lambda)_\lambda)$ generated by the adversary. If $e(\mathbf{b}_1, U^*) = 1_{\mathbb{G}_t}$, then the verification fails, by definition of Verif. Hence, the first component of U^* must be non-zero, in the basis $\mathbb{B}^*$. We now consider the value $e(u, U^*) \cdot$

$\mathbf{G}_0$ Initialization of the EUF security game
For the (at most) K different id's and S different indices i's.

$$
\begin{array}{rcccccccccccl}
\hline
\mathbf{k}^*_{\mathsf{id},0} = (& \delta_{\mathsf{id}} & \phi_{\mathsf{id},0} & 0 & 0 &)_{\mathbb{B}^*} \\
U^*_i = (& \delta_i & \zeta_i & 0 & 0 &)_{\mathbb{B}^*} \\
u = (& -s_0 - s & 0 & \kappa_0 & 0 &)_{\mathbb{B}} \\
\hline
\mathbf{r}^*_{\mathsf{id},1} = (& \delta_{\mathsf{id}} & 0 & 0 & \psi_{\mathsf{id},1} & 0 & 0 & 0 & 0 &)_{\mathbb{H}^*} \\
\mathbf{r}^*_{\mathsf{id},2} = (& 0 & \delta_{\mathsf{id}} & 0 & \psi_{\mathsf{id},2} & 0 & 0 & 0 & 0 &)_{\mathbb{H}^*} \\
\mathbf{r}^*_{\mathsf{id},3} = (& 0 & 0 & \delta_{\mathsf{id}} & \psi_{\mathsf{id},3} & 0 & 0 & 0 & 0 &)_{\mathbb{H}^*} \\
V^*_i = (& \delta_i & \delta_i H_i & \delta_i H'_i & \nu_i & 0 & 0 & 0 & 0 &)_{\mathbb{H}^*} \\
v = (& s + \theta \bar{H} + \theta' \bar{H}' & -\theta & -\theta' & 0 & \kappa & 0 & 0 & 0 &)_{\mathbb{H}} \\
\hline
\mathbf{k}^*_{\mathsf{id},t} = (& \delta_{\mathsf{id}} & \pi_{\mathsf{id},t} & \pi_{\mathsf{id},t} t & \phi_{\mathsf{id},t} & 0 & 0 & 0 & 0 & 0 & 0 &)_{\mathbb{D}^*} \\
S^*_{i,\lambda} = (& \beta'_{i,\lambda} & \gamma_{i,\lambda} & \gamma_{i,\lambda} t_\lambda & q_{i,\lambda} & 0 & 0 & 0 & 0 & 0 & 0 &)_{\mathbb{D}^*} \\
c_\lambda = (& s_\lambda & \theta_\lambda t_\lambda, & -\theta_\lambda & 0 & \kappa_\lambda & 0 & 0 & 0 & 0 & 0 &)_{\mathbb{D}} \\
\hline
\end{array}
$$

$\mathbf{G}_1$ r_λ is a r_0-labeling, ω is random
$\mathsf{SubSpace\text{-}Ind}$ on $(\mathbb{B}, \mathbb{B}^*)_{3,4}$, $(\mathbb{D}, \mathbb{D}^*)_{5,6}$ and $(\mathbb{H}, \mathbb{H}^*)_{4,5}$

$$
\begin{array}{rcccccccccccl}
u = (& -s_0 - s & 0 & \kappa_0 & \boxed{-r_0} &)_{\mathbb{B}} \\
v = (& s + \theta \bar{H} + \theta' \bar{H}' & -\theta & -\theta' & 0 & \kappa & \boxed{\omega} & 0 & 0 &)_{\mathbb{H}} \\
c_\lambda = (& s_\lambda & \theta_\lambda t_\lambda & -\theta_\lambda & 0 & \kappa_\lambda & \boxed{r_\lambda} & 0 & 0 & 0 & 0 &)_{\mathbb{D}}
\end{array}
$$

$\mathbf{G}_2$ δ''_{id} all random: Hybrid sub-sequence (see Figure 8)

$$
\begin{array}{rcccccccccccl}
\mathbf{k}^*_{\mathsf{id},0} = (& \delta_{\mathsf{id}} & \phi_{\mathsf{id},0} & 0 & \boxed{\delta''_{\mathsf{id}}} &)_{\mathbb{B}^*} \\
\mathbf{k}^*_{\mathsf{id},t} = (& \delta_{\mathsf{id}} & \pi_{\mathsf{id},t} & \pi_{\mathsf{id},t} t & \phi_{\mathsf{id},t} & 0 & 0 & 0 & 0 & 0 & 0 &)_{\mathbb{D}^*}
\end{array}
$$

$\mathbf{G}_3$ r'_0 random: Formal change of basis on $(\mathbb{B}, \mathbb{B}^*)_4$

$$
\begin{array}{rcccccl}
u = (& -s_0 - s & 0 & \kappa_0 & \boxed{r'_0} &)_{\mathbb{B}}
\end{array}
$$

$\mathbf{G}_4$ ρ_i, τ_i all random: Hybrid sub-sequence (see Figure 9)

$$
\begin{array}{rcccccccccl}
U^*_i = (& \delta_i & \zeta_i & 0 & \boxed{\rho_i} &)_{\mathbb{B}^*} \\
V^*_i = (& \delta_i & \delta_i H_i & \delta_i H'_i & \nu_i & 0 & \boxed{\tau_i} & 0 & 0 &)_{\mathbb{H}}
\end{array}
$$

$\mathbf{G}_5$ s' random : Formal change of basis on $(\mathbb{B}, \mathbb{B}^*)_{4,1}$

$$
\begin{array}{rcccccl}
u = (& \boxed{s'} & 0 & \kappa_0 & r'_0 &)_{\mathbb{B}}
\end{array}
$$

Fig. 5. Sequence of Games for Unforgeability. Grey rectangles indicate the values changed in each game

$e(v, V^*) \cdot \prod e(c_\lambda, S^*_\lambda)$. Since the coefficient s' of $\mathbf{b}_1$ in u is uniform and independent from all other values, then $e(u, U^*)$ is uniform and independent from all other pairings in the Verif algorithm. This implies $e(u, U^*) \cdot e(v, V^*) \cdot \prod e(c_\lambda, S^*_\lambda) \neq 1_{\mathbb{G}_t}$ except with probability $1/q$: $\mathsf{Adv}_5 \leq 1/q$. As a consequence,

$$\begin{aligned}\mathsf{Adv}^{\mathsf{euf}} = \mathsf{Adv}_0 \leq (6KT + 2S + 2) \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t) + (4T^2K + 6K + 4S) \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t) \\ + (S + 1)/q + \mathsf{Adv}^{\mathsf{coll}}_{\mathcal{H}}(t) + \mathsf{Adv}^{\mathsf{coll}}_{\mathcal{H}'}(t).\end{aligned}$$

Proof with Delegation. We now reduce the EUF proof for our ABS with delegation to the proof without delegation. To do this, we simulate the oracles for delegated keys: keys from ODelegateAttributes can be simulated with OKeyGen, and signatures from ODelegateSig and policy keys from ODelegatePolicy can be simulated with AltSig.

Once we have shown these simulations, we are in a similar game as $\mathbf{G}_0$ for the Existential Unforgeability proof in the case without delegation. The sequence of games can continue the same way, with thus the same security bounds.

The full details of the proof for the simulation of the oracles can be found in Appendix C.2.

5 ABS with Traceability

In usual ABS definitions, and as shown in our construction, one usually expects perfect anonymity. But one could also require traceability of the signer, as in group signatures [Cv91], with an opener able to trace back the signer, and even prove the correct opening.

We propose such a construction, where tracers can be held accountable by a designated Tracing Authority. The traceability we propose does not ensure non-frameability from the Central Authority [EGK14], but it does ensure the traditional traceability property, which guarantees that malicious signers cannot deceive the designated tracer.

The way we design traceability make it compatible with our construction with delegation, from Section 3. This would result in a scheme where delegation of any signing key is possible, and where a Tracing Authority can trace any signature. However, we underline that a scheme that combines both our delegation and tracing can only trace back to the original delegator of the signing key that was used to produce the signature that is being tracked.

A more fine-grained solution for tracing delegated keys and path of delegations can be found in [GM19], but it comes at the cost of a scheme whose signatures' sizes are linear in both the number of attributes used and the length of the delegation path.

5.1 Traceable ABS

This extends the initial definition of an ABS, with the algorithms Setup (with additional tracing key TK and verification key VK), KeyGen, Sig, and Verif, we also consider the Trace and Judge algorithms:

$\mathsf{Trace}(\mathsf{TK}, m, \mathcal{T}, \sigma)$**.** Given the tracing key TK and a valid signature σ on $(m, \mathcal{T})$, the algorithm outputs the identity id of the signer together with a proof π, both set to $\perp$ in case of failure.

$\mathsf{Judge}(\mathsf{VK}, m, \sigma, \mathsf{id}, \pi)$**.** Given the verification key VK, a signature σ for a message m, and a proof π that user id generated $(m, \mathcal{T}, \sigma)$, the algorithm outputs 1 if π is valid or 0 else.

Correctness, unforgeability and anonymity are the same as for a regular ABS (see definitions from [OT11]), except that anonymity cannot be perfect, but computational. We also ask for any valid signature to be traced back to its signer, with a convincing proof π, either for a judge or anybody when VK is public.

Definition 10 (Traceability). *Traceability for ABS is defined by the following game between the adversary and a challenger:*

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

Oracles: *The following oracles can be called in any order and any number of times.*

$\mathsf{OKeyGen}(\mathsf{id}, \Gamma)$**:** *to model* KeyGen*-queries for any identity* id *and any set of attributes* Γ *of its choice, and the adversary gets back the key* $\mathsf{SK}_{\mathsf{id},\Gamma}$*;*

$\mathsf{OSig}(\mathsf{id}, m, \mathcal{T})$**:** *to model* Sig*-queries for any identity* id *and under any policy* $\mathcal{T}$ *of its choice for a message* m*, and the adversary gets the signature* σ*;*

Finalize(b')**:** *The adversary outputs a signature* $(m', \mathcal{T}', \sigma')$*. One asks* $(\mathsf{id}, \pi) = \mathsf{Trace}(\mathsf{TK}, \sigma')$*. If one of the following is true (non-legitimate attack)*

- $(m', \mathcal{T}')$ *has been queried to the* OSig*-oracle;*
- id *has been queried to the* $\mathsf{OKeyGen}$*-oracle with* Γ *such that* $\mathcal{T}'(\Gamma) = 1$ *and* $\mathsf{Judge}(\mathsf{VK}, m', \sigma', \mathsf{id}, \pi) = 1$*;*

then output 0, otherwise output $\mathsf{Verif}(\mathsf{PK}, m', \mathcal{T}', \sigma')$*.*

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

More precisely, we consider the adversary wins the traceability game if it manages to mislead the tracing procedure: by making it either fail or output an honest user (whose key has not been asked to the key-oracle), or by making the result of the tracing impossible to prove. Of course, we will ignore the output if it exactly corresponds to a signing-query.

We will add the restriction that the adversary can only corrupt disjoint sets of identities between any key-query and any other signing-query. We will call it the *distinct-user setting.*

One-Time Linearly-Homomorphic Signature We will rely on a (One-Time) Linearly-Homomorphic Signature ($\mathsf{OT\text{-}LH}$) [LPJY13]. An $\mathsf{OT\text{-}LH}$ scheme is a signature scheme where one can produce a valid signature out of any linear combination of valid signatures, provided he knows those other signatures. In other words, one can produce a valid signature σ of $\sum_i \alpha_i m_i$, provided he knows valid signatures σ_i of the messages m_i.

Another $\mathsf{OT\text{-}LH}$ scheme is proposed in [HP22] which has been proven in the generic group model, with an extractor that provides the coefficients in the linear combination of the initial messages for the new signed message. From this paper, we will also use the following theorem, that states the intractability of the Linear-Square problem:

Theorem 11 (Linear-Square Problem). *Given* n *Square Diffie-Hellman tuples* $(g_i, a_i = g_i^{w_i}, b_i = a_i^{w_i})$*, together with* w_i*, for random* $g_i \overset{\$}{\leftarrow} \mathbb{G}^*$ *and* $w_i \overset{\$}{\leftarrow} \mathbb{Z}_q^*$*, outputting* $(\alpha_i)_{i=1,\ldots,n}$ *such that* $(G = \prod g_i^{\alpha_i}, A = \prod a_i^{\alpha_i}, B = \prod b_i^{\alpha_i})$ *is a valid Square Diffie-Hellman, with at least two non-zero coefficients* α_i*, is computationally hard under the Discrete Logarithm assumption.*

5.2 Construction of Traceable ABS

We consider any $\mathsf{OT\text{-}LH}$ scheme $(\mathsf{KeyGen}', \mathsf{Sig}', \mathsf{DerivSign}', \mathsf{Verif}')$ in $\mathbb{G}_2^n$. We will also use a non-interactive zero-knowledge proof of knowledge $(\mathsf{NIZKPoK\text{-}SqDH}, \mathsf{VERIF\text{-}SqDH})$ of the witness w for a Square Diffie-Hellman tuple $(h_t, h_t^w, h_t^{w^2})$ in $\mathbb{G}_t$ and a non-interactive zero-knowledge proof $(\mathsf{NIZKPoK\text{-}DH}, \mathsf{VERIF\text{-}DH})$ of Diffie-Hellman tuple $(g_t, g_t^w, g_t^\delta, g_t^{\delta w})$ in $\mathbb{G}_t$. For both proofs, one can simply use Schnorr-like proofs [Sch91] with the Fiat-Shamir paradigm [FS87]. They are well-known to provide simulation-soundness [Sah99]. We now detail our construction, with access-trees for policies, where we just complete the signing key $\mathbf{k}_0^*$ with a square Diffie-Hellman tuple where one can identify the signer, if and only if the scalar w_{id} is known. The public value $g_t^{w_{\mathsf{id}}}$ associated to user id will then be enough to verify the tracing, without revealing w_{id}:

$\mathsf{Setup}(1^\kappa)$**.** The algorithm chooses three random dual orthogonal bases, in a pairing-friendly setting $\mathcal{PG} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, e, G_1, G_2, q)$:

$$\begin{aligned} \mathbb{B} &= (\mathbf{b}_1, \ldots, \mathbf{b}_6) & \mathbb{D} &= (\mathbf{d}_1, \ldots, \mathbf{d}_{10}) & \mathbb{H} &= (\mathbf{h}_1, \ldots, \mathbf{h}_8) \\ \mathbb{B}^* &= (\mathbf{b}_1^*, \ldots, \mathbf{b}_6^*) & \mathbb{D}^* &= (\mathbf{d}_1^*, \ldots, \mathbf{d}_{10}^*) & \mathbb{H}^* &= (\mathbf{h}_1^*, \ldots, \mathbf{h}_8^*). \end{aligned}$$

It also chooses two full-domain hash function $\mathcal{H}$ and $\mathcal{H}'$ onto $\mathbb{Z}_q$. The algorithm calls the $\mathsf{OT\text{-}LH}$ signature algorithm $\mathsf{KeyGen}'(1^\kappa, 6)$, for vectors in $\mathbb{G}_2^6$, and gets back the keys sk and vk. It also gets $\Sigma_2 = \mathsf{Sig}'(\mathsf{sk}, \mathbf{b}_2^*)$, and sets the public parameters as $\mathsf{PK} = \{\mathcal{PG}, \mathcal{H}, (\mathbf{b}_1, \mathbf{b}_3, \mathbf{b}_5, \mathbf{b}_6), (\mathbf{b}_2^*, \Sigma_2), (\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_5), (\mathbf{d}_1^*, \mathbf{d}_2^*, \mathbf{d}_3^*, \mathbf{d}_4^*), (\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3, \mathbf{h}_5), (\mathbf{h}_4^*), \mathsf{vk}\}$, and the master secret key is set as $\mathsf{MK} = \{(\mathbf{b}_1^*, \mathbf{b}_5^*, \mathbf{b}_6^*), (\mathbf{h}_1^*, \mathbf{h}_2^*, \mathbf{h}_3^*), \mathsf{sk}\}$. Finally, the tracing key TK and the verification key VK are initialized as empty sets.

$\mathsf{KeyGen}(\mathsf{MK}, \mathsf{id}, \Gamma)$**.** Random scalars $\delta_{\mathsf{id}}, w_{\mathsf{id}} \xleftarrow{\$} \mathbb{Z}_q^*$ are associated to id, with

$$\mathbf{k}_0^* = \delta_{\mathsf{id}} \cdot \mathbf{b}_1^* + \phi_0 \cdot \mathbf{b}_2^* + \delta_{\mathsf{id}} \cdot w_{\mathsf{id}} \cdot \mathbf{b}_5^* + \delta_{\mathsf{id}} \cdot w_{\mathsf{id}}^2 \cdot \mathbf{b}_6^*$$
$$\mathbf{k}_t^* = \delta_{\mathsf{id}} \cdot \mathbf{d}_1^* + \pi_t \cdot (\mathbf{d}_2^* + t \cdot \mathbf{d}_3^*) + \phi_t \cdot \mathbf{d}_4^*$$
$$\mathbf{r}_1^* = \delta_{\mathsf{id}} \cdot \mathbf{h}_1^* + \psi_1 \cdot \mathbf{h}_4^* \qquad \mathbf{r}_2^* = \delta_{\mathsf{id}} \cdot \mathbf{h}_2^* + \psi_2 \cdot \mathbf{h}_4^* \qquad \mathbf{r}_3^* = \delta_{\mathsf{id}} \cdot \mathbf{h}_3^* + \psi_3 \cdot \mathbf{h}_4^*$$

for all attributes $t \in \Gamma$, with $\phi_0, (\phi_t)_t, (\pi_t)_t \xleftarrow{\$} \mathbb{Z}_q^*$ for each t. The algorithm calls for $\Sigma_{\mathsf{id}} = \mathsf{Sig}'(\mathsf{sk}, \mathbf{k}_0^*)$. The signing key $\mathsf{SK}_{\mathsf{id},\Gamma}$ is set as $(w_{\mathsf{id}}, \mathbf{k}_0^*, \Sigma_{\mathsf{id}}, (\mathbf{k}_t^*)_{t \in \Gamma}, \mathbf{r}_1^*, \mathbf{r}_2^*, \mathbf{r}_3^*)$, for id. It can be computed later for new attributes, but only by using the same δ_{id}. The pair $(\mathsf{id}, w_{\mathsf{id}})$ is appended to TK, and $(\mathsf{id}, g_t^{w_{\mathsf{id}}})$ is appended to VK.

$\mathsf{Sig}(\mathsf{SK}_{\mathsf{id},\Gamma}, m, \mathcal{T})$**.** Let $\mathcal{T}' \in \mathsf{EPT}(\mathcal{T}, \Gamma)$ be an Evaluation Pruned Tree, $\nu, \xi, \zeta \xleftarrow{\$} \mathbb{Z}_q^*$. Compute the following 1-labeling of the dual tree $\mathcal{T}^*$: for each leaf λ, choose $\alpha_\lambda = 1$ if $\lambda \in \mathcal{L}_{\mathcal{T}'}$, else $\alpha_\lambda = 0$. Then, choose a random 0-labeling (β_λ) of $\mathcal{T}^*$, and $(q_\lambda)_\lambda, (\omega_\lambda)_\lambda$ random scalars, and set, for $H = \mathcal{H}(\mathcal{T}), H' = \mathcal{H}'(m)$:

$$U^* = \xi \mathbf{k}_0^* + \zeta \mathbf{b}_2^* \qquad S_\lambda^* = \alpha_\lambda \xi \cdot \mathbf{k}_{t_\lambda}^* + \beta_\lambda \mathbf{d}_1^* + \omega_\lambda (\mathbf{d}_2^* + t_\lambda \cdot \mathbf{d}_3^*) + q_\lambda \cdot \mathbf{d}_4^*$$
$$V^* = \xi(\mathbf{r}_1^* + H \cdot \mathbf{r}_2^* + H' \cdot \mathbf{r}_3^*) + \nu \cdot \mathbf{h}_4^*$$

for all the leaves λ, where t_λ is the associated attribute of λ. From the linearly-homomorphic property, one can compute a signature on $U^* \in \mathbb{G}_2^6$:

$$\Sigma = \mathsf{DerivSign}'(\mathsf{vk}, ((\xi, \mathbf{k}_0^*, \Sigma_{\mathsf{id}}), (\zeta, \mathbf{b}_2^*, \Sigma_2))$$

Eventually, using w_{id}, one can generate the proof of Square Diffie-Hellman tuple $\Pi = \mathsf{NIZKPoK\text{-}SqDH}(w_{\mathsf{id}}, (e(\mathbf{b}_1, U^*), e(\mathbf{b}_5, U^*), e(\mathbf{b}_6, U^*)))$, as this tuple is equal to $(h_t, h_t^{w_{\mathsf{id}}}, h_t^{w_{\mathsf{id}}^2})$, for some $h_t \in \mathbb{G}_t$. The final signature consists of the tuple $\sigma = (U^*, V^*, (S_\lambda^*)_\lambda), \Sigma, \Pi)$.

$\mathsf{Verif}(\mathsf{PK}, m, \mathcal{T}, \sigma)$**.** Let $\kappa, \kappa_0, (\kappa_\lambda)_\lambda, s, s_0, \theta, \theta'(\theta_\lambda)_\lambda \xleftarrow{\$} \mathbb{Z}_q$. Let $(s_\lambda)_\lambda$ be a random s_0-labeling of $\mathcal{T}$, then set, for $\bar{H} = \mathcal{H}(\mathcal{T}), \bar{H}' = \mathcal{H}'(\mathcal{T})$:

$$u = -(s_0 + s) \cdot \mathbf{b}_1 + \kappa_0 \cdot \mathbf{b}_3 \qquad c_\lambda = s_\lambda \cdot \mathbf{d}_1 + \theta_\lambda t_\lambda \cdot \mathbf{d}_2 - \theta_\lambda \cdot \mathbf{d}_3 + \kappa_\lambda \cdot \mathbf{d}_5$$
$$v = (s + \theta \bar{H} + \theta' \bar{H}') \cdot \mathbf{h}_1 - \theta \cdot \mathbf{h}_2 - \theta' \cdot \mathbf{h}_3 + \kappa \cdot \mathbf{h}_5$$

Accept if $e(\mathbf{b}_1, U^*) \neq 1_{\mathbb{G}_t}$ and $e(u, U^*) \cdot e(v, V^*) \cdot \prod e(c_\lambda, S_\lambda^*) = 1_{\mathbb{G}_t}$, but also if $\mathsf{Verif}'(\mathsf{vk}, U^*, \Sigma) = 1$ and $\mathsf{VERIF\text{-}SqDH}((e(\mathbf{b}_1, U^*), e(\mathbf{b}_5, U^*), e(\mathbf{b}_6, U^*)), \Pi) = 1$, otherwise reject.

$\mathsf{Trace}(\mathsf{TK}, \sigma')$**.** Compute $B_1 = e(\mathbf{b}_1, U^*)$ and $B_2 = e(\mathbf{b}_5, U^*)$. Then, for $(\mathsf{id}, w_{\mathsf{id}}) \in \mathsf{TK}$, check until $B_1^{w_{\mathsf{id}}} = B_2$. When the equality holds then generate the proof

$$\pi = \mathsf{NIZKPoK\text{-}DH}(g_t, g_t^{w_{\mathsf{id}}}, e(\mathbf{b}_1, U^*), e(\mathbf{b}_5, U^*))$$

and output (id, π). Otherwise output $\perp$.

$\mathsf{Judge}(\mathsf{VK}, m, \sigma', \mathsf{id}, \pi)$**.** Extract $g_t^{w_{\mathsf{id}}}$ corresponding to id from VK and output

$$\mathsf{VERIF\text{-}DH}((g_t, g_t^{w_{\mathsf{id}}}, e(\mathbf{b}_1, U^*), e(\mathbf{b}_5, U^*)), \pi)$$

As VK can be a public list, anybody can run the Judge algorithm. This means that anyone can know the current number of users in the system.

5.3 Correctness

This construction is a slight variation of the previous ABS scheme. Thus the correctness directly ensues from the correctness of the scheme formerly presented in Section 3, the correctness and linear-homomorphic property of the OT-LH scheme, and the completeness of both the zero-knowledge proof.

5.4 Security Results

Since the verification process is even more restrictive than in the previous scheme, one can claim the same unforgeability result:

Theorem 12 (Existential Unforgeability). *The ABS scheme described in Section 5.2 is existentially unforgeable under the collision-resistance of the hash functions $\mathcal{H}, \mathcal{H}'$ and the SXDH assumption.*

Because of the additional elements in the signature (which are useful for tracing), the signature is no longer perfectly anonymous, but still computationally anonymous:

Theorem 13 (Computational Anonymity). *The ABS scheme described in Section 5.2 is computationally anonymous, when w_0 and w_1, in SK_0 and SK_1, are unknown, under the Decisional Square Diffie-Hellman assumption in $\mathbb{G}_2$, and the perfect zero-knowledge of the* NIZKPoK-SqDH *in the ROM.*

Proof. The additional elements are the Square Diffie-Hellman tuple in the 1-st, 5-th, and 6-th components of $U^* = (\delta', \phi'_0, 0, 0, \delta' \cdot w_{\mathsf{id}}, \delta' \cdot w_{\mathsf{id}}^2)_{\mathbb{B}^*}$, the signature Σ, and the proof Π.

The Square Diffie-Hellman tuple in U^* can be generated from a Square Diffie-Hellman tuple $(\delta' G_2, w \cdot \delta' G_2, w^2 \cdot \delta' G_2) \in \mathbb{G}_2^3$. Under the Decisional Square Diffie-Hellman assumption in $\mathbb{G}_2$, such a tuple is indistinguishable from a random tuple in $\mathbb{G}_2^3$. This makes U^* generated from w_0 or w_1 indistinguishable when those scalars are unknown. Since Σ is a signature of U^* that is itself indistinguishable for w_0 and w_1, Σ is also indistinguishable for w_0 and w_1. Eventually, Π being a zero-knowledge proof on the above tuple, it can be simulated without knowing the witness. It thus does not leak any additional information. Hence the anonymity under the Decisional Square Diffie-Hellman assumption in $\mathbb{G}_2$.

Finally, we state the traceability result, for which the proof is postponed to Appendix C.7

Theorem 14 (Traceability). *The ABS scheme described in Section 5.2 is traceable in the ROM according to the Definition 10, in the distinct-user setting, under the security of the OT-LH signature scheme, the intractability of the Linear-Square problem, the simulation-extractability of the* NIZKPoK-SqDH*, and the soundness of the* NIZKPoK-DH*.*

References

Cv91. David Chaum and Eugène van Heyst. Group signatures. In Donald W. Davies, editor, *EUROCRYPT'91*, volume 547 of *LNCS*, pages 257–265. Springer, Heidelberg, April 1991.

DGK+21. Ivan Damgård, Chaya Ganesh, Hamidreza Khoshakhlagh, Claudio Orlandi, and Luisa Siniscalchi. Balancing privacy and accountability in blockchain identity management. In Kenneth G. Paterson, editor, *CT-RSA 2021*, volume 12704 of *LNCS*, pages 552–576. Springer, Heidelberg, May 2021.

DGM18. Constantin Catalin Dragan, Daniel Gardham, and Mark Manulis. Hierarchical attribute-based signatures. In Jan Camenisch and Panos Papadimitratos, editors, *CANS 18*, volume 11124 of *LNCS*, pages 213–234. Springer, Heidelberg, September / October 2018.

DGP22. Cécile Delerablée, Lénaïck Gouriou, and David Pointcheval. Key-policy ABE with switchable attributes. In Clemente Galdi and Stanislaw Jarecki, editors, *The 13th Conference on Security in Communication Networks (SCN '22)*, volume 13409 of *LNCS*, pages 147–171, Amalfi, Italy, 2022. Springer, Heidelberg. https://eprint.iacr.org/2021/867.

DOT19. Pratish Datta, Tatsuaki Okamoto, and Katsuyuki Takashima. Efficient attribute-based signatures for unbounded arithmetic branching programs. In Dongdai Lin and Kazue Sako, editors, *PKC 2019, Part I*, volume 11442 of *LNCS*, pages 127–158. Springer, Heidelberg, April 2019.

DZL14. Shenglong Ding, Yiming Zhao, and Yuyang Liu. Efficient traceable attribute-based signature. In *2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, pages 582–589, 2014.

EGK14. Ali El Kaafarani, Essam Ghadafi, and Dalia Khader. Decentralized traceable attribute-based signatures. In Josh Benaloh, editor, *CT-RSA 2014*, volume 8366 of *LNCS*, pages 327–348. Springer, Heidelberg, February 2014.

FS87. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Heidelberg, August 1987.

Gha15. Essam Ghadafi. Stronger security notions for decentralized traceable attribute-based signatures and more efficient constructions. In Kaisa Nyberg, editor, *CT-RSA 2015*, volume 9048 of *LNCS*, pages 391–409. Springer, Heidelberg, April 2015.

GM19. Daniel Gardham and Mark Manulis. Hierarchical attribute-based signatures: Short keys and optimal signature length. In Robert H. Deng, Valérie Gauthier-Umaña, Martín Ochoa, and Moti Yung, editors, *ACNS 19*, volume 11464 of *LNCS*, pages 89–109. Springer, Heidelberg, June 2019.

GPSW06. Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 89–98. ACM Press, October / November 2006. Available as Cryptology ePrint Archive Report 2006/309.

HP22. Chloé Hébant and David Pointcheval. Traceable constant-size multi-authority credentials. In Clemente Galdi and Stanislaw Jarecki, editors, *Security and Cryptography for Networks*, pages 411–434, Cham, 2022. Springer International Publishing.

HPP20. Chloé Hébant, Duong Hieu Phan, and David Pointcheval. Linearly-homomorphic signatures and scalable mix-nets. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part II*, volume 12111 of *LNCS*, pages 597–627. Springer, Heidelberg, May 2020.

LMY14. Weiwei Liu, Yi Mu, and Guomin Yang. Attribute-based signing right delegation. In Man Ho Au, Barbara Carminati, and C.-C. Jay Kuo, editors, *Network and System Security*, pages 323–334, Cham, 2014. Springer International Publishing.

LOS$^+$10. Allison B. Lewko, Tatsuaki Okamoto, Amit Sahai, Katsuyuki Takashima, and Brent Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 62–91. Springer, Heidelberg, May / June 2010.

LPJY13. Benoît Libert, Thomas Peters, Marc Joye, and Moti Yung. Linearly homomorphic structure-preserving signatures and their applications. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 289–307. Springer, Heidelberg, August 2013.

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

MPR11. Hemanta K. Maji, Manoj Prabhakaran, and Mike Rosulek. Attribute-based signatures. In Aggelos Kiayias, editor, *CT-RSA 2011*, volume 6558 of *LNCS*, pages 376–392. Springer, Heidelberg, February 2011.

OT11. Tatsuaki Okamoto and Katsuyuki Takashima. Efficient attribute-based signatures for non-monotone predicates in the standard model. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *PKC 2011*, volume 6571 of *LNCS*, pages 35–52. Springer, Heidelberg, March 2011.

OT12. Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure unbounded inner-product and attribute-based encryption. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 349–366. Springer, Heidelberg, December 2012.

OT13. Tatsuaki Okamoto and Katsuyuki Takashima. Decentralized attribute-based signatures. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *PKC 2013*, volume 7778 of *LNCS*, pages 125–142. Springer, Heidelberg, February / March 2013.

RST01. Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 552–565. Springer, Heidelberg, December 2001.

Sah99. Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th FOCS*, pages 543–553. IEEE Computer Society Press, October 1999.

Sch91. Claus-Peter Schnorr. Factoring integers and computing discrete logarithms via Diophantine approximations. In Donald W. Davies, editor, *EUROCRYPT'91*, volume 547 of *LNCS*, pages 281–293. Springer, Heidelberg, April 1991.

SKAH18. Yusuke Sakai, Shuichi Katsumata, Nuttapong Attrapadung, and Goichiro Hanaoka. Attribute-based signatures for unbounded languages from standard assumptions. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part II*, volume 11273 of *LNCS*, pages 493–522. Springer, Heidelberg, December 2018.

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

A One-Time Linearly-Homomorphic Signature

For our traceable ABS scheme, we will make use of a (One-Time) Linearly-Homomorphic Signature ($\mathsf{OT\text{-}LH}$). Let us first recall the definition.

$\mathsf{KeyGen}(1^\kappa, n)$**.** From the security parameter κ, and a dimension n, the algorithm outputs a signing key sk and a verification key vk;

$\mathsf{Sig}(\mathsf{sk}, m)$**.** For a signing key sk and a message m of dimension n, the algorithm outputs a signature σ;

$\mathsf{DerivSign}(\mathsf{vk}, (\alpha_i, m_i, \sigma_i))$**.** For a verification key vk, several messages m_i together with their signatures σ_i, and some coefficients α_i, the algorithm outputs a signature σ of the linear combination $\sum_i \alpha_i m_i$;

$\mathsf{Verif}(\mathsf{vk}, m, \sigma)$**.** Given a verification key vk, a message m, and a signature σ, the algorithm outputs 1 for accept or 0 for reject;

Correctness and unforgeability are similar as for usual signature schemes, except that an output (m', σ') will be considered a forgery if m' is not in the span of the messages m_i asked to the signing oracle. Indeed, linear combination of signatures on the m_i's is accessible to the adversary. We will then denote by $\mathsf{Adv}^{\mathsf{euf}}_{\mathsf{OT\text{-}LH}}$ the best advantage an adversary can have in generating a valid signature for a message out of the span of the initially signed messages. Furthermore, any signature generated by linear combinations using $\mathsf{DerivSign}$ should be perfectly indistinguishable from a fresh signature generated by Sig.

Such linearly-homomorphic signatures have been proposed in the literature, as in [LPJY13]. But we can also use the simplified version from [HP22] which has been proven in the generic group model, even together with an extractor that provides the coefficients in the linear combination of the initial messages for the new signed message. From this paper, we will also use the following theorem, that states the intractability of the Linear-Square problem:

Theorem 15 (Linear-Square Problem). *Given n Square Diffie-Hellman tuples $(g_i, a_i = g_i^{w_i}, b_i = a_i^{w_i})$, together with w_i, for random $g_i \xleftarrow{\$} \mathbb{G}^*$ and $w_i \xleftarrow{\$} \mathbb{Z}_q^*$, outputting $(\alpha_i)_{i=1,\dots,n}$ such that $(G = \prod g_i^{\alpha_i}, A = \prod a_i^{\alpha_i}, B = \prod b_i^{\alpha_i})$ is a valid Square Diffie-Hellman, with at least two non-zero coefficients α_i, is computationally hard under the Discrete Logarithm assumption.*

B Schematic View of Change of Basis theorems.

We present in Figure 6 a schematic view of the 4 change of basis theorems that are used during the security proof. We refer to [DGP22] for a full proof of the first 3 theorems, and the proof for the adaptive Indexing theorem can be found in Section C.2.

C Proofs

In the following proofs, we will sometimes use the following DSDH assumption, which is implied by the DDH assumption:

Definition 16 (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 $2 \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}}(T)$.

SubSpace-Ind on $(\mathbb{B},\mathbb{B}^*)_{1,2}$ with $\mathbf{b}_2^*$ hidden.

$$\begin{array}{rl} \mathbf{c} = & (\ \colorbox{lightgreen}{x_1}\ \ \colorbox{lightgray}{x_2}\ \ x_3\ )_{\mathbb{B}} \approx (\ \colorbox{lightgreen}{x_1}\ \ \colorbox{lightgray}{x'_2}\ \ x_3\ )_{\mathbb{B}} \\ \mathbf{k}^* = & (\ \colorbox{lightgreen}{y_1}\ \ y_2\ \ y_3\ )_{\mathbb{B}^*} = (\ \colorbox{lightgreen}{y_1}\ \ y_2\ \ y_3\ )_{\mathbb{B}^*} \end{array}$$

Swap-Ind on $(\mathbb{B},\mathbb{B}^*)_{1,2,3}$ with $\mathbf{b}_1^*, \mathbf{b}_2^*$ hidden.

$$\begin{array}{rl} \mathbf{c} = & (\ \colorbox{lightgray}{x_1}\ \ \colorbox{lightgray}{0}\ \ \colorbox{lightgreen}{x_3}\ )_{\mathbb{B}} \approx (\ \colorbox{lightgray}{0}\ \ \colorbox{lightgray}{x_1}\ \ \colorbox{lightgreen}{x_3}\ )_{\mathbb{B}} \\ \mathbf{k}^* = & (\ y_1\ \ y_1\ \ y_3\ )_{\mathbb{B}^*} = (\ y_1\ \ y_1\ \ y_3\ )_{\mathbb{B}^*} \end{array}$$

Index-Ind (static) on $(\mathbb{B},\mathbb{B}^*)_{1,2,3}$ with $\mathbf{b}_3^*$ hidden, if $p \neq t$.

$$\begin{array}{rl} \mathbf{c} = & (\ \colorbox{lightgreen}{σ} \cdot (1,p)\ \ \colorbox{lightgray}{x_3}\ )_{\mathbb{B}} \approx (\ \colorbox{lightgreen}{σ} \cdot (1,p)\ \ \colorbox{lightgray}{x'_3}\ )_{\mathbb{B}} \\ \mathbf{k}^* = & (\ \colorbox{lightgreen}{π} \cdot (t,-1)\ \ y_3\ )_{\mathbb{B}^*} = (\ \colorbox{lightgreen}{π} \cdot (t,-1)\ \ y_3\ )_{\mathbb{B}^*} \end{array}$$

Index-Ind (adaptive) on $(\mathbb{B},\mathbb{B}^*)_{1,2,3,4,5,6}$ with $\mathbf{b}_4, \mathbf{b}_5, \mathbf{b}_5^*, \mathbf{b}_6, \mathbf{b}_6^*$ hidden, if $(x,x') \neq (y,y')$ and random ρ.

$$\begin{array}{rl} \mathbf{c} = & (\ \colorbox{lightgreen}{σ} \cdot (1,\ \ y,\ \ y')\ \ \colorbox{lightgray}{x_4}\ \ 0\ \ 0\ )_{\mathbb{B}} \\ \approx & (\ \colorbox{lightgreen}{σ} \cdot (1,\ \ y,\ \ y')\ \ \colorbox{lightgray}{x'_4}\ \ 0\ \ 0\ )_{\mathbb{B}} \\ \mathbf{k}^* = & (\ \colorbox{lightgreen}{π} \cdot (x+\rho x', -1, -\rho)\ \ y_3\ \ 0\ \ 0\ )_{\mathbb{B}^*} \end{array}$$

Colored cells $\colorbox{lightgreen}{$x$}$ are **random** values, while gray cells $\colorbox{lightgray}{$x$}$ are **any** value (possibly chosen).

Fig. 6. Computationally indistinguishable changes of Basis

C.1 Proof of Dual-Tree Propositions

We remind the two propositions and prove them.

Proposition 6. *If $(a_\lambda)_\lambda$ is an a_0-labeling of $\mathcal{T}$, and $(b_\lambda)_\lambda$ is a b_0-labeling of its dual tree $\mathcal{T}^*$, then $\sum_{\lambda \in \mathcal{L}} a_\lambda b_\lambda = a_0 b_0$.*

Proof. We proceed by induction on the depth ℓ of the access-trees $\mathcal{T}$.

When $\ell = 1$, there are only two cases: any tree is either a root node labeled with an AND gate and any number of children, or it is a root node labeled with an OR gate and any number of children, and $\mathcal{T}^*$ is the alternative situation. Hence, by considering $\mathcal{T}$ with an AND-gate at the root and $\mathcal{T}^*$ with an OR-gate at the root, we address both cases at once (we just have to exchange $\mathcal{T}$ and $\mathcal{T}^*$ for the other case): $(a_\lambda)_\lambda$ is an a_0-labeling of $\mathcal{T}$, and $(b_\lambda)_\lambda$ is a b_0-labeling of $\mathcal{T}^*$. Since $\mathcal{L} = \mathsf{children}(\rho)$, because of the AND-gate, $a_0 = \sum_{\kappa \in \mathcal{L}} a_\kappa$, and because of the OR-gate, for all $\kappa \in \mathcal{L}$, $b_\kappa = b_0$: $\sum_{\lambda \in \mathcal{L}} a_\lambda b_\lambda = b_0 \sum_{\lambda \in \mathcal{L}} a_\lambda = a_0 b_0$.

Now we suppose, for the induction step, that this property holds for all $k \leq \ell \in \mathbb{N}$, and prove it holds for $\ell+1$ as well. Again, let us consider $\mathcal{T}$ an access-tree of depth $\ell+1$ with an AND-gate at the root, and $(a_\lambda)_\lambda$ an a_0-labeling of $\mathcal{T}$, then $\mathcal{T}^*$ is an access-tree of depth $\ell+1$ with an OR-gate at the root, $(b_\lambda)_\lambda$ is a b_0-labeling of $\mathcal{T}^*$ (the other case just consists in switching $\mathcal{T}$ and $\mathcal{T}^*$). Now, roots' children are subtrees of depth at most ℓ. We note $\mathcal{T}_\kappa$ the subtree rooted at $\kappa \in \mathsf{children}(\rho)$, and $\mathcal{L}_\kappa$ its leaves, and note that $(\mathcal{L}_\kappa)_\kappa$ is a strict partition of $\mathcal{L}$. As the dual tree is built by just switching AND/OR gates, the dual-tree of $\mathcal{T}_\kappa$ (the subtree rooted at κ) is the subtree of the dual-tree of $\mathcal{T}^*$ rooted at κ, that we can thus denote $\mathcal{T}_\kappa^*$ without any ambiguity.

By definition of the labelings, $a_0 = \sum_{\kappa \in \mathsf{children}(\rho)} a_\kappa$ for $\mathcal{T}$, and in $\mathcal{T}^*$, for all $\kappa \in \mathsf{children}(\rho)$, $b_\kappa = b_0$. Then, as above, $a_0 b_0 = \sum_{\kappa \in \mathsf{children}(\rho)} a_\kappa b_\kappa$. However, we also know from the induction hypothesis on all the subtrees $\mathcal{T}_\kappa$ and $\mathcal{T}_\kappa^*$, rooted at $\kappa \in \mathsf{children}(\rho)$ over the sets of leaves $\mathcal{L}_\kappa$, of depth at most ℓ, $\sum_{\lambda \in \mathcal{L}_\kappa} a_\lambda b_\lambda = a_\kappa b_\kappa$. Because of the partition property of the $\mathcal{L}_\kappa$'s into $\mathcal{L}$, $\sum_{\lambda \in \mathcal{L}} a_\lambda b_\lambda = \sum_{\kappa \in \mathsf{children}(\rho)} \sum_{\lambda \in \mathcal{L}_\kappa} a_\lambda b_\lambda = \sum_{\kappa \in \mathsf{children}(\rho)} a_\kappa b_\kappa = a_0 b_0$.

Proposition 7. *Let $\mathcal{T}$ be an access-tree and Γ a set of attributes so that $\mathcal{T}(\Gamma) = 1$. Then, for any Evaluation Pruned Tree $\mathcal{T}' \in \mathsf{EPT}(\mathcal{T}, \Gamma)$, there is a 1-labeling $(b_\lambda)_\lambda$ of the dual $\mathcal{T}^*$ which verifies: $b_\lambda = 1$ for all $\lambda \in \mathcal{L}_{\mathcal{T}'}$ and $b_\lambda = 0$ for all $\lambda \notin \mathcal{L}_{\mathcal{T}'}$.*

Proof. Let $\mathcal{T}'$ be an Evaluation Pruned Tree from $\mathsf{EPT}(\mathcal{T},\Gamma)$, where $\mathcal{T}(\Gamma)=1$. By definition of an EPT, $\mathcal{T}'$ has only one child on OR gates that come from $\mathcal{T}$, and all children on AND gates that come from $\mathcal{T}$. This translates to its dual $\mathcal{T}'^*$ having AND gates with only one child, and OR gates having all children. From there, we can easily construct a 1-labeling of $\mathcal{T}'^*$ noted $(b_\lambda)_\lambda$ where $b_\lambda=1$, for all $\lambda\in\mathcal{L}_{\mathcal{T}'}$. Indeed, since AND gates have a unique child, its label is identical to the one of the parent, and OR gates always have identical labels for the children than the one of the parent, all the 1's then go up to the root. We expand this into a 1-labeling of $\mathcal{T}^*$ by setting $b_\lambda=0$ for all $\lambda\notin\mathcal{L}_{\mathcal{T}'}$.

C.2 Proof of Indexing for Dimension 3 Orthogonal Vectors

We have presented a simple version of the theorem, but a stronger version can be considered, as in the SubSpace-Ind, Swap-Ind and 2-Dimensional Index-Ind, with bases $(\mathbb{B},\mathbb{B}^*)$ of any size n, where the last positions can be filled with any valuses $x_7,\dots,x_n,y_7,\dots y_n\in\mathbb{Z}_q$.

The Index-Ind property can be *static* or *adaptive* [OT12,DGP22] regarding p and t, i.e. p and t can be chosen before or after seeing the public parameters (with some basis vectors). The difference between both versions is important when considering the proof of an attribute-based construction, as it draws the line between bounded and unbounded universe of attributes. With static Index-Ind, the attributes (p and t) must be chosen before seeing the public parameters, which restricts attributes from a bounded universe of attributes (guessing the adversary's choice from an unbounded universe of attributes only work with negligeable probability). Meanwhile, adaptive indexing allows the simulator to answer the adversary's queries for any attribute and at any point during the security game.

We note that there is a small amount of dimensions where vectors have components set to 0. This is because these dimensions will serve as a fodder during the proof to manipulate the vectors in an indistinguishable manner, in a way that is similar to the ideas used in the the Dual System Encryption to prove adaptive security for identity and attribute-based constructions [Wat09].

We stress that in this theorem, π and σ are unknown and not under control, but $\sigma\cdot G_2$ can be known, as seen in the proof.

Proof. The proof follows a sequence of games presented in Figure 7.

Game $\mathbf{G}_0$: The adversary can choose $(y,y')\neq(x,x')$ and α,β in $\mathbb{Z}_q$, but $\pi,\sigma,\rho\overset{\$}{\leftarrow}\mathbb{Z}_q$:

$$\mathbf{u}=(\pi(x+\rho x',-1,-\rho),\beta,0,0)_{\mathbb{B}}\qquad\qquad \mathbf{v}^*=(\sigma(1,y,y'),0,0,0)_{\mathbb{B}^*}$$

Game $\mathbf{G}_1$: We replicate the first vector $(x+\rho x',-1,-\rho)$ into $(x+\rho x',-1)$, with additional $\zeta\overset{\$}{\leftarrow}\mathbb{Z}_p$:

$$\mathbf{u}=(\pi(x+\rho x',-1,-\rho),\beta,\zeta(x+\rho x',-1))_{\mathbb{B}}\qquad\qquad \mathbf{v}^*=(\sigma(1,y,y'),0,0,0)_{\mathbb{B}^*}$$

To show the indistinguishability, one applies the SubSpace-Ind property on $(\mathbb{B},\mathbb{B}^*)_{1,2,5,6}$. 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 random, which are indistinguishable under the DDH assumption in $\mathbb{G}_1$.

Let us assume we start from random dual orthogonal bases $(\mathbb{V},\mathbb{V}^*)$. Then we define the matrices

$$B=\begin{pmatrix}1&0&a&0\\0&1&0&a\\0&0&1&0\\0&0&0&1\end{pmatrix}_{1,2,5,6}\qquad\qquad B'=\begin{pmatrix}1&0&0&0\\0&1&0&0\\-a&0&1&0\\0&-a&0&1\end{pmatrix}_{1,2,5,6}$$

$$\mathbb{B}=B\cdot\mathbb{V}\qquad\qquad\qquad\mathbb{B}^*=B'\cdot\mathbb{V}^*$$

$\mathbf{G}_0$	Initial situation before indexing							
	$\mathbf{u} = ($	$\pi((x+\rho x'),$	$-1,$	$-\rho)$	β	0	0	$)_{\mathbb{B}}$
	$\mathbf{v}^* = ($	$\sigma(1,$	$y,$	$y')$	0	0	0	$)_{\mathbb{B}^*}$
$\mathbf{G}_1$	$\mathsf{SubSpace\text{-}Ind}$ on $(\mathbb{B}, \mathbb{B}^*)_{1,2,5,6}$							
	$\mathbf{u} = ($	$\pi((x+\rho x'),$	$-1,$	$-\rho)$	β	$\zeta(x+\rho x',$	$-1)$	$)_{\mathbb{B}}$
	$\mathbf{v}^* = ($	$\sigma(1,$	$y,$	$y')$	0	0	0	$)_{\mathbb{B}^*}$
$\mathbf{G}_2$	$\mathsf{SubSpace\text{-}Ind}$ on $(\mathbb{B}^*, \mathbb{B})_{1,2,3,5,6}$							
	$\mathbf{u} = ($	$\pi((x+\rho x'),$	$-1,$	$-\rho)$	β	$\zeta(x+\rho x',$	$-1)$	$)_{\mathbb{B}}$
	$\mathbf{v}^* = ($	$\sigma(1,$	$y,$	$y')$	0	$\theta(1,$	$y+\rho y')$	$)_{\mathbb{B}^*}$
$\mathbf{G}_3$	Formal change on $(\mathbb{B}, \mathbb{B}^*)_{5,6}$							
	$\mathbf{u} = ($	$\pi((x+\rho x'),$	$-1,$	$-\rho)$	β	u_1	u_2	$)_{\mathbb{B}}$
	$\mathbf{v}^* = ($	$\sigma(1,$	$y,$	$y')$	0	v_1	v_2	$)_{\mathbb{B}^*}$
$\mathbf{G}_4$	$\mathsf{SubSpace\text{-}Ind}$ on $(\mathbb{B}^*, \mathbb{B})_{5,4}$							
	$\mathbf{u} = ($	$\pi((x+\rho x'),$	$-1,$	$-\rho)$	β	u_1	u_2	$)_{\mathbb{B}}$
	$\mathbf{v}^* = ($	$\sigma(1,$	$y,$	$y')$	α	v_1	v_2	$)_{\mathbb{B}^*}$
$\mathbf{G}_5$	Formal change on $(\mathbb{B}, \mathbb{B}^*)_{5,6}$							
	$\mathbf{u} = ($	$\pi((x+\rho x'),$	$-1,$	$-\rho)$	β	$\zeta(x+\rho x',$	$-1)$	$)_{\mathbb{B}}$
	$\mathbf{v}^* = ($	$\sigma(1,$	$y,$	$y')$	α	$\theta(1,$	$y+\rho y')$	$)_{\mathbb{B}^*}$
$\mathbf{G}_6$	$\mathsf{SubSpace\text{-}Ind}$ on $(\mathbb{B}^*, \mathbb{B})_{1,2,3,5,6}$							
	$\mathbf{u} = ($	$\pi((x+\rho x'),$	$-1,$	$-\rho)$	β	$\zeta(x+\rho x',$	$-1)$	$)_{\mathbb{B}}$
	$\mathbf{v}^* = ($	$\sigma(1,$	$y,$	$y')$	α	0	0	$)_{\mathbb{B}^*}$
$\mathbf{G}_7$	$\mathsf{SubSpace\text{-}Ind}$ on $(\mathbb{B}, \mathbb{B}^*)_{1,2,5,6}$							
	$\mathbf{u} = ($	$\pi((x+\rho x'),$	$-1,$	$-\rho)$	β	0	0	$)_{\mathbb{B}}$
	$\mathbf{v}^* = ($	$\sigma(1,$	$y,$	$y')$	α	0	0	$)_{\mathbb{B}^*}$

Fig. 7. Sequence of Games for $\mathsf{Index\text{-}Ind}$ Property

The vectors $\mathbf{b}_5^*, \mathbf{b}_6^*$ can not be computed, but they are hidden from the adversary's view, and are not used in any vector. We compute the new vectors:

$$\begin{aligned}
\mathbf{u} &= (b(x+\rho x', -1, -\rho), \beta, c(x+\rho x', -1))_{\mathbb{V}} \\
&= (b(x+\rho x', -1, -\rho), \beta, (c-ab)(x+\rho x', -1)_{\mathbb{B}} \\
&= (b(x+\rho x', -1, -\rho), \beta, \tau(x+\rho x', -1)_{\mathbb{B}} \\
\mathbf{v}^* &= (\sigma(1, y, y'), 0, 0, 0)_{\mathbb{V}^*} = (\sigma(1, y, y'), 0, 0, 0)_{\mathbb{B}^*}
\end{aligned}$$

One can note that when $\tau = 0$, this is the previous game, and when τ random, we are in the new game, with $\pi = b$ and $\zeta = \tau$: $\mathsf{Adv}_0 - \mathsf{Adv}_1 \leq \mathsf{Adv}_{\mathbb{G}_1}^{\mathsf{ddh}}(t)$.

Game $\mathbf{G}_2$: We replicate the second non-orthogonal vector $(1, y, y')$ into $(1, y+\rho y')$, with additional $\theta \xleftarrow{\$} \mathbb{Z}_p$:

$$\begin{aligned}
\mathbf{u} &= (\pi(x+\rho x', -1, -\rho), \beta, \zeta(x+\rho x', -1))_{\mathbb{B}} \\
\mathbf{v}^* &= (\sigma(1, y, y'), 0, \theta(1, y+\rho y'))_{\mathbb{B}^*}
\end{aligned}$$

To show the indistinguishability, one applies the $\mathsf{SubSpace\text{-}Ind}$ property on $(\mathbb{B}^*, \mathbb{B})_{1,2,3,5,6}$. Indeed, we can consider a triple $(a \cdot G_2, b \cdot G_2, c \cdot G_2)$, where $c = ab + \tau \bmod q$ with either $\tau = 0$ or random, which are indistinguishable under the DDH assumption in $\mathbb{G}_2$.

Let us assume we start from random dual orthogonal bases $(\mathbb{V},\mathbb{V}^*)$. Then we define the matrices

$$B' = \begin{pmatrix} 1 & 0 & 0 & a & 0 \\ 0 & 1 & 0 & 0 & a \\ 0 & 0 & 1 & 0 & a\rho \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}_{1,2,3,5,6} \qquad B = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -a & 0 & 0 & 1 & 0 \\ 0 & -a & -a\rho & 0 & 1 \end{pmatrix}_{1,2,3,5,6}$$

$$\mathbb{B}^* = B' \cdot \mathbb{V}^* \qquad \mathbb{B} = B \cdot \mathbb{V}$$

The vectors $\mathbf{b}_5, \mathbf{b}_6$ can not be computed, but they are hidden from the adversary's view. We compute the new vectors:

$$\begin{aligned} \mathbf{v}^* &= (b(1,y,y'),0,c(1,y+\rho y'))_{\mathbb{V}^*} = (b(1,y,y'),0,(c-ab)(1,y+\rho y'))_{\mathbb{B}^*} \\ &= (b(1,y,y'),0,\tau(1,y+\rho y'))_{\mathbb{B}^*} \\ \mathbf{u} &= (\pi'(x+\rho x',-1,-\rho),\beta,\zeta(x+\rho x',-1))_{\mathbb{V}} \\ &= ((\pi'+a\zeta)(x+\rho x',-1,-\rho),\beta,\zeta(x+\rho x',-1))_{\mathbb{B}} \end{aligned}$$

One can note that when $\tau = 0$, this is the previous game, and when τ random, we are in the new game, with $\pi = \pi' + a\zeta$, $\sigma = b$, and $\theta = \tau$: $\mathsf{Adv}_1 - \mathsf{Adv}_2 \leq \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t)$.

Game $\mathbf{G}_3$: Since $(y,y') \neq (x,x')$, with the random ρ, excepted with probability bounded by $1/q$, $x+\rho x' \neq y + \rho y'$, and so the vectors $(1, y+\rho y')$ and $(x+\rho x',-1)$ are non-orthogonal. They can be randomized with random scalars $u_1,u_2,v_1,v_2 \xleftarrow{\$} \mathbb{Z}_p$:

$$\mathbf{u} = (\pi(x+\rho x',-1,-\rho),\beta,u_1,u_2)_{\mathbb{B}} \qquad \mathbf{v}^* = (\sigma(1,y,y'),0,v_1,v_2)_{\mathbb{B}^*}$$

To show the indistinguishability, one makes a formal change of basis on $(\mathbb{B},\mathbb{B}^*)_{4,5}$, with a random unitary matrix Z, with $z_1z_4 - z_2z_3 = 1$:

$$B = Z = \begin{pmatrix} z_1 & z_2 \\ z_3 & z_4 \end{pmatrix}_{5,6} \qquad B' = \begin{pmatrix} z_4 & -z_3 \\ -z_2 & z_1 \end{pmatrix}_{5,6}$$

$$\mathbb{B} = B \cdot \mathbb{V} \qquad \mathbb{B}^* = B' \cdot \mathbb{V}^*$$

This only impacts the hidden vectors $(\mathbf{b}_5,\mathbf{b}_6)$, $(\mathbf{b}_5^*,\mathbf{b}_6^*)$. If one defines $\mathbf{u}$ and $\mathbf{v}^*$ in $(\mathbb{V},\mathbb{V}^*)$, this translates in $(\mathbb{B},\mathbb{B}^*)$:

$$\begin{aligned} \mathbf{u} &= (\pi(x+\rho x',-1,-\rho),\beta,\zeta(x+\rho x',-1))_{\mathbb{V}} \\ &= (\pi(x+\rho x',-1,-\rho),\beta,\zeta((x+\rho x')z_1 - z_2,(x+\rho x')z_3 - z_4))_{\mathbb{B}} \\ \mathbf{v}^* &= (\sigma(1,y,y'),0,\theta(1,y+\rho y'))_{\mathbb{V}^*} \\ &= (\sigma(1,y,y'),0,\theta(z_4 - (y+\rho y')z_3, -z_2 + (y+\rho y')z_1))_{\mathbb{B}^*} \end{aligned}$$

Let us consider random $u_1,u_2,v_1,v_2 \xleftarrow{\$} \mathbb{Z}_p$, and solve the system:

$$\left\{\begin{aligned} \zeta((x+\rho x')z_1 - z_2) &= u_1 \\ \rho((x+\rho x')z_3 - z_4) &= u_2 \\ \theta(z_4 - (y+\rho y')z_3) &= v_1 \\ \theta(-z_2 + (y+\rho y')z_1) &= v_2 \end{aligned}\right. \qquad \left\{\begin{aligned} (x+\rho x')z_1 - z_2 &= u_1/\rho \\ -z_2 + (y+\rho y')z_1 &= v_2/\theta \\ (x+\rho x')z_3 - z_4 &= u_2/\rho \\ z_4 - (y+\rho y')z_3 &= v_1/\theta \end{aligned}\right.$$

$$\left\{\begin{aligned} ((x+\rho x') - (y+\rho y'))z_1 &= u_1/\rho - v_2/\theta \\ -z_2 + (y+\rho y')z_1 &= v_2/\theta \\ ((x+\rho x') - (y+\rho y'))z_3 &= u_2/\rho + v_1/\theta \\ z_4 - (y+\rho y')z_3 &= v_1/\theta \end{aligned}\right.$$

This system admits a solution if $y+\rho y' \neq x+\rho x'$, which holds with overwhelming probability for a random $\rho \overset{\$}{\leftarrow} \mathbb{Z}_q$ if $(y,y') \neq (x,x')$. And with random θ and random unitary matrix Z,

$$\mathbf{u} = (\pi(x+\rho x', -1, -\rho), \beta, u_1, u_2)_{\mathbb{B}} \qquad \mathbf{v}^* = (\sigma(1,y,y'), 0, v_1, v_2)_{\mathbb{B}^*}$$

with random scalars $u_1, u_2, v_1, v_2 \overset{\$}{\leftarrow} \mathbb{Z}_p$.
As a consequence, in bases $(\mathbb{V}, \mathbb{V}^*)$, we are in the previous game, and in bases $(\mathbb{B}, \mathbb{B}^*)$, we are in the new game, if $(y,y') \neq (x,x')$: $\mathsf{Adv}_2 - \mathsf{Adv}_3 \leq 1/q$, on the random choice of ρ.

Game $\mathbf{G}_4$: We now randomize the fourth component in $\mathbf{v}^*$:

$$\mathbf{u} = (\pi(x+\rho x', -1, -\rho), \beta, u_1, u_2)_{\mathbb{B}} \qquad \mathbf{v}^* = (\sigma(1,y,y'), \alpha, v_1, v_2)_{\mathbb{B}^*}$$

To show the indistinguishability, one applies the $\mathsf{SubSpace\text{-}Ind}$ property on $(\mathbb{B}^*, \mathbb{B})_{5,4}$. Indeed, we can consider a triple $(a \cdot G_2, b \cdot G_2, c \cdot G_2)$, where $c = ab + \tau \bmod q$ with either $\tau = 0$ or $\tau = \alpha$, which are indistinguishable under the DDH assumption in $\mathbb{G}_2$.
Let us assume we start from random dual orthogonal bases $(\mathbb{V}, \mathbb{V}^*)$. Then we define the matrices

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

The vectors $\mathbf{b}_4$ can not be computed, but it is not provided the adversary's view. We compute the new vectors:

$$\begin{aligned} \mathbf{v}^* &= (\sigma(1,y,y'), c, b, v_2)_{\mathbb{V}^*} = (\sigma(1,y,y'), c-ab, b, v_2)_{\mathbb{B}^*} \\ &= (\sigma(1,y,y'), \tau, b, v_2)_{\mathbb{B}^*} \\ \mathbf{u} &= (\pi(x+\rho x', -1, -\rho), \beta, u_1', u_2)_{\mathbb{V}} = (\pi(x+\rho x', -1, -\rho), \beta, u_1' + a\beta, u_2)_{\mathbb{B}} \end{aligned}$$

One can note that when $\tau = 0$, this is the previous game, and when $\tau = \alpha$, we are in the new game, with $v_1 = b$ and $u_1 = u_1' + \beta$: $\mathsf{Adv}_3 - \mathsf{Adv}_4 \leq 2 \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t)$.

Game $\mathbf{G}_5$: We now undo the game $\mathbf{G}_3$:

$$\mathbf{u} = (\pi(x+\rho x', -1, -\rho), \beta, \zeta(x+\rho x', -1))_{\mathbb{B}} \qquad \mathbf{v}^* = (\sigma(1,y,y'), \alpha, \theta(1, y+\rho y'))_{\mathbb{B}^*}$$

The same analysis can lead to $\mathsf{Adv}_4 = \mathsf{Adv}_5$, as we have already aborter is case of bad choice for ρ.

Game $\mathbf{G}_6$: We undo the game $\mathbf{G}_2$:

$$\mathbf{u} = (\pi(x+\rho x', -1, -\rho), \beta, \zeta(x+\rho x', -1))_{\mathbb{B}} \qquad \mathbf{v}^* = (\sigma(1,y,y'), \alpha, 0, 0)_{\mathbb{B}^*}$$

The same analysis can lead to $\mathsf{Adv}_5 - \mathsf{Adv}_6 \leq \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t)$.

Game $\mathbf{G}_7$: We undo the game $\mathbf{G}_1$:

$$\mathbf{u} = (\pi(x+\rho x', -1, -\rho), \beta, 0, 0)_{\mathbb{B}} \qquad \mathbf{v}^* = (\sigma(1,y,y'), \alpha, 0, 0)_{\mathbb{B}^*}$$

The same analysis can lead to $\mathsf{Adv}_6 - \mathsf{Adv}_7 \leq \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t)$.

The global difference of advantages is bounded by $4 \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t) + 2 \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t) + 1/q$.

Proof of Theorem 8: Existential Unforgeability, with Delegation. We now reduce the EUF proof for our ABS with delegation to the proof without delegation. To do this, simulate the oracles for delegated keys: keys from ODelegateAttributes can be simulated with OKeyGen, and signatures from ODelegateSig and policy keys from ODelegatePolicy can be simulated with AltSig. For this reason, we will use the same notation id to count ODelegateAttributes and OKeyGen queries, and the notation i to count OSig, ODelegatePolicy and ODelegateSig.

Game $\mathbf{G}_0$: Setup, KeyGen, Sig and Verif work exactly as in game $\mathbf{G}_0$.
We detail here the distribution of all the new algorithms output. Keys generated by the Delegate-Attributes algorithm, for user id and subset Γ', follow the distribution

$$\begin{aligned} & & \mathbf{r}^*_{\mathsf{id},1} &= (\alpha\delta_{\mathsf{id}}, 0, 0, \psi_{\mathsf{id},1}, 0^4)_{\mathbb{H}^*} \\ \mathbf{r}^*_{\mathsf{id},2} &= (0, \alpha\delta_{\mathsf{id}}, 0, \psi_{\mathsf{id},2}, 0^4)_{\mathbb{H}^*} & \mathbf{r}^*_{\mathsf{id},3} &= (0, 0, \alpha\delta_{\mathsf{id}}, \psi_{\mathsf{id},3}, 0^4)_{\mathbb{H}^*} \\ \mathbf{k}^*_{\mathsf{id},0} &= (\alpha\delta_{\mathsf{id}}, \phi_{\mathsf{id},0}, 0^2)_{\mathbb{B}^*} & \mathbf{k}^*_{\mathsf{id},t} &= (\alpha\delta_{\mathsf{id}}, \alpha\pi_{\mathsf{id},t}(1,t), \phi_{\mathsf{id},t}, 0^6)_{\mathbb{D}^*} \qquad \forall t \in \Gamma' \end{aligned}$$

for random $\alpha, \delta_{\mathsf{id}}, \psi_{\mathsf{id},1}, \psi_{\mathsf{id},2}, \psi_{\mathsf{id},3}, \phi_{\mathsf{id},0}, (\phi_{\mathsf{id},t})_t, (\pi_{\mathsf{id},t})_t \xleftarrow{\$} \mathbb{Z}_q$ for $t \in \Gamma'$.
The i-th policy keys generated by the Delegate-Policy algorithm, for user id and access-tree $\mathcal{T}$, follow the distribution, for $H = \mathcal{H}(\mathcal{T})$:

$$\begin{aligned} U^*_i &= (\delta_i, \zeta_i, 0^2)_{\mathbb{B}^*} & S^*_\lambda &= ((\alpha_{\lambda,i}\delta_i + \beta_{\lambda,i}), \omega_{i,\lambda}(1, t_\lambda), q_{i,\lambda}, 0^6)_{\mathbb{D}^*} \\ V^* &= (\delta_i \cdot (1, H, 0), \nu_i, 0^4)_{\mathbb{H}^*} & \mathbf{r}^*_3 &= (\delta_i \cdot (0,0,1), \psi'_{i,3}, 0^4)_{\mathbb{H}^*} \end{aligned}$$

for random scalars $\delta_i, \zeta_i, (q_{i,\lambda})_\lambda, (\omega_{i,\lambda})_\lambda, \nu_i, \psi_{i,3} \xleftarrow{\$} \mathbb{Z}_q$ for $\lambda \in \mathcal{L}_\mathcal{T}$.
Signatures generated via policy keys with the DelegateSig algorithm for the i-th signature are generated as, for $H_i = \mathcal{H}(\mathcal{T}_i)$, $H'_i = \mathcal{H}'(m_i)$:

$$\begin{aligned} U^* &= (\delta_i, \zeta_i, 0^2)_{\mathbb{B}^*} & S^*_\lambda &= ((\alpha_{\lambda,i}\delta_i + \beta_{\lambda,i}), \omega_{\lambda,i}(1, t_\lambda), q_{i,\lambda}, 0^6)_{\mathbb{D}^*} \\ V^* &= (\delta_i \cdot (1, H_i, H'_i), \nu_i, 0^4)_{\mathbb{H}^*} \end{aligned}$$

for random scalars $\delta_i, \zeta_i, \nu_i, (q_{i,\lambda})_\lambda, (\omega_{i,\lambda})_\lambda \xleftarrow{\$} \mathbb{Z}_q^*$, and still $(\alpha_{\lambda,i})$ a 1-labeling of $\mathcal{T}^*$, and $(\beta_{\lambda,i})$ a random 0-labeling of $\mathcal{T}^*$.

Game $\mathbf{G}_1$: We simulate all Delegate-Attributes queries using KeyGen exclusively. When the adversary ask for Delegate-Attributes on the set Γ' from another key $\mathsf{SK}_{\mathsf{id},\Gamma}$, where $\Gamma' \subset \Gamma$, we simulate the answer as $\mathsf{KeyGen}(\mathsf{MK}, \mathsf{id}', \Gamma')$ for a new random id'. As the correctness analysis has shown, the distribution between original keys and delegated keys is exactly the same, hence: $\mathsf{Adv}_0 = \mathsf{Adv}_1$

Game $\mathbf{G}_2$: We simulate all Sig and DelegateSig queries with the AltSig algorithm. Queries for the i-th Sig, or the i-th signature with DelegateSig, on access-tree $\mathcal{T}_i$ and message m_i, is simulated as, for $H_i = \mathcal{H}(\mathcal{T}_i), H'_i = \mathcal{H}'_i(m_i)$:

$$\begin{aligned} U^*_i &= (\delta_i, \zeta_i, 0^2)_{\mathbb{B}^*} & S^*_{i,\lambda} &= (\delta_i\beta'_{i,\lambda}, \gamma_{\lambda,i}(1, t_\lambda), q_{i,\lambda}, 0,^6\ )_{\mathbb{D}^*} \\ V^*_i &= (\delta_i \cdot (1, H_i, H'_i), \nu_i, 0^4)_{\mathbb{H}^*} \end{aligned}$$

where $(\beta'_\lambda)_\lambda$ is a random 1-labeling of $\mathcal{T}^*$ and $(\gamma_{\lambda,i}) \xleftarrow{\$} \mathbb{Z}_q^*$. As shown in the above perfect anonymity proof, the distribution is exactly the same, hence the simulation is perfect: $\mathsf{Adv}_1 = \mathsf{Adv}_2$.

Game $\mathbf{G}_3$: We simulate the Delegate-Policy queries with the AltSig algorithm, with only a simple tweak on the element V^*. Queries for the i-th Delegate-Policy, on identity $\bar{\mathsf{id}}$, access-tree $\mathcal{T}_i$ and message m_i, is simulated with $\mathsf{AltSig}(\mathsf{MK}, m, \mathcal{T}_i)$ for a random m to get back

$(U_i^*, V_i^*, (S_{i,\lambda}^*)_{\lambda\in\mathcal{L}_{\mathcal{T}_i}})$. We then set $\mathbf{r}_{i,3}^* = \delta_i\cdot\mathbf{h}_3^* + \psi_{i,3}\mathbf{h}_4^*$ and $V_i'^* = V_i^* - H_i'\cdot\mathbf{r}_{i,3}^*$, for $\psi_{i,3} \xleftarrow{\$} \mathbb{Z}_q^*$: $V_i'^* = V_i^* - H_i'\cdot\mathbf{r}_{i,3}^* = (\delta_i\cdot(1, H_i, H_i'), \nu_i, 0^4)_{\mathbb{H}^*} - (\delta_i\cdot(0,0,H_i'), \psi_{i,3}H_i', 0^4)_{\mathbb{H}^*}$, which is thus as $(\delta_i\cdot(1, H_i, 0), \nu_i', 0^4)_{\mathbb{H}^*}$, for $\nu_i' \xleftarrow{\$} \mathbb{Z}_q^*$. To properly simulate these queries, we only need to ensure that we can simulate $\mathbf{r}_{i,3}^*$ in all games where we modify the answers of the signature oracle. We finally output the policy key from the query as: $(U_i^*, V_i'^*, \mathbf{r}_{i,3}^*, (S_{i,\lambda}^*)_{\lambda\in\mathcal{L}_{\mathcal{T}_i}})$. Once again the simulation is perfect: $\mathsf{Adv}_2 = \mathsf{Adv}_3$.

Then, we are in a similar game as $\mathbf{G}_0$ for the Existential Unforgeability proof in the case without delegation. The sequence of games can continue the same way, with thus the same security bounds.

C.3 Proof of Theorem 8: Existential Unforgeability, without Delegation.

The security proof follows the sequence of games presented on Figure 5.

Game $\mathbf{G}_0$: From the correctness of the signature and the perfect anonymity, keys generated by the KeyGen algorithm, for user id, follow the distribution:

$$\begin{aligned}
\mathbf{k}_{\mathsf{id},0}^* &= (\delta_{\mathsf{id}}, \phi_{\mathsf{id},0}, 0^2)_{\mathbb{B}^*} & \mathbf{k}_{\mathsf{id},t}^* &= (\delta_{\mathsf{id}}, \pi_{\mathsf{id},t}(1,t), \phi_{\mathsf{id},t}, 0^6)_{\mathbb{D}^*} & \forall t\\
\mathbf{r}_{\mathsf{id},1}^* &= (\delta_{\mathsf{id}}, 0, 0, \psi_{\mathsf{id},1}, 0^4)_{\mathbb{H}^*} & \mathbf{r}_{\mathsf{id},2}^* &= (0, \delta_{\mathsf{id}}, 0, \psi_{\mathsf{id},2}, 0^4)_{\mathbb{H}^*} &\\
& & \mathbf{r}_{\mathsf{id},3}^* &= (0, 0, \delta_{\mathsf{id}}, \psi_{\mathsf{id},3}, 0^4)_{\mathbb{H}^*} &
\end{aligned}$$

and the i-th signature generated by the Sig algorithm follows

$$\begin{aligned}
U_i^* &= (\delta_i, \zeta_i, 0^2)_{\mathbb{B}^*} & S_{i,\lambda}^* &= (\beta_{i,\lambda}', \gamma_{i,\lambda}(1, t_\lambda), q_{i,\lambda}, 0^6)_{\mathbb{D}^*}\\
& & V_i^* &= (\delta_i(1, H_i, H_i'), \nu_i, 0^4)_{\mathbb{H}^*}
\end{aligned}$$

where $H_i = \mathcal{H}(\mathcal{T}_i), H_i' = \mathcal{H}'(m_i)$.
For the decision of validity of the forgery $S = (U^*, V^*, (S_\lambda^*)_\lambda)$ on message m' and policy $\mathcal{T}'$, one uses

$$\begin{aligned}
u &= (-s_0 - s, 0, \kappa_0, 0)_{\mathbb{B}} & v &= (s + \theta\bar{H} + \theta'\bar{H}', -\theta, -\theta', 0, \kappa, 0, 0, 0)_{\mathbb{H}}\\
& & c_\lambda &= (s_\lambda, \theta_\lambda t_\lambda, -\theta_\lambda, 0, \kappa_\lambda, 0, 0, 0, 0, 0)_{\mathbb{D}}
\end{aligned}$$

where $\bar{H} = \mathcal{H}(\mathcal{T}), \bar{H}' = \mathcal{H}'(m)$, and $(\mathcal{T}, m) \neq (\mathcal{T}_i, m_i)$ for all i. Instead of outputting just the decision, one can consider the challenger outputs $(u, v, (c_\lambda)_\lambda)$, and everybody can make the final verification:

$$e(\mathbf{b}_1, U^*) \neq 1_{\mathbb{G}_t} \qquad\qquad e(u, U^*)\cdot e(v, V^*)\cdot\prod e(c_\lambda, S_\lambda^*) = 1_{\mathbb{G}_t}$$

And we denote by Adv_0 the probability of the validity of the forgery. Our goal is to show this is negligible.

Game $\mathbf{G}_1$: We change the verification vectors into

$$\begin{aligned}
u &= (-s_0 - s, 0, \kappa_0, -r_0)_{\mathbb{B}} & v &= (s + \theta\bar{H} + \theta'\bar{H}', -\theta, -\theta', 0, \kappa, \omega, 0, 0)_{\mathbb{H}}\\
& & c_\lambda &= (s_\lambda, \theta_\lambda t_\lambda, -\theta_\lambda, 0, \kappa_\lambda, r_\lambda, 0, 0, 0, 0)_{\mathbb{D}}
\end{aligned}$$

where r_0 and ω are random scalars and $(r_\lambda)_\lambda$ is a random r_0-labeling for the tree-policy $\mathcal{T}'$. The previous game and this game are indistinguishable under the DDH assumption in $\mathbb{G}_1$: one applies the $\mathsf{SubSpace\text{-}Ind}$ property on $(\mathbb{B}, \mathbb{B}^*)_{3,4}$, $(\mathbb{D}, \mathbb{D}^*)_{4,5}$ and $(\mathbb{H}, \mathbb{H}^*)_{4,5}$. 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 $\tau = 1$, which are indistinguishable under the DDH assumption in $\mathbb{G}_1$.

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

$$\begin{array}{llll}
B = \begin{pmatrix} 1 & -a \\ 0 & 1 \end{pmatrix}_{3,4} & B' = \begin{pmatrix} 1 & 0 \\ a & 1 \end{pmatrix}_{3,4} & D = \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}_{5,6} & D' = \begin{pmatrix} 1 & 0 \\ -a & 1 \end{pmatrix}_{5,6} \\
\mathbb{B} = B \cdot \mathbb{U} & \mathbb{B}^* = B' \cdot \mathbb{U}^* & \mathbb{D} = D \cdot \mathbb{V} & \mathbb{D}^* = D' \cdot \mathbb{V}^* \\
H = \begin{pmatrix} 1 & -a \\ 0 & 1 \end{pmatrix}_{4,5} & H' = \begin{pmatrix} 1 & 0 \\ a & 1 \end{pmatrix}_{4,5} & & \\
\mathbb{H} = H \cdot \mathbb{W} & \mathbb{H}^* = H' \cdot \mathbb{W}^* & &
\end{array}$$

The vectors $\mathbf{b}_4^*$, $\mathbf{d}_5^*$, and $\mathbf{h}_5^*$ can not be computed, but they are hidden from the adversary's view, and are not used in any vector. We compute the new vectors:

$$\begin{aligned}
u &= (-s_0 - s, 0, \kappa_0, 0)_{\mathbb{B}} - (0, 0, br_0, -cr_0)_{\mathbb{U}} \\
&= (-s_0 - s, 0, \kappa_0, 0)_{\mathbb{B}} - (0, 0, b_1 r_0, -(c - ab)r_0)_{\mathbb{B}} \\
&= (-s_0 - s, 0, \kappa_0 + br_0, -\tau r_0)_{\mathbb{B}} \\
v &= (s + \theta \bar{H} + \theta' \bar{H}', -\theta, -\theta', 0, \kappa, 0, 0, 0)_{\mathbb{H}} + (0, 0, 0, 0, -b\omega, c\omega, 0, 0)_{\mathbb{W}} \\
&= (s + \theta \bar{H} + \theta' \bar{H}', -\theta, -\theta', 0, \kappa, 0, 0, 0)_{\mathbb{H}} + (0, 0, 0, 0, -b\omega, (c - ab)\omega, 0, 0)_{\mathbb{H}} \\
&= (s + \theta \bar{H} + \theta' \bar{H}', -\theta, -\theta', 0, \kappa - b\omega, \tau\omega, 0, 0)_{\mathbb{H}} \\
c_\lambda &= (s_\lambda, \theta_\lambda t_\lambda, -\theta_\lambda, 0, \kappa_\lambda, 0, 0, 0, 0, 0)_{\mathbb{D}} + (0, 0, 0, 0, -br_\lambda, cr_\lambda, 0, 0, 0, 0)_{\mathbb{V}} \\
&= (s_\lambda, \theta_\lambda t_\lambda, -\theta_\lambda, 0, \kappa_\lambda, 0, 0, 0, 0, 0)_{\mathbb{D}} + (0, 0, 0, 0, -br_\lambda, (c - ab)r_\lambda, 0, 0, 0, 0)_{\mathbb{D}} \\
&= (s_\lambda, \theta_\lambda t_\lambda, -\theta_\lambda, 0, \kappa_\lambda - br_\lambda, \tau r_\lambda, 0, 0, 0, 0)_{\mathbb{D}}
\end{aligned}$$

One can easily note that when $\tau = 0$, this is the previous game, and when $\tau = 1$, we are in the new game.
On the other side, keys and signatures are unchanged, as their values on the corresponding unknown basis vectors are 0. They can thus be directly defined in $\mathbb{B}^*$, $\mathbb{D}^*$, and $\mathbb{H}^*$. We thus have $\mathsf{Adv}_0 - \mathsf{Adv}_1 \leq 2 \times \mathsf{Adv}_{\mathbb{G}_1}^{\mathsf{ddh}}(t)$ (as shown in [DGP22]).

Game $\mathbf{G}_2$: We introduce a random value δ''_{id} in every key in basis $\mathbb{B}$, in the component corresponding to the random value r_0 that was just introduced in the verification vector u. In order to do this, we proceed with an hybrid game on the key queries, modifying them one id at a time. We will denote the current key by Δ, and we update the Δ-th key as:

$$\mathbf{k}_{\Delta,0}^* = (\delta_\Delta, \phi_{\mathsf{id},0}, 0, \delta''_\Delta)_{\mathbb{B}^*}$$

When $\Delta = 0$, no key has been modified, this is exactly the game $\mathbf{G}_1$: $\mathbf{G}_1 = \mathbf{G}_{1.0.0}$, whereas for $\Delta = K$, all the keys have been modified, this is exactly the expected game $\mathbf{G}_2$: $\mathbf{G}_2 = \mathbf{G}_{1.K.0}$. In Appendix C.4, we show that for each Δ,

$$\mathsf{Adv}_{1.\Delta.0} - \mathsf{Adv}_{1.\Delta+1.0} \leq 6T \times \mathsf{Adv}_{\mathbb{G}_1}^{\mathsf{ddh}}(t) + (4T^2 + 6) \times \mathsf{Adv}_{\mathbb{G}_2}^{\mathsf{ddh}}(t).$$

Hence, globally, we have

$$\mathsf{Adv}_1 - \mathsf{Adv}_2 \leq 6KT \times \mathsf{Adv}_{\mathbb{G}_1}^{\mathsf{ddh}}(t) + (4T^2 + 6)K \times \mathsf{Adv}_{\mathbb{G}_2}^{\mathsf{ddh}}(t).$$

Game $\mathbf{G}_3$: In this game, we replace r_0 in the verification vector by a random independent $r'_0 \xleftarrow{\$} \mathbb{Z}_q$:

$$u = (-s_0 - s, 0, \kappa_0, r'_0)_{\mathbb{B}}$$

To do this, we proceed with a formal change of basis $\mathbb{B}$. Let us assume we start from random dual orthogonal bases $(\mathbb{U}, \mathbb{U}^*)$. Then we define the matrices, with random $\theta \overset{\$}{\leftarrow} \mathbb{Z}_q^*$

$$B = \big(\theta\big)_4 \qquad\qquad B' = \big(1/\theta\big)_4$$
$$\mathbb{B} = B \cdot \mathbb{U} \qquad\qquad \mathbb{B}^* = B' \cdot \mathbb{U}^*$$

which modifies only the hidden basis vectors $\mathbf{b}_4, \mathbf{b}_4^*$. Since they are not in the adversary's view, the advantage is not modified: $\mathsf{Adv}_3 = \mathsf{Adv}_2$. Furthermore

$$\mathbf{k}^*_{\Delta,0} = (\delta_\Delta, \phi_{id,0}, 0, \delta''_\Delta)_{\mathbb{U}^*} = (\delta_\Delta, \phi_{id,0}, 0, \theta\delta''_\Delta)_{\mathbb{B}^*}$$
$$u = (-s_0 - s, 0, \kappa_0, r_0)_{\mathbb{U}} = (-s_0 - s, 0, \kappa_0, r_0/\theta)_{\mathbb{B}}$$

Which replaces the random value δ''_Δ by another random value $\theta\delta''_\Delta$ that follows the same uniform distribution, and $r'_0 = -r_0/\theta$ follows a uniform independent distribution, also independent from the r_0-labeling $(r_\lambda)_\lambda$.

Game $\mathbf{G}_4$: We now update generated signatures, with random values in coordinates corresponding to the random ω that was introduced in the verification vector v in game $\mathbf{G}_1$. In order to do this, we proceed with an hybrid game on the signature queries, modifying them one i at a time. We will denote the current signature by j, and we update the j-th signature as:

$$U_j^* = (\delta_j, \zeta_j, 0, \rho_j)_{\mathbb{B}^*}$$
$$V_j^* = (\delta_j(1, H_j, H'_j), \nu_j, 0, \tau_j, 0, 0)_{\mathbb{H}^*}$$

with $\rho_j, \tau_j \overset{\$}{\leftarrow} \mathbb{Z}_q$.
When $j = 0$, no signature has been modified, this is exactly the game $\mathbf{G}_3$: $\mathbf{G}_3 = \mathbf{G}_{3.0.0}$, whereas for $j = S$, all the signatures have been modified, this is exactly the expected game $\mathbf{G}_4$: $\mathbf{G}_4 = \mathbf{G}_{3.S.0}$. In Section C.5, we show that for each j,

$$\mathsf{Adv}_{3.j.0} - \mathsf{Adv}_{3.j+1.0} \leq 4 \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t) + 2 \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t) + 1/q,$$

if $(\bar{H}, \bar{H}') \neq (H_j, H'_j)$, which holds under the collision resistance of the two hash functions. Hence, globally, we have

$$\mathsf{Adv}_3 - \mathsf{Adv}_4 \leq S \times (4 \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t) + 2 \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t) + 1/q) + \mathsf{Adv}^{\mathsf{coll}}_{\mathcal{H}}(t) + \mathsf{Adv}^{\mathsf{coll}}_{\mathcal{H}'}(t).$$

Game $\mathbf{G}_5$: In this final game, we make the verification vector reject all the signatures by removing the original secret on the first position:

$$u = (s', 0, \kappa_0, r'_0)_{\mathbb{B}}$$

To do this, we define the matrices, with $\Theta \overset{\$}{\leftarrow} \mathbb{Z}_q$

$$B' = \begin{pmatrix} 1 & -\Theta \\ 0 & 1 \end{pmatrix}_{1,4} \qquad\qquad B = \begin{pmatrix} 1 & 0 \\ \Theta & 1 \end{pmatrix}_{1,4}$$
$$\mathbb{B}^* = B' \cdot \mathbb{U}^* \qquad\qquad \mathbb{B} = B \cdot \mathbb{U}$$

which modifies the hidden vectors $\mathbf{b}_4, \mathbf{b}_1^*$. Since they are not in the adversary's view, the advantage is not modified: $\mathsf{Adv}_5 = \mathsf{Adv}_4$. The verification vector is modified as

$$u = (-s_0 - s, 0, \kappa_0, r'_0)_{\mathbb{U}} = (-s_0 - s - \Theta r'_0, 0, \kappa_0, r'_0)_{\mathbb{B}} = (s', 0, \kappa_0, r'_0)_{\mathbb{B}}$$

with $s' := -s_0 - s - \Theta r'_0$ that is uniformly distributed. Meanwhile, the keys and signatures are modified as follows:

$$\begin{aligned}\mathbf{k}^*_{\mathsf{id},0} &= (\delta_{\mathsf{id}}, \phi_{\mathsf{id},0}, 0, \delta''_{\mathsf{id}})_{\mathbb{U}^*} = (\delta_{\mathsf{id}}, \phi_{\mathsf{id},0}, 0, \delta''_{\mathsf{id}} + \Theta\delta_{\mathsf{id}})_{\mathbb{B}^*} = (\delta_{\mathsf{id}}, \phi_{\mathsf{id},0}, 0, \delta'_{\mathsf{id}})_{\mathbb{B}^*} \\ U^*_i &= (\delta_i, \zeta_i, 0, \rho_i)_{\mathbb{U}^*} = (\delta_i, \zeta_i, 0, \rho_i + \Theta\delta_i)_{\mathbb{B}^*} = (\delta_i, \zeta_i, 0, \rho'_i)_{\mathbb{B}^*}\end{aligned}$$

If we combine all the steps:

$$\begin{aligned}\mathsf{Adv}_0 - \mathsf{Adv}_5 \leq\ & 2 \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t) \\ & + 6KT \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t) + (4T^2 + 6)K \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t) + 0 \\ & + S \times (4 \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t) + 2 \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t) + 1/q) + \mathsf{Adv}^{\mathsf{coll}}_{\mathcal{H}}(t) + \mathsf{Adv}^{\mathsf{coll}}_{\mathcal{H}'}(t) \\ & \leq (2S + 2 + 6KT) \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t) + (4T^2K + 6K + 4S) \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t) \\ & + S/q + \mathsf{Adv}^{\mathsf{coll}}_{\mathcal{H}}(t) + \mathsf{Adv}^{\mathsf{coll}}_{\mathcal{H}'}(t)\end{aligned}$$

C.4 Existential Unforgeability: Gap between $\mathbf{G}_1$ and $\mathbf{G}_2$

In this sequence of games, detailed on Figure 8, we will stop tracking elements on bases $\mathbb{H}, \mathbb{H}^*$ as they are not modified.

$\mathbf{G}_{1.\Delta.0}$	Hybrid sequence from $\mathbf{G}_1$ to $\mathbf{G}_2$											
$\mathsf{id} \geq \Delta$	$\mathbf{k}^*_{\mathsf{id},0} = ($	δ_{id}	$\phi_{\mathsf{id},0}$	0	0							$)_{\mathbb{B}^*}$
$\mathsf{id} < \Delta$	$\mathbf{k}^*_{\mathsf{id},0} = ($	δ_{id}	$\phi_{\mathsf{id},0}$	0	δ''_{id}							$)_{\mathbb{B}^*}$
	$U^*_i = ($	δ_i	ζ_i	0	0							$)_{\mathbb{B}^*}$
	$u = ($	$-s_0 - s$	0	κ_0	$-r_0$							$)_{\mathbb{B}}$
	$\mathbf{r}^*_{\mathsf{id},1} = ($	δ_{id}	0	0	$\psi_{\mathsf{id},1}$	0	0	0	0			$)_{\mathbb{H}^*}$
	$\mathbf{r}^*_{\mathsf{id},2} = ($	0	δ_{id}	0	$\psi_{\mathsf{id},2}$	0	0	0	0			$)_{\mathbb{H}^*}$
	$\mathbf{r}^*_{\mathsf{id},3} = ($	0	0	δ_{id}	$\psi_{\mathsf{id},3}$	0	0	0	0			$)_{\mathbb{H}^*}$
	$V^*_i = ($	δ_i	$\delta_i H_i$	$\delta_i H'_i$	ν_{id}	0	0	0	0			$)_{\mathbb{H}^*}$
	$v = ($	$s + \theta\bar{H} + \theta\bar{H}'$	$-\theta$	$-\theta'$	0	κ	ω	0	0			$)_{\mathbb{H}}$
	$\mathbf{k}^*_{\mathsf{id},t} = ($	δ_{id}	$\pi_{\mathsf{id},t}$	$\pi_{\mathsf{id},t}t$	$\phi_{\mathsf{id},t}$	0	0	0	0	0	0	$)_{\mathbb{D}^*}$
	$S^*_{i,\lambda} = ($	$\beta'_{i,\lambda}$	$\gamma_{i,\lambda}$	$\gamma_{i,\lambda}t_\lambda$	$q_{i,\lambda}$	0	0	0	0	0	0	$)_{\mathbb{D}^*}$
	$c_\lambda = ($	s_λ	$\theta_\lambda t_\lambda$	$-\theta_\lambda$	0	κ_λ	r_λ	0	0	0	0	$)_{\mathbb{D}}$
$\mathbf{G}_{1.\Delta.1}$	Add random δ'_Δ: $\mathsf{SubSpace\text{-}Ind}$ on $(\mathbb{B}^*, \mathbb{B})_{1,4}$ and on $(\mathbb{D}^*, \mathbb{D})_{1,6}$											
	$\mathbf{k}^*_{\Delta,0} = ($	δ_Δ	$\phi_{\Delta,0}$	0	δ'_Δ							$)_{\mathbb{B}^*}$
	$\mathbf{k}^*_{\Delta,t} = ($	δ_Δ	$\pi_{\Delta,t}$	$\pi_{\Delta,t}t$	$\phi_{\Delta,t}$	0	δ'_Δ	0	0	0	0	$)_{\mathbb{D}^*}$
$\mathbf{G}_{1.\Delta.2}$	Add random $\delta'_\Delta z_t$: $\mathsf{SubSpace\text{-}Ind}$ on $(\mathbb{D}^*, \mathbb{D})_{4,8}$											
	$\mathbf{k}^*_{\Delta,t} = ($	δ_Δ	π_Δ	$\pi_\Delta t$	$\phi_{\Delta,t}$	0	δ'_Δ	0	$\delta'_\Delta z_t$	0	0	$)_{\mathbb{D}^*}$
$\mathbf{G}_{1.\Delta.3}$	Hybrid game (see Figure 10)											
	$c_\lambda = ($	s_λ	$\theta_\lambda t_\lambda$	$-\theta_\lambda$	0	κ_λ	0	0	$\frac{r_\lambda}{z_{t_\lambda}}$	0	0	$)_{\mathbb{D}}$
$\mathbf{G}_{1.\Delta.4}$	Policy argument: r_0 unpredictable, then δ''_Δ random											
	$\mathbf{k}^*_{\Delta,0} = ($	δ_Δ	$\phi_{\Delta,0}$	0	δ''_Δ							$)_{\mathbb{B}^*}$
$\mathbf{G}_{1.\Delta.5}$	Undo $\mathbf{G}_{1.\Delta.3}$: Hybrid game											
	$c_\lambda = ($	s_λ	$\theta_\lambda t_\lambda$	$-\theta_\lambda$	0	κ_λ	r_λ	0	0	0	0	$)_{\mathbb{D}}$
$\mathbf{G}_{1.\Delta.6}$	Undo $\mathbf{G}_{1.\Delta.2}$: $\mathsf{SubSpace\text{-}Ind}$ on $(\mathbb{D}^*, \mathbb{D})_{4,8}$											
	$\mathbf{k}^*_{\Delta,t} = ($	δ_Δ	$\pi_{\Delta,t}$	$\pi_{\Delta,t}t$	$\phi_{\Delta,t}$	0	δ'_Δ	0	0	0	0	$)_{\mathbb{D}^*}$
$\mathbf{G}_{1.\Delta.7}$	Undo $\mathbf{G}_{1.\Delta.1}$: $\mathsf{SubSpace\text{-}Ind}$ on $(\mathbb{D}^*, \mathbb{D})_{1,6}$											
	$\mathbf{k}^*_{\Delta,t} = ($	δ_Δ	$\pi_{\Delta,t}$	$\pi_{\Delta,t}t$	$\phi_{\Delta,t}$	0	0	0	0	0	0	$)_{\mathbb{D}^*}$

Fig. 8. Existential Unforgeability: Gap between $\mathbf{G}_1$ and $\mathbf{G}_2$

Game $\mathbf{G}_{1.\Delta.0}$: The state of the game at that point is the following, for the keys

$$\begin{array}{ll} \mathsf{id} < \Delta & \mathbf{k}^*_{\mathsf{id},0} = (\delta_{\mathsf{id}}, \phi_{\mathsf{id},0}, 0, \delta''_{\mathsf{id}})_{\mathbb{B}^*} \\ \mathsf{id} \geq \Delta & \mathbf{k}^*_{\mathsf{id},0} = (\delta_{\mathsf{id}}, \phi_{\mathsf{id},0}, 0, 0)_{\mathbb{B}^*} \\ \forall \mathsf{id}, \forall t & \mathbf{k}^*_{\mathsf{id},t} = (\delta_{\mathsf{id}}, \pi_{\mathsf{id},t}(1,t), \phi_{\mathsf{id},t}, 0, 0, 0, 0, 0, 0)_{\mathbb{D}^*} \end{array}$$

for the signatures,

$$U^*_i = (\delta_i, \zeta_i, 0, 0)_{\mathbb{B}^*} \qquad S^*_{i,\lambda} = (\beta'_{i,\lambda}, \gamma_{i,\lambda}(1, t_\lambda), q_{i,\lambda}, 0, 0, 0, 0, 0, 0)_{\mathbb{D}^*}$$

and the verification vectors

$$u = (-s_0 - s, 0, \kappa_0, -r_0)_{\mathbb{B}} \qquad c_\lambda = (s_\lambda, \theta_\lambda t_\lambda, -\theta_\lambda, 0, \kappa_\lambda, r_\lambda, 0, 0, 0, 0)_{\mathbb{D}}$$

Game $\mathbf{G}_{1.\Delta.1}$: We change the Δ-th key into, for a random $\delta'_\Delta \xleftarrow{\$} \mathbb{Z}_q$

$$\mathbf{k}^*_{\Delta,0} = (\delta_\Delta, \phi_{\Delta,0}, 0, \delta'_\Delta)_{\mathbb{B}^*} \qquad \mathbf{k}^*_{\Delta,t} = (\delta_\Delta, \pi_{\Delta,t}(1,t), \phi_{\Delta,t}, 0, \delta'_\Delta, 0, 0, 0, 0)_{\mathbb{D}^*}$$

Other vectors are not modified.

The previous game and this game are indistinguishable under the DDH assumption in $\mathbb{G}_2$: one applies the $\mathsf{SubSpace\text{-}Ind}$ property, on $(\mathbb{B}^*, \mathbb{B})_{2,4}$ and $(\mathbb{D}^*, \mathbb{D})_{1,6}$. Indeed, we can consider a triple $(a \cdot G_2, b \cdot G_2, c \cdot G_2)$, where $c = ab + \tau \bmod q$ with either $\tau = 0$ or $\tau = \delta'_\Delta \xleftarrow{\$} \mathbb{Z}_q$, which are indistinguishable under the DDH assumption in $\mathbb{G}_2$.

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

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

The vectors $\mathbf{b}_4$ and $\mathbf{d}_6$ can not be computed, but they are hidden from the adversary's view. The Δ-th key is now computed as

$$\begin{aligned} \mathbf{k}^*_{\Delta,0} &= (\delta_\Delta, \phi_{\Delta,0}, 0, 0)_{\mathbb{B}^*} + (b, 0, 0, c)_{\mathbb{U}^*} = (\delta_\Delta + b, \phi_{\Delta,0}, 0, \tau)_{\mathbb{B}^*} \\ \mathbf{k}^*_{\Delta,t} &= (\delta_\Delta, \pi_\Delta(1,t), \phi_{\Delta,t}, 0, 0, 0, 0, 0, 0)_{\mathbb{D}^*} + (b, 0, 0, 0, 0, c, 0, 0, 0, 0)_{\mathbb{V}^*} \\ &= (\delta_\Delta + b, \pi_{\Delta,t}(1,t), \phi_{\Delta,t}, 0, \tau, 0, 0, 0, 0)_{\mathbb{D}^*} \end{aligned}$$

When $\tau = 0$, we are in the previous game, meanwhile when $\tau = \delta'_\Delta \xleftarrow{\$} \mathbb{Z}_q$ we are in the new game. In both cases, the random value δ_Δ are replaced by $\delta_\Delta + b$ that follow the same distribution. We can also update $\mathbf{r}^*_{\Delta,1}, \mathbf{r}^*_{\Delta,2}$ and $\mathbf{r}^*_{\Delta,3}$ as $b \cdot G_2$ is given.

Since $\mathbf{b}_4$ and $\mathbf{d}_6$ cannot be computed, one has to generate the verification vectors in the original bases:

$$\begin{aligned} u &= (-s_0 - s, 0, \kappa_0, -r_0)_{\mathbb{U}} \\ &= (-s_0 - ar_0 - s, 0, \kappa_0, -r_0)_{\mathbb{B}} = (-s'_0 - s, 0, \kappa_0, -r_0)_{\mathbb{B}} \\ c_\lambda &= (s_\lambda, \theta_\lambda t_\lambda, -\theta_\lambda, 0, \kappa_\lambda, r_\lambda, 0, 0, 0, 0)_{\mathbb{V}} \\ &= (s_\lambda + ar_\lambda, \theta_\lambda t_\lambda, -\theta_\lambda, 0, \kappa_\lambda, r_\lambda, 0, 0, 0, 0)_{\mathbb{D}} = (s'_\lambda, \theta_\lambda t_\lambda, -\theta_\lambda, 0, \kappa_\lambda, r_\lambda, 0, 0, 0, 0)_{\mathbb{D}} \end{aligned}$$

where $s'_0 = s_0 + ar_0$ and $s'_\lambda = s_\lambda + ar_\lambda$. Since $(r_\lambda)_\lambda$ and $(s_\lambda)_\lambda$ are random r_0 and s_0-labeling (respectively), then any linear combination $(s'_\lambda = s_\lambda + ar_\lambda)_\lambda$ is a random $s'_0 = s_0 + ar_0$-labeling. Hence, $\mathsf{Adv}_{1.\Delta.0} - \mathsf{Adv}_{1.\Delta.1} \leq \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t)$.

Game $\mathbf{G}_{1.\varDelta.2}$: We again change the keys into

$$\mathbf{k}^*_{\varDelta,t} = (\delta_\varDelta, \pi_{\varDelta,t}(1,t), \phi_{\varDelta,t}, 0, \delta'_\varDelta, 0, \delta'_\varDelta z_t, 0, 0)_{\mathbb{D}^*}$$

for random scalars $z_t \xleftarrow{\$} \mathbb{Z}_q$.
The previous game and this game are indistinguishable under the DDH assumption in $\mathbb{G}_2$: one applies the $\mathsf{SubSpace\text{-}Ind}$ property on $(\mathbb{D}^*, \mathbb{D})_{4,8}$. Indeed, we can consider a triple $(a \cdot G_2, b \cdot G_2, c \cdot G_2)$, where $c = ab + \tau \bmod q$ with either $\tau = 0$ or $\tau = 1$, which are indistinguishable under the DSDH assumption in bG_2.
One chooses additional scalars $\alpha_t = \delta'_\varDelta z_t$ and $\beta_t \xleftarrow{\$} \mathbb{Z}_q$ to virtually set $b_t = \alpha_t \cdot b + \beta_t$ and $c_t = \alpha_t \cdot c + \beta_t \cdot a$, which makes $c_t - ab_t = \alpha_t \cdot \tau = \delta'_\varDelta z_t \cdot \tau$.
Let us assume we start from random dual orthogonal bases $(\mathbb{V}, \mathbb{V}^*)$. Then we define the matrices

$$D = \begin{pmatrix} 1 & 0 \\ -a & 1 \end{pmatrix}_{4,8} \qquad D' = \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}_{4,8}$$

$$\mathbb{D} = D \cdot \mathbb{V} \qquad \mathbb{D}^* = D' \cdot \mathbb{V}^*$$

The vector $\mathbf{d}_7$ can not be computed, but it is hidden from the adversary's view. The $\varDelta$-th key is now computed as

$$\begin{aligned}
\mathbf{k}^*_{\varDelta,t} &= (\delta_\varDelta, \pi_{\varDelta,t}(1,t), \phi_{\varDelta,t}, 0, \delta'_\varDelta, 0, 0, 0, 0)_{\mathbb{D}^*} + (0, 0, 0, b_t, 0, 0, 0, c_t, 0, 0)_{\mathbb{V}^*} \\
&= (\delta_\varDelta, \pi_{\varDelta,t}(1,t), \phi_{\varDelta,t} + b_t, 0, \delta'_\varDelta, 0, \alpha_t\tau, 0, 0)_{\mathbb{D}^*} \\
&= (\delta_\varDelta, \pi_{\varDelta,t}(1,t), \phi'_{\varDelta,t}, 0, \delta'_\varDelta, 0, \delta'_\varDelta z_t \cdot \tau, 0, 0)_{\mathbb{D}^*}
\end{aligned}$$

When $\tau = 0$, we are in the previous game, meanwhile when $\tau = 1$ we are in the new game.
Vectors in $\mathbb{D}$ are not modified, and are directly simulated in $\mathbb{D}$, as their components are 0 on the 7-th coordinate. Hence, $\mathsf{Adv}_{1.\varDelta.1} - \mathsf{Adv}_{1.\varDelta.2} \leq \mathsf{Adv}^{\mathsf{dsdh}}_{\mathbb{G}_2}(t)$.

Game $\mathbf{G}_{1.\varDelta.3}$: We hide the shares r_λ in every verification vectors, in front of the value $\delta'_\varDelta$ that was just introduced in the keys. In order to do this, we proceed with an hybrid game on the attribute indices, modifying them one t at a time, using the first time the attribute t appears in the game. We will denote the current attribute by p, also identified to its index in the order of appearance. We transform the verification vectors for all λ such that $t_\lambda = p$ into

$$c_\lambda = (s_\lambda, \theta_\lambda p, -\theta_\lambda, 0, \kappa_\lambda, 0, 0, r_\lambda / z_p)_{\mathbb{D}}$$

When $p = 0$ (for the order of appearance, which means before the first one), this is exactly the game $\mathbf{G}_{1.\varDelta.2}$: $\mathbf{G}_{1.\varDelta.2} = \mathbf{G}_{1.\varDelta.2.0.0}$, whereas for $p = T$ (for the order of appearance, which means the last one) this is exactly the expected game $\mathbf{G}_{1.\varDelta.3}$: $\mathbf{G}_{1.\varDelta.3} = \mathbf{G}_{1.\varDelta.2.T.0}$.
From Appendix C.6, for each p, we prove that

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

Hence, globally, we have

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

Game $\mathbf{G}_{1.\varDelta.4}$: First, one can note that the scalars z_t used in the verification vectors and in the $\varDelta$-th key mask the r_λ in the verification vectors. Since attributes in the $\varDelta$-th key do not satisfy the policy $\mathcal{T}'$ of the forgery, not enough r_λ can be known (others are perfectly private), and then r_0 is perfectly unpredictable, it can be replaced by a random value r'_0, in an intermediate game.

Then, we proceed with a formal change of basis $\mathbb{B}$. Let us assume we start from random dual orthogonal bases $(\mathbb{U}, \mathbb{U}^*)$. Then we define the matrices, with $\theta = r'_0/r_0$, for a random $r'_0 \overset{\$}{\leftarrow} \mathbb{Z}_q^*$

$$\begin{aligned} B &= \big(\theta\big)_4 & \qquad B' &= \big(1/\theta\big)_4 \\ \mathbb{B} &= B \cdot \mathbb{U} & \qquad \mathbb{B}^* &= B' \cdot \mathbb{U}^* \end{aligned}$$

which modifies only the hidden basis vectors $\mathbf{b}_4, \mathbf{b}_4^*$. Since they are not in the adversary's view, the advantage is not modified: $\mathsf{Adv}_{1.\varDelta.4} = \mathsf{Adv}_{1.\varDelta.3}$. Then,

$$\begin{aligned} \mathsf{id} < \varDelta \qquad & \mathbf{k}^*_{\mathsf{id},0} = (\delta_{\mathsf{id}}, \phi_{\mathsf{id},0}, 0, \delta''_{\mathsf{id}})_{\mathbb{U}^*} = (\delta_{\mathsf{id}}, \phi_{\mathsf{id},0}, 0, \theta\delta''_{\mathsf{id}})_{\mathbb{B}^*} \\ & \mathbf{k}^*_{\varDelta,0} = (\delta_{\varDelta}, \phi_{\varDelta,0}, 0, \delta'_{\varDelta})_{\mathbb{U}^*} = (\delta_{\varDelta}, \phi_{\varDelta,0}, 0, \theta\delta'_{\varDelta})_{\mathbb{B}^*} \\ \mathsf{id} > \varDelta \qquad & \mathbf{k}^*_{\mathsf{id},0} = (\delta_{\mathsf{id}}, \phi_{\mathsf{id},0}, 0, 0)_{\mathbb{U}^*} = (\delta_{\mathsf{id}}, \phi_{\mathsf{id},0}, 0, 0)_{\mathbb{B}^*} \\ & u = (-s_0 - s, 0, \kappa_0, -r'_0)_{\mathbb{U}} = (-s_0 - s, 0, \kappa_0, -r'_0/\theta)_{\mathbb{U}} \\ & \quad = (-s_0 - s, 0, \kappa_0, -r_0)_{\mathbb{U}} \end{aligned}$$

Definitions in $(\mathbb{U}, \mathbb{U}^*)$ are the above intermediate game, and definitions in $(\mathbb{D}, \mathbb{D}^*)$ correspond to the new game as for $\mathsf{id} < \varDelta$, δ''_{id} are already random values, then $\theta\delta''_{\mathsf{id}}$ is also uniformly random (whatever independent θ is); and since r'_0 is random, $\theta\delta'_{\varDelta} = r'_0\delta'_{\varDelta}/r_0$ is uniformly random, and independent.

Game $\mathbf{G}_{1.\varDelta.5}$: In this game, we undo $\mathbf{G}_{1.\varDelta.3}$. Then, as above, $\mathsf{Adv}_{1.\varDelta.4} - \mathsf{Adv}_{1.\varDelta.5} \leq 3T \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t) + 2T^2 \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t)$.

Game $\mathbf{G}_{1.\varDelta.6}$: In this game, we undo $\mathbf{G}_{1.\varDelta.2}$. Then $\mathsf{Adv}_{1.\varDelta.5} - \mathsf{Adv}_{1.\varDelta.6} \leq \mathsf{Adv}^{\mathsf{dsdh}}_{\mathbb{G}_2}(t)$.

Game $\mathbf{G}_{1.\varDelta.7}$: In this game, we undo $\mathbf{G}_{1.\varDelta.1}$. Then $\mathsf{Adv}_{1.\varDelta.6} - \mathsf{Adv}_{1.\varDelta.7} \leq \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t)$.

The hybrid on $\varDelta$ is over, as one can see: $G_{1.\varDelta+1.0} = G_{1.\varDelta.7}$. We can now proceed on the hybrid on $\varDelta + 1$, until $\varDelta = K$.

C.5 Existential Unforgeability: Gap between $\mathbf{G}_3$ and $\mathbf{G}_4$

In this sequence, we will stop tracking elements on bases $\mathbb{D}, \mathbb{D}^*$ as they are not modified. See Figure 9.

$\mathbf{G}_{3.j.0}$	Hybrid sequence from $\mathbf{G}_3$ to $\mathbf{G}_4$										
$i \geq j$	$U_i^* = ($	$\xi_i\delta_i$	ζ_i	0	0					$)_{\mathbb{B}^*}$	
$i < j$	$U_i^* = ($	$\xi_i\delta_i$	ζ_i	0	ρ_i					$)_{\mathbb{B}^*}$	
	$u = ($	$-s_0 - s$	0	κ_0	$-r_0$					$)_{\mathbb{B}}$	
$i \geq j$	$V_i^* = ($	$\xi_i\delta_i$	$\xi_i\delta_i H_i$	$\xi_i\delta_i H'_i$	ν_i	0	0	0	0	$)_{\mathbb{H}^*}$	
$i < j$	$V_i^* = ($	$\xi_i\delta_i$	$\xi_i\delta_i H_i$	$\xi_i\delta_i H'_i$	ν_i	0	τ_i	0	0	$)_{\mathbb{H}^*}$	
	$v = ($	$s+\theta\bar{H}+\theta'\bar{H}'$	$-\theta$	$-\theta'$	0	κ	ω	0	0	$)_{\mathbb{H}}$	
$\mathbf{G}_{3.j.1}$	Add random ρ_j: $\mathsf{SubSpace\text{-}Ind}$ on $(\mathbb{B}^*, \mathbb{B})_{1,4}$ and on $(\mathbb{H}^*, \mathbb{H})_{1,6}$										
	$U_j^* = ($	$\xi_j\delta_j$	ζ_j	0	ρ_j					$)_{\mathbb{B}^*}$	
	$V_j^* = ($	$\xi_j\delta_j$	$\xi_j\delta_j H_j$	$\xi_j\delta_j H'_j$	ν_j	0	$\frac{\rho_j r_0}{\omega}$	0	0	$)_{\mathbb{H}^*}$	
$\mathbf{G}_{3.j.2}$	Randomize $\rho_j r_0/\omega$: $\mathsf{Index\text{-}Ind}$ on $(\mathbb{H}^*, \mathbb{H})_{1,2,3,6,7,8}$										
	$U_j^* = ($	$\xi_j\delta_j$	ζ_j	0	ρ_j					$)_{\mathbb{B}^*}$	
	$V_j^* = ($	$\xi_j\delta_j$	$\xi_j\delta_j H_j$	$\xi_j\delta_j H'_j$	ν_j	0	τ_j	0	0	$)_{\mathbb{H}^*}$	

Fig. 9. Existential Unforgeability: Gap between $\mathbf{G}_3$ and $\mathbf{G}_4$

Game $\mathbf{G}_{3.j.0}$: The state of the game at that point is the following, for the keys, the signatures, and the verification vectors

$$\begin{array}{llll}
 & \mathbf{k}^*_{\mathsf{id},0} = (\delta_{\mathsf{id}}, \phi_{\mathsf{id},0}, 0, \delta''_{\mathsf{id}})_{\mathbb{B}^*} & & \mathbf{r}^*_{\mathsf{id},1} = (\delta_{\mathsf{id}}, 0, 0, \psi_{\mathsf{id},1}, 0, 0, 0, 0)_{\mathbb{H}^*} \\
 & & & \mathbf{r}^*_{\mathsf{id},2} = (0, \delta_{\mathsf{id}}, 0, \psi_{\mathsf{id},2}, 0, 0, 0, 0)_{\mathbb{H}^*} \\
 & & & \mathbf{r}^*_{\mathsf{id},3} = (0, 0, \delta_{\mathsf{id}}, \psi_{\mathsf{id},3}, 0, 0, 0, 0)_{\mathbb{H}^*} \\
i \geq j & U^*_i = (\xi_i\delta_i, \zeta_i, 0, 0)_{\mathbb{B}^*} & & V^*_i = (\xi_i\delta_i(1, H_i, H'_i), \nu_i, 0, 0, 0, 0)_{\mathbb{H}^*} \\
i < j & U^*_i = (\xi_i\delta_i, \zeta_i, 0, \rho_i)_{\mathbb{B}^*} & & V^*_i = (\xi_i\delta_i(1, H_i, H'_i), \nu_i, 0, \tau_i, 0, 0)_{\mathbb{H}^*} \\
 & u = (-s_0 - s, 0, \kappa_0, -r_0)_{\mathbb{B}} & & v = (s + \theta\bar{H} + \theta'\bar{H}', -\theta, -\theta', 0, \kappa, \\
 & & & \qquad \omega, 0, 0)_{\mathbb{H}}
\end{array}$$

Game $\mathbf{G}_{3.j.1}$: We change the j-th signature into:

$$U^*_j = (\xi_j\delta_j, \zeta_j, 0, \rho_j)_{\mathbb{B}^*} \qquad V^*_j = (\xi_j\delta_j(1, H_j, H'_j), \nu_j, 0, \rho_j \cdot r_0/\omega, 0, 0)_{\mathbb{H}^*}$$

with a random $\rho_j \xleftarrow{\$} \mathbb{Z}_q$. The previous game and this game are indistinguishable under the DDH assumption in $\mathbb{G}_2$: one applies the $\mathsf{SubSpace\text{-}Ind}$ property, on $(\mathbb{B}, \mathbb{B}^*)_{1,4}$ and $(\mathbb{H}, \mathbb{H}^*)_{1,6}$. Indeed, we can consider a triple $(a{\cdot}G_2, b{\cdot}G_2, c{\cdot}G_2)$, where $c = ab{+}\tau \bmod q$ with either $\tau = 0$ or random, which are indistinguishable situations under the DDH assumption in $\mathbb{G}_2$. One notes that we can virtually set $a' = r_0/\omega \cdot a$ and $c' = r_0/\omega \cdot c$, which makes $c' - a'b = r_0/\omega \cdot \tau$. Let us assume we start from random dual orthogonal bases $(\mathbb{U}, \mathbb{U}^*)$, $(\mathbb{W}, \mathbb{W}^*)$. Then we define the matrices

$$\begin{array}{llll}
B' = \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}_{1,4} & B = \begin{pmatrix} 1 & 0 \\ -a & 1 \end{pmatrix}_{1,4} & H' = \begin{pmatrix} 1 & a' \\ 0 & 1 \end{pmatrix}_{1,6} & H = \begin{pmatrix} 1 & 0 \\ -a' & 1 \end{pmatrix}_{1,6} \\
\mathbb{B}^* = B' \cdot \mathbb{U}^* & \mathbb{B} = B \cdot \mathbb{U} & \mathbb{H}^* = H' \cdot \mathbb{W}^* & \mathbb{H} = H \cdot \mathbb{W}
\end{array}$$

The vectors $\mathbf{b}_4$ and $\mathbf{h}_6$ can not be computed, but they are hidden from the adversary's view. The j-th signature is now computed as:

$$\begin{aligned}
U^*_j &= (b, \zeta_j, 0, c)_{\mathbb{U}^*} = (b, \zeta_j, 0, \tau)_{\mathbb{B}^*} \\
V^*_j &= (b(1, H_j, H'_j), \nu_j, 0, c', 0, 0)_{\mathbb{W}^*} = (b(1, H_j, H'_j), \nu_j, 0, r_0/\omega \cdot \tau, 0, 0)_{\mathbb{H}^*}
\end{aligned}$$

This is the expected signature, with $\xi_j = b/\delta_j$. Since $b \cdot G_2$ is known, we can use it to simulate $S^*_{j,\lambda}$. When $\tau = 0$, we are in the previous game, meanwhile when $\tau = \rho_j$ is random, we are in the new game.
Since the vectors $\mathbf{b}_4$ and $\mathbf{h}_6$ can not be computed, we cannot define the verification vectors in the new bases:

$$\begin{aligned}
u &= (-s_0 - s, 0, \kappa_0, -r_0)_{\mathbb{U}} = (-s_0 - s - ar_0, 0, \kappa_0, -r_0)_{\mathbb{B}} \\
&= (-s_0 - s', 0, \kappa_0, -r_0)_{\mathbb{B}} \\
v &= (s + \theta\bar{H} + \theta'\bar{H}', -\theta, -\theta', 0, \kappa, \omega, 0, 0)_{\mathbb{W}} \\
&= (s + a'\omega + \theta\bar{H} + \theta'\bar{H}', -\theta, -\theta', 0, \kappa, \omega, 0, 0)_{\mathbb{H}} \\
&= (s + ar_0 + \theta\bar{H} + \theta'\bar{H}', -\theta, -\theta', 0, \kappa, \omega, 0, 0)_{\mathbb{H}} \\
&= (s' + \theta\bar{H} + \theta'\bar{H}', -\theta, -\theta', 0, \kappa, \omega, 0, 0)_{\mathbb{H}}
\end{aligned}$$

They remain consistent, as one can simply replace the random s by $s' = s + ar_0$. Hence, $\mathsf{Adv}_{3.j.0} - \mathsf{Adv}_{3.j.1} \leq \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t)$.

Game $\mathbf{G}_{3.j.2}$: We change again the j-th signature into:

$$U^*_j = (\xi_j\delta_j, \zeta_j, 0, \rho_j)_{\mathbb{B}^*} \qquad V^*_j = (\xi_j\delta_j(1, H_j, H'_j), \nu_j, 0, \tau_j, 0, 0)_{\mathbb{H}^*}$$

with random and independent $\rho_j, \tau_j \xleftarrow{\$} \mathbb{Z}_q$.
To do this, we use the $\mathsf{Index\text{-}Ind}$ Property from Theorem 2 on $(\mathbb{H}, \mathbb{H}^*)$ on the 6 coordinates 1,2,3,6,7,8, of dimension 6 from the view $(\mathbf{h}_1^*, \mathbf{h}_2^*, \mathbf{h}_3^*, \mathbf{h}_6^*, \mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3)$ with hidden vectors $(\mathbf{h}_7^*, \mathbf{h}_8^*, \mathbf{h}_6, \mathbf{h}_7, \mathbf{h}_8)$, with

$$\begin{aligned} v &= (s + \theta\bar{H} + \theta'\bar{H}', -\theta, -\theta', 0, \kappa, \omega, 0, 0)_{\mathbb{H}} \\ &= (s + \theta(\bar{H} + \rho\bar{H}'), -\theta, -\rho\theta, 0, \kappa, \omega, 0, 0)_{\mathbb{H}} \\ &= (s, 0, 0, 0, \kappa, 0, 0, 0)_{\mathbb{H}} + (\theta(\bar{H} + \rho\bar{H}', -1, -\rho), 0, 0, \omega, 0, 0)_{\mathbb{H}} \\ V_j^* &= (\xi_j\delta_j(1, H_j, H_j'), \nu_j, 0, \tau_j, 0, 0)_{\mathbb{H}^*} \\ &= (0, 0, 0, \nu_j, 0, 0, 0, 0)_{\mathbb{H}^*} + (\xi_j\delta_j(1, H_j, H_j'), 0, 0, \tau_j, 0, 0)_{\mathbb{H}^*} \end{aligned}$$

with $\rho = \theta'/\theta$, where ρ needs to be decided before the start of the game, as θ, θ' can be too. Under the collision resistance of the hash functions, we can assume that $\bar{H} \neq H_j$ and $\bar{H}' \neq H_j'$. Then, one cannot distinguish between the two following

$$\begin{aligned} V_j^* &= (\xi_j\delta_j(1, H_j, H_j'), \nu_j, 0, r_0/\omega \cdot \tau, 0, 0)_{\mathbb{H}^*} \\ V_j^* &= (\xi_j\delta_j(1, H_j, H_j'), \nu_j, 0, \tau_j, 0, 0)_{\mathbb{H}^*} \end{aligned}$$

with random τ_j, and known random $\xi_j\delta_j \cdot G_2$, as the latter can either be chosen or is the $b \cdot G_2$ from the DDH instances. Hence, $\mathsf{Adv}_{3.j.1} - \mathsf{Adv}_{3.j.2} \leq 4 \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t) + 2 \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t) + 1/q$.

C.6 Existential Unforgeability: Hybrid Sequence

In this sequence, we only follow elements in bases $\mathbb{D}, \mathbb{D}^*$ as other vectors are not modified. See Figure 10.

Game $\mathbf{G}_{1.\Delta.2.p.0}$: The state of the game at that point is the following, for the keys

$$\begin{aligned} \mathbf{k}^*_{\Delta,t} &= (\delta_\Delta, \pi_{\Delta,t}(1, t), \phi_{\Delta,t}, 0, \delta'_\Delta, 0, \delta'_\Delta z_t, 0, 0)_{\mathbb{D}^*} \\ \mathbf{k}^*_{\mathsf{id},t} &= (\delta_{\mathsf{id}}, \pi_{\mathsf{id},t}(1, t), \phi_{\mathsf{id},t}, 0, 0, 0, 0, 0, 0)_{\mathbb{D}^*} \end{aligned}$$

the signatures,

$$S^*_{i,\lambda} = (\beta'_{i,\lambda}, \gamma_{i,\lambda}(1, t_\lambda), q_{i,\lambda}, 0, 0, 0, 0, 0, 0)_{\mathbb{D}^*}$$

and ciphertexts

$$\begin{aligned} c_\lambda &= (s_\lambda, \theta_\lambda t_\lambda, -\theta_\lambda, 0, \kappa_\lambda, r_\lambda, 0, 0, 0, 0)_{\mathbb{D}} & t > p \\ c_\lambda &= (s_\lambda, \theta_\lambda t_\lambda, -\theta_\lambda, 0, \kappa_\lambda, 0, 0, r_\lambda/z_t, 0, 0)_{\mathbb{D}} & t \leq p \end{aligned}$$

Game $\mathbf{G}_{1.\Delta.2.p.1}$: We change the keys into:

$$\begin{aligned} \mathbf{k}^*_{\Delta,t} &= (\delta_\Delta, \pi_{\Delta,t}(1, t), \phi_{\Delta,t}, 0, \delta'_\Delta, \delta'_\Delta, \delta'_\Delta z_t, 0, 0)_{\mathbb{D}^*} \\ \mathbf{k}^*_{\mathsf{id},t} &= (\delta_{\mathsf{id}}, \pi_{\mathsf{id},t}(1, t), \phi_{\mathsf{id},t}, 0, 0, 0, 0, 0, 0)_{\mathbb{D}^*} \end{aligned}$$

To do this, we define the matrices

$$D' = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}_{6,7} \qquad D = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}_{6,7}$$

which modifies the hidden vectors $\mathbf{d}_7, \mathbf{d}_6^*$. Since they are not in the adversary's view, the advantage is perfect.

$\mathbf{G}_{1.\Delta.2.p.0}$	Hybrid sequence from $\mathbf{G}_{1.\Delta.3}$ to $\mathbf{G}_{1.\Delta.4}$											
$\mathsf{id} > \Delta$	$\mathbf{k}^*_{\mathsf{id},0} = ($	δ_{id}	$\phi_{\mathsf{id},0}$	0	0							$)_{\mathbb{B}^*}$
	$\mathbf{k}^*_{\Delta,0} = ($	δ_Δ	$\phi_{\Delta,0}$	0	δ'_Δ							$)_{\mathbb{B}^*}$
$\mathsf{id} < \Delta$	$\mathbf{k}^*_{\mathsf{id},0} = ($	δ_{id}	$\phi_{\mathsf{id},0}$	0	δ''_{id}							$)_{\mathbb{B}^*}$
	$U^*_i = ($	$\xi_i\delta_i$	ζ_i	0	0							$)_{\mathbb{B}^*}$
	$u = ($	$-s_0 - s$	0	κ_0	$-r_0$							$)_{\mathbb{B}}$
	$\mathbf{r}^*_{\mathsf{id},1} = ($	δ_{id}	0	0	$\psi_{\mathsf{id},1}$	0	0	0	0			$)_{\mathbb{H}^*}$
	$\mathbf{r}^*_{\mathsf{id},2} = ($	0	δ_{id}	0	$\psi_{\mathsf{id},2}$	0	0	0	0			$)_{\mathbb{H}^*}$
	$\mathbf{r}^*_{\mathsf{id},3} = ($	0	0	δ_{id}	$\psi_{\mathsf{id},3}$	0	0	0	0			$)_{\mathbb{H}^*}$
	$V^*_i = ($	$\xi_i\delta_i$	$\xi_i\delta_i H_i$	$\xi_i\delta_i H_i$	ν_{id}	0	0	0	0			$)_{\mathbb{H}^*}$
	$v = ($	$s+\theta\bar{H}+\theta'\bar{H}'$	$-\theta$	$-\theta'$	0	0	ω	0	0			$)_{\mathbb{H}}$
$\mathsf{id} \neq \Delta$	$\mathbf{k}^*_{\mathsf{id},t} = ($	δ_{id}	$\pi_{\mathsf{id},t}$	$\pi_{\mathsf{id},t}t$	$\phi_{\mathsf{id},t}$	0	0	0	0	0	0	$)_{\mathbb{D}^*}$
	$\mathbf{k}^*_{\Delta,t} = ($	δ_Δ	$\pi_{\Delta,t}$	$\pi_{\Delta,t}t$	$\phi_{\Delta,t}$	0	δ'_Δ	0	$\delta'_\Delta z_t$	0	0	$)_{\mathbb{D}^*}$
	$S^*_{i,\lambda} = ($	$\beta'_{i,\lambda}$	$\gamma_{i,\lambda}$	$\gamma_{i,\lambda}t_\lambda$	$q_{i,\lambda}$	0	0	0	0	0	0	$)_{\mathbb{D}^*}$
$t_\lambda \geq p$	$c_\lambda = ($	s_λ	$\theta_\lambda t_\lambda$	$-\theta_\lambda$	0	κ_λ	r_λ	0	0	0	0	$)_{\mathbb{D}}$
$t_\lambda < p$	$c_\lambda = ($	s_λ	$\theta_\lambda t_\lambda$	$-\theta_\lambda$	0	κ_λ	0	0	$\frac{r_\lambda}{z_t}$	0	0	$)_{\mathbb{D}}$
$\mathbf{G}_{1.\Delta.2.p.1}$	Duplicate δ'_Δ in $\mathbb{D}^*$: Formal change of basis on $(\mathbb{D}^*, \mathbb{D})_{6,7}$											
	$\mathbf{k}^*_{\Delta,t} = ($	δ_Δ	$\pi_{\Delta,t}$	$\pi_{\Delta,t}t$	$\phi_{\Delta,t}$	0	δ'_Δ	δ'_Δ	$\delta'_\Delta z_t$	0	0	$)_{\mathbb{D}^*}$
$\mathbf{G}_{1.\Delta.2.p.2}$	Swap r_λ: Swap-Ind on $(\mathbb{D}, \mathbb{D}^*)_{5,6,7}$											
$t_\lambda = p$	$c_\lambda = ($	s_λ	$\theta_\lambda p$	$-\theta_\lambda$	0	κ_λ	0	r_λ	0	0	0	$)_{\mathbb{D}}$
$\mathbf{G}_{1.\Delta.2.p.3}$	Index-Ind for all $t \neq p$ on $(\mathbb{D}^*, \mathbb{D})_{2,3,7,9,10}$											
$t \neq p$	$\mathbf{k}^*_{\Delta,t} = ($	δ_Δ	$\pi_{\Delta,t}$	$\pi_{\Delta,t}t$	$\phi_{\Delta,t}$	0	δ'_Δ	$\frac{\delta'_\Delta z_t}{z_p}$	$\delta'_\Delta z_t$	0	0	$)_{\mathbb{D}^*}$
$\mathbf{G}_{1.\Delta.2.p.4}$	Remove α: Formal change of basis on $(\mathbb{D}^*, \mathbb{D})_{7,8}$											
	$\mathbf{k}^*_{\Delta,t} = ($	δ_Δ	π_Δ	$\pi_\Delta t$	$\phi_{\Delta,t}$	0	δ'_Δ	0	$\delta'_\Delta z_t$	0	0	$)_{\mathbb{D}^*}$
$t_\lambda = p$	$c_\lambda = ($	s_λ	$\theta_\lambda p$	$-\theta_\lambda$	0	κ_λ	0	α	$\frac{r_\lambda}{z_p}$	0	0	$)_{\mathbb{D}}$
$\mathbf{G}_{1.\Delta.2.p.5}$	SubSpace-Ind on $(\mathbb{D}, \mathbb{D}^*)_{5,7}$											
$t_\lambda = p$	$c_\lambda = ($	s_λ	$\theta_\lambda p$	$-\theta_\lambda$	0	κ_λ	0	0	$\frac{r_\lambda}{z_p}$	0	0	$)_{\mathbb{D}}$

Fig. 10. Existential Unforgeability: Hybrid Sequence

The keys are modified in the following way. Note that keys other than $\varDelta$ and signatures are unmodified as they all have a 0 in the 6-th position.

$$\begin{aligned}\mathbf{k}^*_{\varDelta,t} &= (\delta_\varDelta, \pi_{\varDelta,t}(1,t), \phi_{\varDelta,t}, 0, \delta'_\varDelta, 0, \delta'_\varDelta z_t, 0, 0)_{\mathbb{V}^*}\\ &= (\delta_\varDelta, \pi_{\varDelta,t}(1,t), \phi_{\varDelta,t}, 0, \delta'_\varDelta, \delta'_\varDelta, \delta'_\varDelta z_t, 0, 0)_{\mathbb{D}^*}\end{aligned}$$

Meanwhile, the ciphertexts are not modified because they all have a 0 in the 7-th position. The adversary gains no advantage in this game: $\mathsf{Adv}_{1.\varDelta.2.p.0} = \mathsf{Adv}_{1.\varDelta.2.p.1}$.

Game $\mathbf{G}_{1.\varDelta.2.p.2}$: We change only the verification texts linked to the p-th attribute into:

$$t_\lambda = p, \quad c_\lambda = (s_\lambda, \theta_\lambda p, -\theta_\lambda, 0, \kappa_\lambda, 0, r_\lambda, 0, 0, 0)_{\mathbb{D}}$$

The previous game and this game are indistinguishable under the DSDH assumption in $\mathbb{G}_1$: one applies the $\mathsf{Swap}\text{-}\mathsf{Ind}$ property, on $(\mathbb{D}, \mathbb{D}^*)_{5,6,7}$. Indeed, we can consider a triple $(a \cdot G_1, b \cdot G_1, d \cdot G_1)$, where $d = ab + \tau \bmod q$ with either $\tau = 0$ or $\tau = 1$, which are indistinguishable situations under the DSDH assumption.

One chooses additional scalars $\alpha_\lambda = -r_\lambda$ and $\beta_\lambda \stackrel{\$}{\leftarrow} \mathbb{Z}_q$ to virtually set $b_\lambda = \alpha_\lambda \cdot b + \beta_\lambda$ and $d_\lambda = \alpha_\lambda \cdot d + \beta_\lambda \cdot a$, which makes $d_\lambda - ab_\lambda = \alpha_\lambda \cdot \tau = -r_\lambda \cdot \tau$.

$$D = \begin{pmatrix} 1 & a & -a \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}_{5,6,7} \qquad D' = \begin{pmatrix} 1 & 0 & 0 \\ -a & 1 & 0 \\ a & 0 & 1 \end{pmatrix}_{5,6,7}$$

$$\mathbb{D}^* = D' \cdot \mathbb{V}^* \qquad \mathbb{D} = D \cdot \mathbb{V}$$

The vectors $\mathbf{d}^*_6, \mathbf{d}^*_7$ can not be computed, but they are not in the view of the adversary. The verification texts for the p-th attribute is changed as follows

$$\begin{aligned}c_\lambda &= (s_\lambda, \theta_\lambda p, -\theta_\lambda, 0, 0, r_\lambda, 0, 0, 0, 0)_{\mathbb{D}} + (0, 0, 0, 0, b_\lambda, d_\lambda, -d_\lambda, 0, 0, 0)_{\mathbb{V}}\\ &= (s_\lambda, \theta_\lambda p, -\theta_\lambda, 0, b_\lambda, r_\lambda + \alpha_\lambda \cdot \tau, -\alpha_\lambda \cdot \tau, 0, 0, 0)_{\mathbb{D}}\\ &= (s_\lambda, \theta_\lambda p, -\theta_\lambda, 0, b_\lambda, r_\lambda - r_\lambda \cdot \tau, r_\lambda \cdot \tau, 0, 0, 0)_{\mathbb{D}}\end{aligned}$$

When $\tau = 0$, we are in the previous game, meanwhile when $\tau = 1$, we are in the next game. Other verification texts are generated in $\mathbb{D}$ directly.

Keys and signatures are unchanged because they have the same value on 6th and 7th columns, either 0 or $\delta'_\varDelta$. Notably, those with 0 on these positions are keys different than $\varDelta$, and can be fully generated in $\mathbb{D}^*$.

$$\begin{aligned}\mathbf{k}^*_{\varDelta,t} &= (\delta_\varDelta, \pi_{\varDelta,t}(1,t), \phi_{\varDelta,t}, 0, \delta'_\varDelta, \delta'_\varDelta, \delta'_\varDelta z_t, 0, 0)_{\mathbb{V}^*}\\ &= (\delta_\varDelta, \pi_{\varDelta,t}(1,t), \phi_{\varDelta,t}, a \cdot \delta'_\varDelta - a \cdot \delta'_\varDelta, \delta'_\varDelta, \delta'_\varDelta, \delta'_\varDelta z_t, 0, 0)_{\mathbb{D}^*}\\ &= (\delta_\varDelta, \pi_{\varDelta,t}(1,t), \phi_{\varDelta,t}, 0, \delta'_\varDelta, \delta'_\varDelta, \delta'_\varDelta z_t, 0, 0)_{\mathbb{D}^*}\end{aligned}$$

The advantage of the adversary is: $\mathsf{Adv}_{1.\varDelta.2.p.1} - \mathsf{Adv}_{1.\varDelta.2.p.2} \leq \mathsf{Adv}^{\mathsf{dsdh}}_{\mathbb{G}_1}(t)$.

Game $\mathbf{G}_{1.\varDelta.2.p.3}$: We keep the $\delta'_\varDelta$ value (at the 7-th hidden position) in the key for the p-th attribute only, and replace all values in other keys by $\delta'_\varDelta z_t / z_p$:

$$\begin{aligned}\mathbf{k}^*_{\varDelta,p} &= (\delta_\varDelta, \pi_{\varDelta,p}(1,p), \phi_{\varDelta,p}, 0, \delta'_\varDelta, \delta'_\varDelta, \delta'_\varDelta z_p, 0, 0)_{\mathbb{D}^*}\\ \mathbf{k}^*_{\varDelta,t} &= (\delta_\varDelta, \pi_{\varDelta,t}(1,t), \phi_{\varDelta,t}, 0, \delta'_\varDelta, \delta'_\varDelta z_t / z_p, \delta'_\varDelta z_t, 0, 0)_{\mathbb{D}^*}\\ \mathbf{k}^*_{\mathsf{id},t} &= (\delta_{\mathsf{id}}, \pi_{\mathsf{id},t}(1,t), \phi_{\mathsf{id},t}, 0, 0, 0, 0, 0, 0)_{\mathbb{D}^*}\end{aligned}$$

To show this is possible without impacting the other vectors, we use the $\mathsf{Index}\text{-}\mathsf{Ind}$ property, but in another level of sequence of hybrid games, for $\gamma \in \{1, \ldots, T\} \backslash \{p\}$. We will again enumerate γ in their order of appearance in the security game (whether in key queries, or signature queries).

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

$$\begin{aligned}
\mathbf{k}^*_{\Delta,p} &= (\delta_\Delta, \pi_{\Delta,p}(1,p), \phi_{\Delta,p}, 0, \delta'_\Delta, \delta'_\Delta, \delta'_\Delta z_p, 0, 0)_{\mathbb{D}^*} \\
\mathbf{k}^*_{\Delta,t} &= (\delta_\Delta, \pi_{\Delta,t}(1,t), \phi_{\Delta,t}, 0, \delta'_\Delta, \delta'_\Delta z_t/z_p, \delta'_\Delta z_t, 0, 0)_{\mathbb{D}^*} && p \neq t < \gamma \\
\mathbf{k}^*_{\Delta,t} &= (\delta_\Delta, \pi_{\Delta,t}(1,t), \phi_{\Delta,t}, 0, \delta'_\Delta, \delta'_\Delta, \delta'_\Delta z_t, 0, 0)_{\mathbb{D}^*} && t \geq \gamma \\
c_\lambda &= (s_\lambda, \theta_\lambda t_\lambda, -\theta_\lambda, 0, \kappa_\lambda, 0, 0, r_\lambda/z_t, 0, 0)_{\mathbb{D}} && t < p \\
c_\lambda &= (s_\lambda, \theta_\lambda p, -\theta_\lambda, 0, \kappa_\lambda, 0, r_\lambda, 0, 0, 0)_{\mathbb{D}} \\
c_\lambda &= (s_\lambda, \theta_\lambda t_\lambda, -\theta_\lambda, 0, \kappa_\lambda, r_\lambda, 0, 0, 0, 0)_{\mathbb{D}} && t > p
\end{aligned}$$

When $\gamma = 1$, this is the previous game: $\mathbf{G}_{1.\Delta.2.p.2.1} = \mathbf{G}_{1.\Delta.2.p.2}$, whereas with $\gamma = T+1$, this is the current game: $\mathbf{G}_{1.\Delta.2.p.2.T+1} = \mathbf{G}_{1.\Delta.2.p.3}$.
To do this, we use the Index-Ind Property from Section 2.2 on $(\mathbb{D}, \mathbb{D}^*)$ on the 5 coordinates 2, 3, 6, 9, 10 of dimension 5 from the view $(\mathbf{d}^*_2, \mathbf{d}^*_3, \mathbf{d}^*_7, \mathbf{d}_2, \mathbf{d}_3)$ with hidden vectors $(\mathbf{d}^*_9, \mathbf{d}^*_{10}, \mathbf{d}_7, \mathbf{d}_9, \mathbf{d}_{10})$, with

$$\begin{aligned}
c_\lambda &= (s_\lambda, \theta_\lambda p, -\theta_\lambda, 0, \kappa_\lambda, 0, r_\lambda, 0, 0, 0)_{\mathbb{D}} \\
\mathbf{k}^*_{\Delta,t} &= (\delta_\Delta, \pi_{\Delta,t}(1,t), \phi_{\Delta,t}, 0, \delta'_\Delta, \delta'_\Delta, \delta'_\Delta z_t, 0, 0)_{\mathbb{D}^*} && t = \gamma
\end{aligned}$$

Hence, we have $\mathsf{Adv}_{1.\Delta.2.p.2} - \mathsf{Adv}_{1.\Delta.2.p.3} \leq 4 \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_2}(t) + 2 \times \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t)$

Game $\mathbf{G}_{1.\Delta.2.p.4}$: We use a theoretic information change to uniformize the keys and verification texts. We change the key into:

$$\begin{aligned}
\mathbf{k}^*_{\Delta,t} &= (\delta_\Delta, \pi_{\Delta,t}(1,t), \phi_{\Delta,t}, 0, \delta'_\Delta, 0, \delta'_\Delta z_t, 0, 0)_{\mathbb{D}^*} \\
\mathbf{k}^*_{\Delta,p} &= (\delta_\Delta, \pi_{\Delta,p}(1,p), \phi_{\Delta,p}, 0, \delta'_\Delta, 0, \delta'_\Delta z_p, 0, 0)_{\mathbb{D}^*}
\end{aligned}$$

Meanwhile the verification texts associated to attribute p are changed into:

$$c_\lambda = (s_\lambda, \theta_\lambda p, -\theta_\lambda, 0, \kappa_\lambda, 0, r_\lambda, r_\lambda/z_p, 0, 0)_{\mathbb{D}} \qquad t = p$$

To do this we use the following matrices:

$$D' = \begin{pmatrix} 1 & 0 \\ 1/z_p & 1 \end{pmatrix}_{7,8} \qquad D = \begin{pmatrix} 1 & -1/z_p \\ 0 & 1 \end{pmatrix}_{7,8}$$

The vectors $\mathbf{d}_7, \mathbf{d}^*_7, \mathbf{d}^*_8$ must not be in the adversary's view, but since they are hidden the advantage is perfect.
Keys for Δ are modified in the following way:

$$\begin{aligned}
\mathbf{k}^*_{\Delta,t} &= (\delta_\Delta, \pi_{\Delta,t}(1,t), \phi_{\Delta,t}, 0, \delta'_\Delta, \delta'_\Delta z_t/z_p, \delta'_\Delta z_t, 0, 0)_{\mathbb{V}^*} \\
&= (\delta_\Delta, \pi_{\Delta,t}(1,t), \phi_{\Delta,t}, 0, \delta'_\Delta, 0, \delta'_\Delta z_t, 0, 0)_{\mathbb{D}^*} \\
\mathbf{k}^*_{\Delta,p} &= (\delta_\Delta, \pi_{\Delta,p}(1,p), \phi_{\Delta,p}, 0, \delta'_\Delta, \delta'_\Delta, \delta'_\Delta z_p, 0, 0)_{\mathbb{V}^*} \\
&= (\delta_\Delta, \pi_{\Delta,p}(1,p), \phi_{\Delta,p}, 0, \delta'_\Delta, 0, \delta'_\Delta z_p, 0, 0)_{\mathbb{D}^*}
\end{aligned}$$

Verification texts associated to p are changed as well:

$$\begin{aligned}
c_\lambda &= (s_\lambda, \theta_\lambda p, -\theta_\lambda, 0, \kappa_\lambda, 0, r_\lambda, 0, 0, 0)_{\mathbb{V}} && t = p \\
&= (s_\lambda, \theta_\lambda p, -\theta_\lambda, 0, \kappa_\lambda, 0, r_\lambda, r_\lambda/z_p, 0, 0)_{\mathbb{D}}
\end{aligned}$$

We note that for $t \neq p, c_\lambda$ are unchanged. The same goes for all keys different than Δ and all signatures, as their component on the relevant positions are all 0 (7-th for verification texts, 7-th and 8-th for keys and signatures).
The adversary gains no advantage in this game: $\mathsf{Adv}_{1.\Delta.2.p.3} = \mathsf{Adv}_{1.\Delta.2.p.4}$.

Game $\mathbf{G}_{1.\Delta.2.p.5}$: We remove r_λ from the verification texts associated to the p-th attribute.

$$c_\lambda = (s_\lambda, \theta_\lambda t_\lambda, -\theta_\lambda, 0, \kappa_\lambda, 0, 0, r_\lambda/z_p, 0, 0)_{\mathbb{D}} \qquad\qquad t_\lambda = p$$

The previous game and this game are indistinguishable under the DSDH assumption in $\mathbb{G}_1$: one applies the $\mathsf{SubSpace\text{-}Ind}$ property, on $(\mathbb{D}, \mathbb{D}^*)_{5,7}$. Indeed, we can consider a triple $(a \cdot G_1, b \cdot G_1, d \cdot G_1)$, where $d = ab + \tau \bmod q$ with either $\tau = 0$ or $\tau = 1$, which are indistinguishable situations under the DSDH assumption.
One chooses additional scalar $\beta_\lambda \xleftarrow{\$} \mathbb{Z}_q$ to virtually set $b_\lambda = r_\lambda \cdot b + \beta_\lambda$ and $d_\lambda = r_\lambda \cdot d + \beta_\lambda \cdot a$, which makes $d_\lambda - ab_\lambda = r_\lambda \cdot \tau$.

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

The vector $\mathbf{d}_7^*$ cannot be computed, but they are not in the adversary's view. The verification texts for each λ so that $t_\lambda = p$ are changed in the following way:

$$\begin{aligned} c_\lambda &= (s_\lambda, \theta_\lambda p, -\theta_\lambda, 0, 0, 0, 0, r_\lambda, r_\lambda/z_p, 0, 0)_{\mathbb{D}} + (0, 0, 0, 0, b_\lambda, 0, d_\lambda, 0, 0, 0)_{\mathbb{V}} \\ &= (s_\lambda, \theta_\lambda p, -\theta_\lambda, 0, b_\lambda, 0, r_\lambda \cdot \tau, r_\lambda/z_p, 0, 0)_{\mathbb{D}} \end{aligned}$$

When $\tau = 1$, we are in the previous game, meanwhile when $\tau = 0$, we are in the next game. Other verification texts are unchanged and can be fully generated in $\mathbb{V}$. Keys and signatures are unchanged and can be fully generated in $\mathbb{V}^*$ because they all have a value of 0 on the 7-th position at that point in the hybrid game. The advantage of the adversary is: $\mathsf{Adv}_{1.\Delta.2.p.4} - \mathsf{Adv}_{1.\Delta.2.p.5} \leq \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_1}(t)$.

C.7 Proof of Theorem 14: Traceability

In this proof, we will first recall the way we can simulate the keys and the signatures. Then, we will show how the linearly-homomorphic signature and the Linear-Square problem will prevent attacks:

Game $\mathbf{G}_0$: As shown in the previous section, keys generated by the KeyGen algorithm, for user id, follow the distribution:

$$\begin{aligned} \mathbf{k}^*_{\mathsf{id},0} &= (\delta_{\mathsf{id}}, \phi_{\mathsf{id},0}, 0, 0, \delta_{\mathsf{id}} \cdot w_{\mathsf{id}}, \delta_{\mathsf{id}} \cdot w_{\mathsf{id}}^2)_{\mathbb{B}^*} & \mathbf{r}^*_{\mathsf{id},1} &= (\delta_{\mathsf{id}}, 0, 0, \psi_{\mathsf{id},1}, 0, 0, 0, 0)_{\mathbb{H}^*} \\ \mathbf{k}^*_{\mathsf{id},t} &= (\delta_{\mathsf{id}}, \pi_{\mathsf{id},t}(1, t), \phi_{\mathsf{id},t}, 0, 0, 0, 0, 0, 0)_{\mathbb{D}^*} & \mathbf{r}^*_{\mathsf{id},2} &= (0, \delta_{\mathsf{id}}, 0, \psi_{\mathsf{id},2}, 0, 0, 0, 0)_{\mathbb{H}^*} \\ \Sigma_{\mathsf{id}} &= \mathsf{Sig}'(\mathsf{sk}, \mathbf{k}^*_{\mathsf{id},0}) & \mathbf{r}^*_{\mathsf{id},3} &= (0, 0, \delta_{\mathsf{id}}, \psi_{\mathsf{id},3}, 0, 0, 0, 0)_{\mathbb{H}^*} \end{aligned}$$

The i-th signature generated by the Sig algorithm follows the distribution

$$\begin{aligned} U_i^* &= (\xi_i \delta_i, \zeta_i, 0, 0, \xi_i \delta_i \cdot w_i, \xi_i \delta_i \cdot w_i^2)_{\mathbb{B}^*} \quad V_i^* = (\xi_i \delta_i (1, H_i, H_i'), \nu_i, 0, 0, 0, 0)_{\mathbb{H}^*} \\ S^*_{i,\lambda} &= (\beta'_{i,\lambda}, \gamma_{i,\lambda}(1, t_\lambda), q_{i,\lambda}, 0, 0, 0, 0, 0, 0)_{\mathbb{D}^*} \end{aligned}$$

where $\delta_i, w_i, \mathbf{k}^*_{i,0}, \Sigma_i$ correspond to new and fresh $\delta_{\mathsf{id}}, w_{\mathsf{id}}, \mathbf{k}^*_{\mathsf{id},0}, \Sigma_{\mathsf{id}}$ of the signer id (for an implicitly freshly generated key as signing queries and key queries should not correspond to the same identities, as we are in the distinct-user setting), and $H_i = \mathcal{H}(\mathcal{T}_i), H_i' = \mathcal{H}'(m_i)$, together with

$$\begin{aligned} \Sigma &= \mathsf{DerivSign}'(\mathsf{vk}, ((\xi_i, \mathbf{k}^*_{i,0}, \Sigma_i), (\zeta_i, \mathbf{b}_2^*, \Sigma_2)) \\ \Pi &= \mathsf{NIZKPoK\text{-}SqDH}(w_i, (e(\mathbf{b}_1, U^*), e(\mathbf{b}_5, U^*), e(\mathbf{b}_6, U^*))) \end{aligned}$$

For the decision of the challenge signature $\sigma = (U^*, V^*, (S^*_\lambda)_\lambda, \Sigma, \Pi)$ on message m and policy $\mathcal{T}$, different from any query-answer to the signing oracle, one uses

$$u = (-s_0 - s, 0, \kappa_0, 0, 0, 0)_{\mathbb{B}} \quad v = (s + \theta \cdot \bar{H} + \theta' \cdot \bar{H}', -\theta, -\theta', 0, \kappa, 0, 0, 0)_{\mathbb{H}}$$
$$c_\lambda = (s_\lambda, \theta_\lambda(t_\lambda, -1), 0, \kappa_\lambda, 0, 0, 0, 0, 0)_{\mathbb{D}}$$

where $(\bar{H}, \bar{H}') \neq (H_i, H'_i)$ for all i. Instead of outputting just the decision, one can consider the challenger outputs $(u, v, (c_\lambda)_\lambda)$, and everybody can make the final verification, with $B_1 = e(\mathbf{b}_1, U^*)$, $B_2 = e(\mathbf{b}_5, U^*)$, and $B_3 = e(\mathbf{b}_6, U^*)$:

$$e(\mathbf{b}_1, U^*) \neq 1_{\mathbb{G}_t} \qquad e(u, U^*) \cdot e(v, V^*) \cdot \prod e(c_\lambda, S^*_\lambda) = 1_{\mathbb{G}_t}$$
$$\mathsf{Verif}'(\mathsf{vk}, U^*, \Sigma) = 1 \qquad \mathsf{VERIF\text{-}SqDH}((B_1, B_2, B_3), \Pi) = 1$$

For the tracing procedure, one checks $B_1^{w_{\mathsf{id}}} = B_2$ for all the id's asked to the key generation oracle. If no w_{id} matches, we output 1. Otherwise we output 0. We denote by Adv_0 the probability to output 1 in this game.

Game $\mathbf{G}_1$: We replace Σ by a fresh signature (during the signing queries), $\Sigma = \mathsf{Sig}'(\mathsf{sk}, U^*)$, as signatures from $\mathsf{DerivSign}'$ and Sig' follows perfectly indistinguishable distributions in an $\mathsf{OT\text{-}LH}$ signature scheme: $\mathsf{Adv}_0 = \mathsf{Adv}_1$.

Game $\mathbf{G}_2$: The simulator generates the proofs Π, during the signing queries only, with the zero-knowledge simulator. Thanks to the (perfect) zero-knowledge property, this game is indistinguishable from the previous one: $\mathsf{Adv}_1 = \mathsf{Adv}_2$. Now, the simulator does not need to know the scalars w_{id} for signing queries, but only for key queries.

Game $\mathbf{G}_3$: The simulator generates the signatures using Square Diffie-Hellman tuples (h_i, h'_i, h''_i), with unknown scalars w_i: $\mathsf{Adv}_2 = \mathsf{Adv}_3$.

Game $\mathbf{G}_4$: In this game, we always output 0, meaning the tracing procedure always succeeds, and so $\mathsf{Adv}_4 = 0$.

Let us study the gap:

- If we consider $\mathsf{OSig}'(m)$ the signature oracle in the EUF security game of the $\mathsf{OT\text{-}LH}$ scheme, under the unforgeability result on Σ, the U^* of the output signature $(m', \mathcal{T}', \sigma)$ of our adversary necessarily involves a linear combination of the U^*_i, which implies a linear combination of the $\mathbf{k}^*_{\mathsf{id},0}$ and $\mathbf{b}^*_2$. Using the signature from [HPP20], we even get the coefficients of this linear combination;
- If we consider the 1-st, 5-th and 6-th components, which constitute a Square Diffie-Hellman tuple with an exponent that is a (known) linear combination of the scalars involved in the keys or signatures, the Theorem 11 implies that there is necessarily either a w_i involved in a signing-query or a w_{id} involved in a key-query in this U^*;
- With the additional proof of knowledge Π, from the simulation-extractability, one can extract this scalar: which is as hard as breaking the Decisional Square Diffie-Hellman if this is a w_i from a signing-query, because of the challenge (h_i, h'_i, h''_i), which could either be a random tuple or a Square Diffie-Hellman tuple.

As a consequence, except with probability bounded by $\mathsf{Adv}^{\mathsf{euf}}_{\mathsf{OT\text{-}LH}}(t) + \mathsf{Adv}^{\mathsf{lsqp}}(t) + \mathsf{Adv}^{\mathsf{dsqdh}}(t)$, this is necessarily a w_{id} from a key-query, which can thus be extracted by the tracing algorithm: $\mathsf{Adv}_3 - \mathsf{Adv}_4 \leq \mathsf{Adv}^{\mathsf{euf}}_{\mathsf{OT\text{-}LH}}(t) + \mathsf{Adv}^{\mathsf{lsqp}}(t) + \mathsf{Adv}^{\mathsf{dsqdh}}(t)$.