Efficient Attribute-Based Signatures for Non-Monotone Predicates in the Standard Model*

Tatsuaki Okamoto
NTT
okamoto.tatsuaki@lab.ntt.co.jp

Katsuyuki Takashima
Mitsubishi Electric
Takashima.Katsuyuki@aj.MitsubishiElectric.co.jp

August 29, 2014

Abstract

This paper presents a *fully* secure (*adaptive*-predicate unforgeable and private) attribute-based signature (ABS) scheme in the *standard* model. The security of the proposed ABS scheme is proven under standard assumptions, the decisional linear (DLIN) assumption and the existence of collision resistant (CR) hash functions. The admissible predicates of the proposed ABS scheme are more general than those of the existing ABS schemes, i.e., the proposed ABS scheme is the first to support general *non-monotone* predicates, which can be expressed using *NOT* gates as well as AND, OR, and Threshold gates, while the existing ABS schemes only support *monotone* predicates. The proposed ABS scheme is comparably as efficient as (several times worse than) one of the most efficient ABS schemes, which is proven to be secure in the generic group model.

*An extended abstract of a preliminary version of this paper was presented in [29] at Public Key Cryptography – PKC 2011. The journal version [30] provides significant technical contributions over [29], e.g., definition of unforgeability, removing the limitation for "multi-use." Refer to Sections 1.3, 3.2 (Remark 4), and Appendix E.1. This is the full version of [30].

Contents

1 Introduction

1.1 Background

The concept of digital signatures was introduced in the seminal paper by Diffie and Hellman in 1976. In this concept, a pair comprising a secret signing key, sk, and public verification key, pk, is generated for a signer, and signature σ of message m generated using sk is verified by the corresponding pk. Hence, the signer of (m, σ) using sk is identified through pk. Although it is one of the requirements of signatures, there is no flexibility or privacy in the relationship between signers and claims attested by signatures due to the tight relation between sk and pk.

Recently, versatile and privacy-enhanced variants of digital signatures have been studied, where the relation between a signing key and verification key is more flexible or sophisticated. In this class of signatures, the signing key and verification key are parameterized by *attribute* $\boldsymbol{x}$ and *predicate* $\boldsymbol{v}$, respectively, and signed message (m, σ) generated by the signing key with parameter $\boldsymbol{x}$, $\mathsf{sk}_{\boldsymbol{x}}$, is correctly verified by public-key pk and parameter $\boldsymbol{v}$, $(\mathsf{pk}, \boldsymbol{v})$, iff predicate $\boldsymbol{v}$ accepts attribute $\boldsymbol{x}$, i.e., $\boldsymbol{v}(\boldsymbol{x})$ holds. The privacy of signers in this class of signatures requires that a signature (for predicate $\boldsymbol{v}$) generated by $\mathsf{sk}_{\boldsymbol{x}}$ (where $\boldsymbol{v}(\boldsymbol{x})$ holds) release no information regarding attribute $\boldsymbol{x}$ except that $\boldsymbol{v}(\boldsymbol{x})$ holds.

When predicate $\boldsymbol{v}$ is the equality with parameter v (i.e., $\boldsymbol{v}(x)$ holds iff $x = v$), the class of signatures for this predicate is *identity-based signatures* (IBS) [33]. Here note that there is no room for privacy in IBS, since predicate $\boldsymbol{v}$ uniquely identifies attribute x of the signer's secret key, sk_x, such that $x = v$.

This class of signatures with more sophisticated predicates, *attribute-based signatures* (ABS), has been studied [13, 18, 17, 22, 23, 24, 25, 32, 37], where $\boldsymbol{x}$ for signing key $\mathsf{sk}_{\boldsymbol{x}}$ is a tuple of attributes $(x_1, \ldots, x_i)$, and $\boldsymbol{v}$ for verification is a threshold or access structure predicate. The widest class of predicates in the existing ABS schemes are monotone access structures [24, 25].

An example of such monotone access structure predicate $\boldsymbol{v}$ for ABS is (Institute = Univ. A) AND (TH2((Department = Biology), (Gender = Female), (Age = 50's)) OR (Position = Professor)), where TH2 means the threshold gate with threshold value 2. Attribute $\boldsymbol{x}_A$ of Alice is ((Institute := Univ. A), (Department := Biology), (Position := Postdoc), (Age := 30), (Gender := Female))), and attribute $\boldsymbol{x}_B$ of Bob is ((Institute := Univ. A), (Department := Mathematics), (Position := Professor), (Age := 45) (Gender := Male))). Although their attributes, $\boldsymbol{x}_A$ and $\boldsymbol{x}_B$, are quite different, it is clear that $\boldsymbol{v}(\boldsymbol{x}_A)$ and $\boldsymbol{v}(\boldsymbol{x}_B)$ hold, and that there are many other attributes that satisfy $\boldsymbol{v}$. Hence Alice and Bob can generate a signature on this predicate, and due to the privacy requirement of ABS, a signature for $\boldsymbol{v}$ releases no information regarding the attribute or identity of the signer, i.e., Alice or Bob (or other), except that the attribute of the signer satisfies $\boldsymbol{v}$.

There are many applications of ABS such as attribute-based messaging (ABM), attribute-based authentication, trust-negotiation and leaking secrets (see [24, 25] for more details).

The security conditions for ABS are given hereafter (see Section 3.2 for the formal definitions).

Unforgeability: A valid signature should be produced only by a *single* signer whose attribute $\boldsymbol{x}$ satisfies the claimed predicate $\boldsymbol{v}$, not by a collusion of users who pooled their attributes together. More formally, no poly-time adversary can produce a valid signature for a pair comprising predicate and message $(\boldsymbol{v}, m)$, even if the adversary *adaptively* chooses $(\boldsymbol{v}, m)$ after executing secret-key and signing oracle attacks, provided that $\boldsymbol{x}$ where $\boldsymbol{v}(\boldsymbol{x})$ holds is not queried to the secret-key oracle and $(\boldsymbol{v}, m)$ is not queried to the signing oracle (We simply call this unforgeability "*adaptive*-predicate unforgeability" or more simply "unforgeability").

We can also define a *weaker* class of unforgeability, '*selective*-predicate unforgeability,' where an adversary should choose predicate $\boldsymbol{v}$ for the forgery signature before executing secret-key and signing oracle attacks.

Privacy: A signature for predicate $\boldsymbol{v}$ generated using secret key $\mathsf{sk}_{\boldsymbol{x}}$ releases no information regarding attribute $\boldsymbol{x}$ except that $\boldsymbol{v}(\boldsymbol{x})$ holds.

More formally, for any pair of attributes $(\boldsymbol{x}_1, \boldsymbol{x}_2)$, predicate $\boldsymbol{v}$ and message m, for which $\boldsymbol{v}(\boldsymbol{x}_1)$ and $\boldsymbol{v}(\boldsymbol{x}_2)$ hold simultaneously, the distributions of two valid signatures $\sigma(m, \boldsymbol{v}, \mathsf{sk}_{\boldsymbol{x}_1})$ and $\sigma(m, \boldsymbol{v}, \mathsf{sk}_{\boldsymbol{x}_2})$ are equivalent, where $\sigma(m, \boldsymbol{v}, \mathsf{sk}_{\boldsymbol{x}})$ is a correctly generated signature for $(m, \boldsymbol{v})$ using correct secret key $\mathsf{sk}_{\boldsymbol{x}}$ with attribute $\boldsymbol{x}$ (We simply call this condition "*privacy*").

Full Security: We say that an ABS scheme is *fully* secure if it satisfies *adaptive*-predicate unforgeability and *privacy*.

Maji et al. [24, 25] presented ABS schemes for the widest class of predicates among the existing ABS schemes, monotone access structure predicates, which cover threshold predicates as special cases. The scheme shown in [24] is an efficient and practical ABS scheme, but the security was only proven in the generic group model. The schemes in [25] and by Escala et al.[11] are the only existing ABS schemes that were proven to be fully secure in the standard model. They are, however, much less efficient and more complicated than the scheme in [24] since it employs the Groth-Sahai NIZK protocols [12] as building blocks.

Herranz et al.[15], Li et al.[22], Li et al.[23], and Shahandashti et al.[32] presented ABS schemes that are proven to be secure in the standard model. However, the proven security is not the full security, but a weaker level of security with *selective*-predicate unforgeability. Moreover, the admissible predicates in [23] are limited to conjunction or (n, n)-threshold predicates, and those of [22, 32] are limited to (k, n)-threshold predicates. Guo et al.[13] and Yang et al.[37] presented ABS schemes for threshold predicates, but their security definitions do not include the *privacy* condition of ABS.

Khader [18, 17] presented ABS schemes for monotone access structure predicates. These schemes, however, do not satisfy the *privacy* condition of ABS, since they only conceal the identity of the signer. They also reveal the attributes that the signer used to generate the signature. In addition, the security is proven in a non-standard model, the random oracle model.

Based on this background, there are two major problems in the existing ABS schemes.

1. No ABS scheme for *non-monotone* predicates, which can be expressed using NOT gates as well as AND, OR and Threshold gates, has been proposed (even in a weaker security notion or a non-standard model).

2. The only fully secure ABS schemes in the *standard* model [11, 25] are much less efficient than the ABS scheme in the generic group model [24].

Non-monotone predicates should be used in many ABS applications. For example, annual review reports in the Mathematics Department of University A are submitted by reviewers, and these reports are anonymously signed by the reviewers through ABS with some predicates. The predicates may be selected freely by them (signers) except that it should be in the following form: NOT((Institute = Univ. A) AND (Department = Mathematics)) AND $(\cdots)$.

1.2 Our Results

This paper addresses these problems simultaneously.

- This paper proposes the first fully secure (i.e., adaptive-predicate unforgeable and perfectly private) ABS scheme for a wide class of predicates, *non-monotone* access structures, where $\boldsymbol{x}$ for signing key $\mathsf{sk}_{\boldsymbol{x}}$ is a tuple of attributes $(x_1, \ldots, x_i)$, non-monotone predicate $\boldsymbol{v}$ is specified by a *span program* (SP) (M, ρ) along with a tuple of attributes $(v_1, \ldots, v_j)$, and $\boldsymbol{v}(\boldsymbol{x})$ holds iff SP (M, ρ) accepts the truth-value vector of $(\mathsf{T}(x_{i_1} = v_1), \ldots, \mathsf{T}(x_{i_j} = v_j))$. Here, $\mathsf{T}(\psi) := 1$ if ψ is true, and $\mathsf{T}(\psi) := 0$ if ψ is false.

  Our scheme can be generalized using non-monotone access structures combined with *inner-product relations* (see Definition 5 and the remark). More precisely, attribute $\boldsymbol{x}$ for signing key $\mathsf{sk}_{\boldsymbol{x}}$ is a tuple of attribute vectors (e.g., $(\vec{x}_1, \ldots, \vec{x}_i) \in \mathbb{F}_q^{n_1+\cdots+n_i}$), and predicate $\boldsymbol{v}$ for verification is a non-monotone access structure or span program (SP) (M, ρ) along with a tuple of attribute vectors (e.g., $(\vec{v}_1, \ldots, \vec{v}_j) \in \mathbb{F}_q^{n_1+\cdots+n_j}$), where the component-wise inner-product relations for attribute vectors (e.g., $\{\vec{x}_{i_\iota} \cdot \vec{v}_\iota = 0 \text{ or not }\}_{\iota \in \{1,\ldots,j\}}$) are input to SP (M, ρ). Namely, $\boldsymbol{v}(\boldsymbol{x})$ holds iff the truth-value vector of $(\mathsf{T}(\vec{x}_{i_1} \cdot \vec{v}_1 = 0), \ldots, \mathsf{T}(\vec{x}_{i_j} \cdot \vec{v}_j = 0))$ is accepted by SP (M, ρ).

  Remark: In our scheme (Section 4), attribute $\boldsymbol{x}$ is expressed by the form $\Gamma := \{(t, x_t) \mid t \in T \subseteq \{1, \ldots, d\}\}$ in place of just an attribute tuple $(x_1, \ldots, x_i)$, where t identifies a sub-universe or category of attributes, and x_t is an attribute in sub-universe t (examples of (t, x_t) are (Name, Alice) and (Age, 38)). Predicate $\boldsymbol{v}$ is expressed by $\mathbb{S} := (M, \rho)$, where ρ is abused as ρ (defined by SP) combined with $\{(t_i, v_i) \mid i = 1, \ldots, \ell\}$ (see Definitions 4 and 5 for the difference regarding ρ in SP and $\mathbb{S}$).

- The proposed ABS scheme is proven to be fully secure under standard assumptions, the *decisional linear (DLIN)* assumption (over prime order pairing groups) and the existence of *collision resistant (CR) hash* functions, in the *standard* model.

- In contrast to the ABS schemes in [11, 25] that employ the Groth-Sahai NIZK protocols, our ABS scheme is more directly constructed without using any general subprotocols like NIZK. Our construction is based on the dual pairing vector spaces (DPVS) proposed by Okamoto and Takashima [26, 27, 20, 28], which can be realized from *any type of (e.g., symmetric or asymmetric) prime order bilinear pairing groups*. See Section 2.1 for the concept and actual construction of DPVS.

- The efficiency of the proposed ABS scheme is comparable to that of the most efficient ABS scheme in the generic group model [24], and better than those of the existing fully secure ABS schemes in the standard model [11, 25]. See Section 4.4 for a comparison.

- This paper also presents an extension, multi-authority (MA) setting, of the proposed ABS scheme. One of the merits of our MA-ABS scheme is that it is seamlessly extended from the original (single-authority (SA)) setting, in which the signing and verification algorithms of the MA-ABS scheme are essentially the same as those of the original ABS (SA-ABS) scheme.

  In MA-ABS, each authority called an attribute authority is responsible for a single (or multiple) category of attributes, and a user obtains a part of secret key for each attribute from an attribute authority responsible for the category of the attribute. In our MA-ABS model, a central trustee in addition to attribute authorities is required but no interaction among attribute authorities (and the trustee) is necessary, and different attribute authorities may not trust each other, nor even be aware of each other.

  We prove that the proposed MA-ABS scheme is fully secure under the DLIN assumption and CR hash functions in the standard model. Our MA-ABS scheme is almost as efficient as the original SA-ABS scheme.

1.3 Key Techniques

The top level strategy of constructing the proposed ABS scheme is based on Naor's paradigm of converting IBE to signatures. Our ABS scheme is converted from a ciphertext policy (CP) functional encryption (FE) scheme [28], which is adaptively payload-hiding. The description of the CP-FE scheme is given in the full version of [28].

Roughly speaking, in the conversion, a secret signing key, sk_Γ, with attribute set Γ and a verification text, $\vec{\boldsymbol{c}}$, with access structure $\mathbb{S}$ (for signature verification) in our ABS scheme correspond to a secret decryption key, sk_Γ, with Γ and a ciphertext, $\vec{\boldsymbol{c}}$, with $\mathbb{S}$ in the CP-FE scheme, respectively.

Our construction, however, is not straightforward, or still a challenging task, since no counterpart of a signature, $\vec{\boldsymbol{s}}^*$, in the ABS exists in the CP-FE, and the privacy property for signature $\vec{\boldsymbol{s}}^*$ is specific in ABS.

To tackle the issue, we develop a new technique, *re-randomization with specialized delegation*, where signature $\vec{\boldsymbol{s}}^*$ in ABS can be interpreted to be a decryption key specialized to decrypt a ciphertext with access structure $\mathbb{S}$, that is delegated and re-randomized from secret key sk_Γ.

As for the security proof, roughly speaking, the *adaptive*-predicate unforgeability of the ABS under the KeyGen oracle attacks can be guaranteed by the *non-adaptive* payload-hiding security of the CP-FE[1]. This is because, in the security game, a forged signature implies a decryption key specified for the challenge ciphertext to break the payload-hiding, and all secret key and signing queries are made by an adversary *before* giving a forged signature in the adaptive-predicate unforgeability of ABS, where all secret key queries are made by an adversary *before* requesting a challenge ciphertext in the non-adaptive payload-hiding of FE.

Note that there are many subtleties in the proof of unforgeability for the ABS, e.g., the unforgeability should be ensured in the ABS even when publishing $\{\widehat{\mathbb{B}}^*_t\}_{t=1,\ldots,d+1}$ for the privacy requirement, while they are secret in the CP-FE. We develop a novel technique to resolve the difficulty. See Section E for more details.

We now describe a new key technique in this paper, which was not employed in the preliminary version [29] of this paper.

A key technique of proving the (non-)adaptive payload-hiding security of the CP-FE scheme [28] is *pairwise independence lemma* (Lemma 3 in [28]) in the dual system encryption methodology. A drawback of this technique is that it is directly applicable only when there is a one-to-one correspondence (so-called "one-use") between a pair of secret key and ciphertext parts through a map ρ of policy (access structures $\mathbb{S}$), but in general a ciphertext part corresponds to multiple secret key parts (so-called "multi-use"). [28] introduced a technique to treat such a multi-use case by using a generalized pairwise independence lemma, but it costs longer secret keys and ciphertexts than those in one-use, and the public parameter bounds the maximum degree of multi-use. The security (unforgeability) proof of the preliminary version [29] is based on the (generalized) pairwise independence lemma technique and inherits the drawback of this technique. In this paper, to address the issue we introduce a new technique, *one-dimensional localization of inner-product values*, where an unbounded (by the public parameter) number of inner-product values in multi-use are localized into a certain one-dimensional subspace and the other subspaces include no information of the inner-product values, while no information but only a single inner-product value is ensured to be released for a pair of corresponding secret key and ciphertext subspaces in one-use by the pairwise independence lemma. Note that this techique is available for proving the adaptive-predicate unforgeability of the ABS and the non-adaptive payload-hiding security of the CP-FE (but not for the adaptive payload-hiding

[1] Non-adaptive security of CP-FE means that the adversary's key queries may not depend on the challenge ciphertext [1].

security of the CP-FE). For more details, see Section E.

1.4 Related Works

- **Ring and mesh signatures:** Ring and mesh signatures [31, 6] are related to ABS.

  In the ring signatures, the claimed predicate on a signature of message m is that m is endorsed by one of the users identified by the list of public keys $(\mathsf{pk}_1, \mathsf{pk}_2, \ldots)$, or the predicate is a disjunction of a list of public keys. A valid ring signature can be generated by one of the listed users.

  The mesh signatures are an extension of ring signatures, where the predicate is an access structure on a list of pairs comprising a message and public key (m_i, pk_i), and a valid mesh signature can be generated by a person who has enough standard signatures σ_i on m_i, each valid under pk_i, to satisfy the given access structure.

  A crucial difference between mesh signatures and ABS is the security against the collusion of users. In mesh signatures, several users can collude by pooling their signatures together and create signatures that none of them could produce individually. That is, such collusion is considered to be legitimate in mesh signatures. In contrast, the security against collusion attacks is one of the basic requirements in ABS and MA-ABS, as described in Section 1.1.

- **Anonymous credentials (ACs):** Another related concept is ACs [3, 4, 7, 8, 9, 10]. The notion of ACs also provides a functionality for users to demonstrate anonymously possession of attributes, but the goals of ACs and ABS differ in several points.

  As mentioned in [25], ACs and ABS aim at different goals: ACs target very strong anonymity even in the registration phase, whereas under less demanding anonymity requirements in the registration phase, ABS aims to achieve more expressive functionalities, more efficient constructions and new applications. In addition, ABS is a signature scheme and a simpler primitive compared with ACs.

1.5 Notations

When A is a random variable or distribution, $y \xleftarrow{\mathsf{R}} A$ denotes that y is randomly selected from A according to its distribution. When A is a set, $y \xleftarrow{\mathsf{U}} A$ denotes that y is uniformly selected from A. $y := z$ denotes that y is set, defined or substituted by z. When a is a fixed value, $A(x) \to a$ (e.g., $A(x) \to 1$) denotes the event that machine (algorithm) A outputs a on input x. A function $f : \mathbb{N} \to \mathbb{R}$ is *negligible* in λ, if for every constant $c > 0$, there exists an integer n such that $f(\lambda) < \lambda^{-c}$ for all $\lambda > n$.

We denote the finite field of order q by $\mathbb{F}_q$, and $\mathbb{F}_q \setminus \{0\}$ by $\mathbb{F}_q^\times$. A vector symbol denotes a vector representation over $\mathbb{F}_q$, e.g., $\vec{x}$ denotes $(x_1, \ldots, x_n) \in \mathbb{F}_q^n$. For two vectors $\vec{x} = (x_1, \ldots, x_n)$ and $\vec{v} = (v_1, \ldots, v_n)$, $\vec{x} \cdot \vec{v}$ denotes the inner-product $\sum_{i=1}^n x_i v_i$. The vector $\vec{0}$ is abused as the zero vector in $\mathbb{F}_q^n$ for any n. X^{T} denotes the transpose of matrix X. A bold face letter denotes an element of vector space $\mathbb{V}$, e.g., $\boldsymbol{x} \in \mathbb{V}$. When $\boldsymbol{b}_i \in \mathbb{V}$ $(i = 1, \ldots, n)$, $\mathsf{span}\langle \boldsymbol{b}_1, \ldots, \boldsymbol{b}_n \rangle \subseteq \mathbb{V}$ (resp. $\mathsf{span}\langle \vec{x}_1, \ldots, \vec{x}_n \rangle$) denotes the subspace generated by $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ (resp. $\vec{x}_1, \ldots, \vec{x}_n$). For bases $\mathbb{B} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_N)$ and $\mathbb{B}^* := (\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_N^*)$, $(x_1, \ldots, x_N)_{\mathbb{B}} := \sum_{i=1}^N x_i \boldsymbol{b}_i$ and $(y_1, \ldots, y_N)_{\mathbb{B}^*} := \sum_{i=1}^N y_i \boldsymbol{b}_i^*$. For a format of attribute vectors $\vec{n} := (d; n_1, \ldots, n_d)$ that indicates dimensions of vector spaces, $\vec{e}_{t,j}$ denotes the canonical basis vector $(\overbrace{0 \cdots 0}^{j-1}, 1, \overbrace{0 \cdots 0}^{n_t - j}) \in \mathbb{F}_q^{n_t}$ for $t = 1, \ldots, d$ and $j = 1, \ldots, n_t$.

2 Preliminaries

2.1 Dual Pairing Vector Spaces by Direct Product of Symmetric Pairing Groups

In this paper, for simplicity of description, we will present the proposed schemes on the symmetric version of dual pairing vector spaces (DPVS) [26, 27] constructed using symmetric bilinear pairing groups given in Definition 1. Owing to the abstraction of DPVS, the presentation and the security proof of the proposed schemes are essentially the same as those on the asymmetric version of DPVS, $(q, \mathbb{V}, \mathbb{V}^*, \mathbb{G}_T, \mathbb{A}, \mathbb{A}^*, e)$, for which see Appendix A.2. The symmetric version is a specific (self-dual) case of the asymmetric version, where $\mathbb{V} = \mathbb{V}^*$ and $\mathbb{A} = \mathbb{A}^*$.

Definition 1 *"Symmetric bilinear pairing groups"* $(q, \mathbb{G}, \mathbb{G}_T, G, e)$ *are a tuple of a prime* q*, cyclic additive group* $\mathbb{G}$ *and multiplicative group* $\mathbb{G}_T$ *of order* q*,* $G \neq 0 \in \mathbb{G}$*, and a polynomial-time computable nondegenerate bilinear pairing* $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ *i.e.,* $e(sG, tG) = e(G, G)^{st}$ *and* $e(G, G) \neq 1$*.*

Let $\mathcal{G}_{\mathsf{bpg}}$ *be an algorithm that takes input* 1^λ *and outputs a description of bilinear pairing groups* $(q, \mathbb{G}, \mathbb{G}_T, G, e)$ *with security parameter* λ*.*

Definition 2 *"Dual pairing vector spaces (DPVS)"* $(q, \mathbb{V}, \mathbb{G}_T, \mathbb{A}, e)$ *by a direct product of symmetric pairing groups* $(q, \mathbb{G}, \mathbb{G}_T, G, e)$ *are a tuple of prime* q*,* N*-dimensional vector space* $\mathbb{V} := \overbrace{\mathbb{G} \times \cdots \times \mathbb{G}}^{N}$ *over* $\mathbb{F}_q$*, cyclic group* $\mathbb{G}_T$ *of order* q*, canonical basis* $\mathbb{A} := (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N)$ *of* $\mathbb{V}$*, where* $\boldsymbol{a}_i := (\overbrace{0, \ldots, 0}^{i-1}, G, \overbrace{0, \ldots, 0}^{N-i})$*, and pairing* $e : \mathbb{V} \times \mathbb{V} \to \mathbb{G}_T$*.*

The pairing is defined by $e(\boldsymbol{x}, \boldsymbol{y}) := \prod_{i=1}^{N} e(G_i, H_i) \in \mathbb{G}_T$ *where* $\boldsymbol{x} := (G_1, \ldots, G_N) \in \mathbb{V}$ *and* $\boldsymbol{y} := (H_1, \ldots, H_N) \in \mathbb{V}$*. This is nondegenerate bilinear i.e.,* $e(s\boldsymbol{x}, t\boldsymbol{y}) = e(\boldsymbol{x}, \boldsymbol{y})^{st}$ *and if* $e(\boldsymbol{x}, \boldsymbol{y}) = 1$ *for all* $\boldsymbol{y} \in \mathbb{V}$*, then* $\boldsymbol{x} = \boldsymbol{0}$*. For all* i *and* j*,* $e(\boldsymbol{a}_i, \boldsymbol{a}_j) = e(G, G)^{\delta_{i,j}}$ *where* $\delta_{i,j} = 1$ *if* $i = j$*, and* 0 *otherwise, and* $e(G, G) \neq 1 \in \mathbb{G}_T$*.*

DPVS generation algorithm $\mathcal{G}_{\mathsf{dpvs}}$ *takes input* 1^λ *(*$\lambda \in \mathbb{N}$*) and* $N \in \mathbb{N}$*, and outputs a description of* $\mathsf{param}_{\mathbb{V}} := (q, \mathbb{V}, \mathbb{G}_T, \mathbb{A}, e)$ *with security parameter* λ *and* N*-dimensional* $\mathbb{V}$*. It can be constructed by using* $\mathcal{G}_{\mathsf{bpg}}$*.*

Remark 1 For matrix $W := (w_{i,j})_{i,j=1,\ldots,N} \in \mathbb{F}_q^{N \times N}$ and element $\boldsymbol{g} := (G_1, \ldots, G_N)$ in N-dimensional $\mathbb{V}$, $\boldsymbol{g}W$ denotes $(\sum_{i=1}^{N} G_i w_{i,1}, \ldots, \sum_{i=1}^{N} G_i w_{i,N}) = (\sum_{i=1}^{N} w_{i,1} G_i, \ldots, \sum_{i=1}^{N} w_{i,N} G_i)$ by a natural multiplication of a N-dim. row vector and a $N \times N$ matrix. Thus it holds an associative law as $(\boldsymbol{g}W)W^{-1} = \boldsymbol{g}(WW^{-1}) = \boldsymbol{g}$ and a pairing invariance property $e(\boldsymbol{g}W, \boldsymbol{h}(W^{-1})^{\mathrm{T}}) = e(\boldsymbol{g}, \boldsymbol{h})$ for any $\boldsymbol{g}, \boldsymbol{h} \in \mathbb{V}$.

2.2 Decisional Linear (DLIN) Assumption

Definition 3 (DLIN Assumption) *The DLIN problem is to guess* $\beta \in \{0, 1\}$*, given* $(\mathsf{param}_{\mathbb{G}}, G, \xi G, \kappa G, \delta\xi G, \sigma\kappa G, Y_\beta) \xleftarrow{\mathsf{R}} \mathcal{G}_\beta^{\mathsf{DLIN}}(1^\lambda)$*, where*

$$\begin{aligned} \mathcal{G}_\beta^{\mathsf{DLIN}}(1^\lambda) : \mathsf{param}_{\mathbb{G}} &:= (q, \mathbb{G}, \mathbb{G}_T, G, e) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{bpg}}(1^\lambda), \\ \kappa, \delta, \xi, \sigma &\xleftarrow{\mathsf{U}} \mathbb{F}_q, \quad Y_0 := (\delta + \sigma)G, \quad Y_1 \xleftarrow{\mathsf{U}} \mathbb{G}, \\ \text{return} \quad &(\mathsf{param}_{\mathbb{G}}, \ G, \xi G, \kappa G, \delta\xi G, \sigma\kappa G, Y_\beta), \end{aligned}$$

for $\beta \xleftarrow{\mathsf{U}} \{0, 1\}$*. For a probabilistic machine* $\mathcal{E}$*, we define the advantage of* $\mathcal{E}$ *for the DLIN problem as:* $\mathsf{Adv}_{\mathcal{E}}^{\mathsf{DLIN}}(\lambda) := \left| \mathsf{Pr}\left[\mathcal{E}(1^\lambda, \varrho) \to 1 \;\middle|\; \varrho \xleftarrow{\mathsf{R}} \mathcal{G}_0^{\mathsf{DLIN}}(1^\lambda)\right] - \mathsf{Pr}\left[\mathcal{E}(1^\lambda, \varrho) \to 1 \;\middle|\; \varrho \xleftarrow{\mathsf{R}} \mathcal{G}_1^{\mathsf{DLIN}}(1^\lambda)\right] \right|$.

The DLIN assumption is: For any probabilistic polynomial-time adversary $\mathcal{E}$, the advantage $\mathsf{Adv}_{\mathcal{E}}^{\mathsf{DLIN}}(\lambda)$ is negligible in λ.

2.3 Collision Resistant (CR) Hash Functions

Let $\lambda \in \mathbb{N}$ be a security parameter. A collision resistant (CR) hash function family, H, associated with $\mathcal{G}_{\mathsf{bpg}}$ and a polynomial, $poly(\cdot)$, specifies two items:

- A family of key spaces indexed by λ. Each such key space is a probability space on bit strings denoted by KH_λ. There must exist a probabilistic polynomial-time algorithm whose output distribution on input 1^λ is equal to KH_λ.
- A family of hash functions indexed by λ, $\mathsf{hk} \xleftarrow{\mathsf{R}} \mathsf{KH}_\lambda$ and $\mathsf{D} := \{0,1\}^{poly(\lambda)}$. Each such hash function $\mathsf{H}_{\mathsf{hk}}^{\lambda,\mathsf{D}}$ maps an element of D to an element of $\mathbb{F}_q^\times$ with q that is the first element of output $\mathsf{param}_{\mathbb{G}}$ of $\mathcal{G}_{\mathsf{bpg}}(1^\lambda)$. There must exist a deterministic polynomial-time algorithm that on input 1^λ, hk and $\varrho \in \mathsf{D}$, outputs $\mathsf{H}_{\mathsf{hk}}^{\lambda,\mathsf{D}}(\varrho)$.

Let $\mathcal{E}$ be a probabilistic polynomial-time machine. For all λ, we define $\mathsf{Adv}_{\mathcal{E}}^{\mathsf{H},\mathsf{CR}}(\lambda) := \Pr[(\varrho_1,\varrho_2) \in \mathsf{D}^2 \wedge \varrho_1 \neq \varrho_2 \wedge \mathsf{H}_{\mathsf{hk}}^{\lambda,\mathsf{D}}(\varrho_1) = \mathsf{H}_{\mathsf{hk}}^{\lambda,\mathsf{D}}(\varrho_2)]$, where $\mathsf{D} := \{0,1\}^{poly(\lambda)}, \mathsf{hk} \xleftarrow{\mathsf{R}} \mathsf{KH}_\lambda$, and $(\varrho_1,\varrho_2) \xleftarrow{\mathsf{R}} \mathcal{E}(1^\lambda, \mathsf{hk}, \mathsf{D})$. H is a collision resistant (CR) hash function family if for any probabilistic polynomial-time adversary $\mathcal{E}$, $\mathsf{Adv}_{\mathcal{E}}^{\mathsf{H},\mathsf{CR}}(\lambda)$ is negligible in λ.

3 ABS for Non-monotone Predicates

3.1 Span Programs and Non-monotone Access Structures

Definition 4 (Span Programs [2]) *Let $\{p_1,\ldots,p_n\}$ be a set of variables. A span program over $\mathbb{F}_q$ is a labeled matrix, $\hat{M} := (M,\rho)$, where M is a $(\ell \times r)$ matrix over $\mathbb{F}_q$ and ρ is a labeling of the rows of M by literals from $\{p_1,\ldots,p_n,\neg p_1,\ldots,\neg p_n\}$ (every row is labeled by one literal), i.e., $\rho : \{1,\ldots,\ell\} \to \{p_1,\ldots,p_n,\neg p_1,\ldots,\neg p_n\}$.*

A span program accepts or rejects an input by the following criterion. For every input sequence $\delta \in \{0,1\}^n$ define submatrix M_δ of M consisting of those rows whose labels are set to 1 by the input δ, i.e., either rows labeled by some p_i such that $\delta_i = 1$ or rows labeled by some by some $\neg p_i$ such that $\delta_i = 0$. (i.e., $\gamma : \{1,\ldots,\ell\} \to \{0,1\}$ is defined by $\gamma(j) = 1$ if $[\rho(j) = p_i] \wedge [\delta_i = 1]$ or $[\rho(j) = \neg p_i] \wedge [\delta_i = 0]$, and $\gamma(j) = 0$ otherwise. $M_\delta := (M_j)_{\gamma(j)=1}$, where M_j is the j-th row of M.)

Span program $\hat{M}$ accepts δ if and only if $\vec{1} \in \mathsf{span}\langle M_\delta\rangle$, i.e., some linear combination of the rows of M_δ gives the all one vector, $\vec{1}$. (The row vector has the value 1 in each coordinate.) A span program computes boolean function f if it accepts exactly those inputs δ where $f(\delta) = 1$.

A span program is called monotone if the labels of the rows are only the positive literals $\{p_1,\ldots,p_n\}$. Monotone span programs compute monotone functions. (So, a span program in general is "non"-monotone.)

We assume that no row M_i $(i = 1,\ldots,\ell)$ of the matrix M is $\vec{0}$. We now introduce a non-monotone access structure with evaluating map γ by using the inner-product of attribute vectors in a general form. Although we will show the notion, security definition and security proof of the proposed ABS scheme in this general form, we will describe the proposed ABS scheme in a simpler form in Section 4.2. We will show this simpler form of Definition 5 in the remark.

Definition 5 (Inner-Products of Attribute Vectors and Access Structures) *$\mathcal{U}_t$ ($t = 1, \ldots, d$ and $\mathcal{U}_t \subset \{0,1\}^*$) is a sub-universe, a set of attributes, each of which is expressed by a pair of sub-universe id and n_t-dimensional vector, i.e., $(t, \vec{v})$, where $t \in \{1, \ldots, d\}$ and $\vec{v} \in \mathbb{F}_q^{n_t} \setminus \{\vec{0}\}$.*

We now define such an attribute to be a variable, p, of span program $\hat{M} := (M, \rho)$ i.e., $p := (t, \vec{v})$. Access structure $\mathbb{S}$ is span program $\hat{M} := (M, \rho)$ along with variables $p := (t, \vec{v}), p' := (t', \vec{v}'), \ldots$, i.e., $\mathbb{S} := (M, \rho)$ such that $\rho : \{1, \ldots, \ell\} \to \{(t, \vec{v}), (t', \vec{v}'), \ldots, \neg(t, \vec{v}), \neg(t', \vec{v}'), \ldots\}$.

Let Γ be a set of attributes, i.e., $\Gamma := \{(t, \vec{x}_t) \mid \vec{x}_t \in \mathbb{F}_q^{n_t} \setminus \{\vec{0}\}, 1 \le t \le d\}$.

When Γ is given to access structure $\mathbb{S}$, map $\gamma : \{1, \ldots, \ell\} \to \{0,1\}$ for span program $\hat{M} := (M, \rho)$ is defined as follows: For $i = 1, \ldots, \ell$, set $\gamma(i) = 1$ if $[\rho(i) = (t, \vec{v}_i)] \wedge [(t, \vec{x}_t) \in \Gamma] \wedge [\vec{v}_i \cdot \vec{x}_t = 0]$ or $[\rho(i) = \neg(t, \vec{v}_i)] \wedge [(t, \vec{x}_t) \in \Gamma] \wedge [\vec{v}_i \cdot \vec{x}_t \neq 0]$. Set $\gamma(i) = 0$ otherwise.

Access structure $\mathbb{S} := (M, \rho)$ accepts Γ iff $\vec{1} \in \mathsf{span}\langle (M_i)_{\gamma(i)=1} \rangle$.

Remark 2 The simplest form of the inner-product relations in the above-mentioned access structures, that is for ABS in Section 4.2, is a special case when $n_t = 2$ for all $t \in \{1, \ldots, d\}$, and $\vec{x} := (1, x)$ and $\vec{v} := (v, -1)$. Hence, $(t, \vec{x}_t) := (t, (1, x_t))$ and $(t, \vec{v}_i) := (t, (v_i, -1))$, but we often denote them shortly by (t, x_t) and (t, v_i). Then, $\mathbb{S} := (M, \rho)$ such that $\rho : \{1, \ldots, \ell\} \to \{(t, v), (t', v'), \ldots \neg(t, v), \neg(t', v'), \ldots\}$ $(v, v', \ldots \in \mathbb{F}_q)$, and $\Gamma := \{(t, x_t) \mid x_t \in \mathbb{F}_q, 1 \le t \le d\}$.

When Γ is given to access structure $\mathbb{S}$, map $\gamma : \{1, \ldots, \ell\} \to \{0,1\}$ for span program $\hat{M} := (M, \rho)$ is defined as follows: For $i = 1, \ldots, \ell$, set $\gamma(i) = 1$ if $[\rho(i) = (t, v_i)] \wedge [(t, x_t) \in \Gamma] \wedge [v_i = x_t]$ or $[\rho(i) = \neg(t, v_i)] \wedge [(t, x_t) \in \Gamma] \wedge [v_i \neq x_t]$. Set $\gamma(i) = 0$ otherwise.

Remark 3 When a user has multiple attributes in a sub-universe (category) t, we can employ dimension $n_t > 2$. For instance, a professor (say Alice) in the science faculty of a university is also a professor in the engineering faculty of this university. If the attribute authority of this university manages sub-universe $t :=$ "faculties of this university", Alice obtains a secret key for $(t, \vec{x}_t := (1, -(a+b), ab) \in {\mathbb{F}_q}^3)$ with $a :=$ "science" and $b :=$ "engineering" from the authority. When a user verifies a signature for an access structure with a single negative attribute $\neg(t,$ "science"), the verification text is encoded as $\neg(t, \vec{v}_i := (a^2, a, 1))$ with $a :=$ "science". Since $\vec{x}_t \cdot \vec{v}_i = 0$, Alice cannot make a valid signature for an access structure with the negative attribute $\neg(t,$ "science"). For such a case with $n_t > 2$, see Section C with a general form of our ABS scheme.

We now construct a secret-sharing scheme for a (non-monotone) access structure (span program).

Definition 6 *A secret-sharing scheme for access structure $\mathbb{S} := (M, \rho)$ is:*

1. *Let M be an $\ell \times r$ matrix, and column vector $\vec{f}^{\mathrm{T}} := (f_1, \ldots, f_r)^{\mathrm{T}} \xleftarrow{\cup} \mathbb{F}_q^r$. Then, $s_0 := \vec{1} \cdot \vec{f}^{\mathrm{T}} = \sum_{k=1}^{r} f_k$ is the secret to be shared, and $\vec{s}^{\mathrm{T}} := (s_1, \ldots, s_\ell)^{\mathrm{T}} := M \cdot \vec{f}^{\mathrm{T}}$ is the vector of ℓ shares of secret s_0 and share s_i belongs to $\rho(i)$.*
2. *If access structure $\mathbb{S} := (M, \rho)$ accepts Γ, i.e., $\vec{1} \in \mathsf{span}\langle (M_i)_{\gamma(i)=1} \rangle$ with $\gamma : \{1, \ldots, \ell\} \to \{0,1\}$, then there exist constants $\{\alpha_i \in \mathbb{F}_q \mid i \in I\}$ such that $I \subseteq \{i \in \{1, \ldots, \ell\} \mid \gamma(i) = 1\}$ and $\sum_{i \in I} \alpha_i s_i = s_0$. Furthermore, these constants $\{\alpha_i\}$ can be computed in time polynomial in the size of matrix M.*

3.2 Definitions and Security of ABS

Definition 7 (Attribute-Based Signatures : ABS) *An attribute-based signature scheme consists of four algorithms.*

Setup *This is a randomized algorithm that takes as input security parameter and format* $\vec{n} := (d; n_1, \ldots, n_d)$ *of attributes. It outputs public parameters* pk *and master key* sk.

KeyGen *This is a randomized algorithm that takes as input a set of attributes,* $\Gamma := \{(t, \vec{x}_t) | \vec{x}_t \in \mathbb{F}_q^{n_t} \setminus \{\vec{0}\}, 1 \leq t \leq d\}$, pk *and* sk. *It outputs signature generation key* sk_Γ.

Sig *This is a randomized algorithm that takes as input message* m, *access structure* $\mathbb{S} := (M, \rho)$, *signature generation key* sk_Γ, *and public parameters* pk *such that* $\mathbb{S}$ *accepts* Γ. *It outputs signature* σ.

Ver *This takes as input message* m, *access structure* $\mathbb{S}$, *signature* σ *and public parameters* pk. *It outputs boolean value* $\mathsf{accept} := 1$ *or* $\mathsf{reject} := 0$.

An ABS scheme should have the following correctness property: for all $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{Setup}(1^\lambda, \vec{n})$, all messages m, all attribute sets Γ, all signing keys $\mathsf{sk}_\Gamma \xleftarrow{R} \mathsf{KeyGen}(\mathsf{pk}, \mathsf{sk}, \Gamma)$, all access structures $\mathbb{S}$ such that $\mathbb{S}$ accepts Γ, and all signatures $\sigma \xleftarrow{R} \mathsf{Sig}(\mathsf{pk}, \mathsf{sk}_\Gamma, m, \mathbb{S})$, it holds that $\mathsf{Ver}(\mathsf{pk}, m, \mathbb{S}, \sigma) = 1$ with probability 1.

Definition 8 (Perfect Privacy) *An ABS scheme is perfectly private, if, for all* $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{Setup}(1^\lambda, \vec{n})$, *all messages* m, *all attribute sets* Γ_1 *and* Γ_2, *all signing keys* $\mathsf{sk}_{\Gamma_1} \xleftarrow{R} \mathsf{KeyGen}(\mathsf{pk}, \mathsf{sk}, \Gamma_1)$ *and* $\mathsf{sk}_{\Gamma_2} \xleftarrow{R} \mathsf{KeyGen}(\mathsf{pk}, \mathsf{sk}, \Gamma_2)$, *all access structures* $\mathbb{S}$ *such that* $\mathbb{S}$ *accepts* Γ_1 *and* $\mathbb{S}$ *accepts* Γ_2, *distributions* $\mathsf{Sig}(\mathsf{pk}, \mathsf{sk}_{\Gamma_1}, m, \mathbb{S})$ *and* $\mathsf{Sig}(\mathsf{pk}, \mathsf{sk}_{\Gamma_2}, m, \mathbb{S})$ *are equal.*

Since the correct distribution on signatures can be perfectly simulated without depending on any specific private information, signatures must not leak any such private information of the signer.

Definition 9 (Unforgeability) *For an adversary,* $\mathcal{A}$, *we define* $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{ABS,UF}}(\lambda)$ *to be the success probability in the following experiment for any security parameter* λ. *An ABS scheme is existentially unforgeable if the success probability of any polynomial-time adversary is negligible:*

1. *Run* $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{Setup}(1^\lambda, \vec{n})$ *and give* pk *to the adversary.*
2. $\mathcal{A}$ *may adaptively makes a polynomial number of queries of the following type:*
   - *[Create key]* $\mathcal{A}$ *asks the challenger to create a signing key for an attribute set* Γ. *The challenger creates a key for* Γ *without giving it to* $\mathcal{A}$.
   - *[Create signature]* $\mathcal{A}$ *specifies a key for predicate* Γ *that has already been created, and asks the challenger to perform a signing operation to create a signature for a message* m *and an access structure* $\mathbb{S}$ *that accepts* Γ. *The challenger computes the signature without giving it to the adversary.*
   - *[Reveal key or signature]* $\mathcal{A}$ *asks the challenger to reveal an already-created key for an attribute set* Γ, *or an already-created signature for an access structure* $\mathbb{S}$.

   Note that when key or signature creation requests are made, $\mathcal{A}$ *does not automatically see the created key or signature.* $\mathcal{A}$ *sees it only when it makes a reveal query.*
3. *At the end, the adversary outputs* $(m', \mathbb{S}', \sigma')$.

We say the adversary succeeds if a correctly-created signature for $(m', \mathbb{S}')$ *was never revealed to the adversary,* $\mathbb{S}'$ *does not accept any* Γ *queried to the create key and reveal (key) oracles, and* $\mathsf{Ver}(\mathsf{pk}, m', \mathbb{S}', \sigma') = 1$.

Remark 4 Since a signing query in the unforgeability definition in [25, 29] is made only with an access structure $\mathbb{S}$, the challenger should *find* an attribute set Γ that satisfies $\mathbb{S}$, and generate a key sk_Γ with Γ and a signature with $\mathbb{S}$ using $(\Gamma, \mathsf{sk}_\Gamma)$. In general, however, the challenger may not always find a suitable Γ from $\mathbb{S}$ in a polynomial time since it includes the problem of solving the satisfiability for any DNF and CNF formulas with polynomial sizes. In this sense, the definition of unforgeability in [25, 29] is problematic.

To address this issue, our definition of unforgeability introduces four types of queries, create and reveal queries for keys and signatures, in a manner similar to the security definition for key-delegation by Shi and Waters [34]. Here, to obtain a signature for $\mathbb{S}$ from the challenger, the adversary is required to give an attribute set Γ that satisfies $\mathbb{S}$ to the challenger in advance (i.e., the challenger has no need to find a suitable Γ by itself.)

4 Proposed ABS Scheme

4.1 Construction Ideas

As mentioned in Section 1.3, our ABS scheme is constructed on a ciphertext policy (CP) functional encryption (FE) scheme [28]. Therefore, the algorithms of the proposed ABS scheme can be described in the light of such correspondence to the CP-FE scheme:

Setup Almost the same as that in the CP-FE scheme except that $\{\widehat{\mathbb{B}}^*_t\}_{t=1,\ldots,d+1}$ are revealed as a *public* parameter in our ABS, while they are *secret* in the CP-FE scheme. They are published in our ABS for the signature generation procedure Sig to meet the *privacy* of signers (for randomization). This implies an important gap between CP-FE and ABS.

KeyGen Almost the same as that in the CP-FE scheme except that a (7 dimensional) space with basis $\mathbb{B}^*_{d+1}$ is additionally introduced in our ABS and two elements $\boldsymbol{k}^*_{d+1,1}$ and $\boldsymbol{k}^*_{d+1,2}$ in this space are included in a secret signing key in order to embed the hash value, $\mathsf{H}^{\lambda,\mathsf{D}}_{\mathsf{hk}}(m \,||\, \mathbb{S})$, of message m and access structure $\mathbb{S}$ in signature $\vec{\boldsymbol{s}}^*$.

Sig Specific in ABS. To meet the privacy condition for $\vec{\boldsymbol{s}}^*$, a novel technique is employed to randomly generate a signature from sk_Γ and $\{\widehat{\mathbb{B}}^*_t\}_{t=1,\ldots,d+1}$.

Ver Signature $\vec{\boldsymbol{s}}^*$ in the ABS is an endorsement to message m by a signer with attributes accepted by access structure $\mathbb{S}$. The signature verification in our ABS checks whether signature (or specific decryption key) $\vec{\boldsymbol{s}}^*$ works as a decryption key to decrypt a verification text (or a ciphertext) associated with $\mathbb{S}$ and $\mathsf{H}^{\lambda,\mathsf{D}}_{\mathsf{hk}}(m \,||\, \mathbb{S})$.

4.2 Construction

For simplicity, here, we describe our ABS scheme for a specific parameter $\vec{n} := (d; 2, \ldots, 2)$ (see the remark of Definition 5). A general form of our ABS scheme is given in Section C.

$$
\begin{aligned}
&\mathsf{Setup}(1^\lambda,\ \vec{n} := (d; 2, \ldots, 2)) :\ \mathsf{param}_{\mathbb{G}} := (q, \mathbb{G}, \mathbb{G}_T, G, e) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{bpg}}(1^\lambda),\\
&\quad \mathsf{hk} \xleftarrow{\mathsf{R}} \mathsf{KH}_\lambda,\ \psi \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times,\ N_0 := 4,\ N_t := 9 \text{ for } t = 1, \ldots, d+1,\\
&\quad \text{for } t = 0, \ldots, d+1,\quad \mathsf{param}_{\mathbb{V}_t} := (q, \mathbb{V}_t, \mathbb{G}_T, \mathbb{A}_t, e) := \mathcal{G}_{\mathsf{dpvs}}(1^\lambda, N_t, \mathsf{param}_{\mathbb{G}}),\\
&\qquad X_t := (\chi_{t,i,j})_{i,j} \xleftarrow{\mathsf{U}} GL(N_t, \mathbb{F}_q),\ (\vartheta_{t,i,j})_{i,j} := \psi \cdot (X_t^{-1})^{\mathrm{T}},\\
&\qquad \boldsymbol{b}_{t,i} := (\chi_{t,i,1}, \ldots, \chi_{t,i,N_t})_{\mathbb{A}_t},\ \mathbb{B}_t := (\boldsymbol{b}_{t,1}, \ldots, \boldsymbol{b}_{t,N_t}),\\
&\qquad \boldsymbol{b}^*_{t,i} := (\vartheta_{t,i,1}, \ldots, \vartheta_{t,i,N_t})_{\mathbb{A}_t},\ \mathbb{B}^*_t := (\boldsymbol{b}^*_{t,1}, \ldots, \boldsymbol{b}^*_{t,N_t}),
\end{aligned}
$$

$g_T := e(G,G)^{\psi}$, $\mathsf{param}_{\vec{n}} := (\{\mathsf{param}_{\mathbb{V}_t}\}_{t=0,\ldots,d+1},\ g_T)$,

$\widehat{\mathbb{B}}_0 := (\boldsymbol{b}_{0,1}, \boldsymbol{b}_{0,4})$, $\widehat{\mathbb{B}}_t := (\boldsymbol{b}_{t,1}, \boldsymbol{b}_{t,2}, \boldsymbol{b}_{t,8}, \boldsymbol{b}_{t,9})$ for $t = 1, \ldots, d$,

$\widehat{\mathbb{B}}_{d+1} := (\boldsymbol{b}_{d+1,1}, \boldsymbol{b}_{d+1,2}, \boldsymbol{b}_{d+1,7})$,

$\widehat{\mathbb{B}}^*_t := (\boldsymbol{b}^*_{t,1}, \boldsymbol{b}^*_{t,2}, \boldsymbol{b}^*_{t,6}, \boldsymbol{b}^*_{t,7})$ for $t = 1, \ldots, d$, $\widehat{\mathbb{B}}^*_{d+1} := (\boldsymbol{b}^*_{d+1,1}, \boldsymbol{b}^*_{d+1,2}, \boldsymbol{b}^*_{d+1,5}, \boldsymbol{b}^*_{d+1,6})$,

$\mathsf{sk} := \boldsymbol{b}^*_{0,1}$, $\mathsf{pk} := (1^\lambda, \mathsf{hk}, \mathsf{param}_{\vec{n}}, \{\widehat{\mathbb{B}}_t\}_{t=0,\ldots,d+1}, \{\widehat{\mathbb{B}}^*_t\}_{t=1,\ldots,d+1}, \boldsymbol{b}^*_{0,3})$.

return sk, pk.

$\mathsf{KeyGen}(\mathsf{pk},\ \mathsf{sk},\ \ \Gamma := \{(t, x_t) \mid 1 \le t \le d\})$:

$\delta \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times$, $\varphi_0, \varphi_{t,\iota}, \varphi_{d+1,1,\iota}, \varphi_{d+1,2,\iota} \xleftarrow{\mathsf{U}} \mathbb{F}_q$ for $t = 1, \ldots, d$; $\iota = 1, 2$;

$\boldsymbol{k}^*_0 := (\ \delta,\ 0,\ \varphi_0,\ 0\ )_{\mathbb{B}^*_0}$,

$\boldsymbol{k}^*_t := (\delta(\ 1,\ x_t\ ),\ \ 0,\ 0,\ 0,\ \ \varphi_{t,1},\ \varphi_{t,2},\ \ 0,\ 0\ )_{\mathbb{B}^*_t}$ for $(t, x_t) \in \Gamma$,

$\boldsymbol{k}^*_{d+1,1} := (\ \delta(\ 1,\ 0\ ),\ \ 0,\ 0,\ \ \varphi_{d+1,1,1},\ \varphi_{d+1,1,2},\ \ 0\ )_{\mathbb{B}^*_{d+1}}$,

$\boldsymbol{k}^*_{d+1,2} := (\ \delta(\ 0,\ 1\ ),\ \ 0,\ 0,\ \ \varphi_{d+1,2,1},\ \varphi_{d+1,2,2},\ \ 0\ )_{\mathbb{B}^*_{d+1}}$,

$T := \{0, (d+1, 1), (d+1, 2)\} \cup \{t \ \mid\ 1 \le t \le d,\ (t, x_t) \in \Gamma\}$,

return $\mathsf{sk}_\Gamma := (\Gamma, \{\boldsymbol{k}^*_t\}_{t \in T})$.

$\mathsf{Sig}(\mathsf{pk},\ \mathsf{sk}_\Gamma,\ m,\ \mathbb{S} := (M, \rho))$: If $\mathbb{S} := (M, \rho)$ accepts $\Gamma := \{(t, x_t)\}$,

then compute I and $\{\alpha_i\}_{i \in I}$ such that $\sum_{i \in I} \alpha_i M_i = \vec{1}$,

and $I \subseteq \{i \in \{1, \ldots, \ell\} \mid\ [\rho(i) = (t, v_i)\ \wedge\ (t, x_t) \in \Gamma\ \wedge\ v_i = x_t]$

$\qquad\qquad \vee\ \ [\rho(i) = \neg(t, v_i)\ \wedge\ (t, x_t) \in \Gamma\ \wedge\ v_i \neq x_t]\ \}$,

$\xi \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times$, $(\beta_i) \xleftarrow{\mathsf{U}} \{(\beta_1, \ldots, \beta_\ell)\ \mid\ \sum_{i=1}^{\ell} \beta_i M_i = \vec{0}\}$,

Remark : If $\det M \neq 0$, the set contains only 0^ℓ, i.e., all $\beta_i = 0$ for $i = 1, \ldots, \ell$.

$\boldsymbol{s}^*_0 := \xi \boldsymbol{k}^*_0 + \boldsymbol{r}^*_0$, where $\boldsymbol{r}^*_0 \xleftarrow{\mathsf{U}} \mathsf{span}\langle \boldsymbol{b}^*_{0,3} \rangle$,

$\boldsymbol{s}^*_i := \gamma_i \cdot \xi \boldsymbol{k}^*_t + \sum_{\iota=1}^{2} y_{i,\iota} \cdot \boldsymbol{b}^*_{t,\iota} + \boldsymbol{r}^*_i$ for $1 \le i \le \ell$,

where $\boldsymbol{r}^*_i \xleftarrow{\mathsf{U}} \mathsf{span}\langle \boldsymbol{b}^*_{t,5}, \boldsymbol{b}^*_{t,6} \rangle$, and $\gamma_i, \vec{y}_i := (y_{i,1}, y_{i,2})$ are defined as

if $i \in I\ \wedge\ \rho(i) = (t, v_i)$, $\gamma_i := \alpha_i$, $\vec{y}_i := \beta_i(1, v_i)$,

if $i \in I\ \wedge\ \rho(i) = \neg(t, v_i)$, $\gamma_i := \dfrac{\alpha_i}{v_i - x_t}$, $\vec{y}_i := \dfrac{\beta_i}{v_i - y_i}(1, y_i)$,

where $y_i \xleftarrow{\mathsf{U}} \mathbb{F}_q \backslash \{v_i\}$,

if $i \notin I\ \wedge\ \rho(i) = (t, v_i)$, $\gamma_i := 0$, $\vec{y}_i := \beta_i(1, v_i)$,

if $i \notin I\ \wedge\ \rho(i) = \neg(t, v_i)$, $\gamma_i := 0$, $\vec{y}_i := \dfrac{\beta_i}{v_i - y_i}(1, y_i)$,

where $y_i \xleftarrow{\mathsf{U}} \mathbb{F}_q \backslash \{v_i\}$,

$\boldsymbol{s}^*_{\ell+1} := \xi(\boldsymbol{k}^*_{d+1,1} + \mathsf{H}^{\lambda,\mathsf{D}}_{\mathsf{hk}}(m \,||\, \mathbb{S}) \cdot \boldsymbol{k}^*_{d+1,2}) + \boldsymbol{r}^*_{\ell+1}$,

where $\boldsymbol{r}^*_{\ell+1} \xleftarrow{\mathsf{U}} \mathsf{span}\langle \boldsymbol{b}^*_{d+1,5}, \boldsymbol{b}^*_{d+1,6} \rangle$,

return $\vec{\boldsymbol{s}}^* := (\boldsymbol{s}^*_0, \ldots, \boldsymbol{s}^*_{\ell+1})$.

$\mathsf{Ver}(\mathsf{pk},\ m,\ \mathbb{S} := (M, \rho), \vec{\boldsymbol{s}}^*)$: $\vec{f} \xleftarrow{\mathsf{U}} \mathbb{F}_q^r$, $\vec{s}^{\mathrm{T}} := (s_1, \ldots, s_\ell)^{\mathrm{T}} := M \cdot \vec{f}^{\mathrm{T}}$,

$s_0 := \vec{1} \cdot \vec{f}^{\mathrm{T}}$, $\eta_0, \eta_{\ell+1}, \theta_{\ell+1}, s_{\ell+1} \xleftarrow{\mathsf{U}} \mathbb{F}_q$,

$\boldsymbol{c}_0 := (\ -s_0 - s_{\ell+1},\ 0,\ 0,\ \eta_0\ )_{\mathbb{B}_0}$,

for $1 \le i \le \ell$,

if $\rho(i) = (t, v_i)$, return 0 if $\boldsymbol{s}_i^* \notin \mathbb{V}_t$, else

$\boldsymbol{c}_i := (\ s_i + \theta_i v_i,\ -\theta_i,\ \ 0,\ 0,\ 0,\ \ 0,\ 0,\ \ \eta_{i,1},\ \eta_{i,2}\ )_{\mathbb{B}_t}$, where $\theta_i, \eta_{i,1}, \eta_{i,2} \xleftarrow{\mathsf{U}} \mathbb{F}_q$,

if $\rho(i) = \neg(t, v_i)$, return 0 if $\boldsymbol{s}_i^* \notin \mathbb{V}_t$, else

$\boldsymbol{c}_i := (\ s_i(\ v_i,\ -1\ ),\ \ 0,\ 0,\ 0,\ \ 0,\ 0,\ \ \eta_{i,1},\ \eta_{i,2}\ )_{\mathbb{B}_t}$, where $\eta_{i,1}, \eta_{i,2} \xleftarrow{\mathsf{U}} \mathbb{F}_q$,

$\boldsymbol{c}_{\ell+1} := (\ s_{\ell+1} - \theta_{\ell+1} \cdot \mathsf{H}_{\mathsf{hk}}^{\lambda,\mathsf{D}}(m \,||\, \mathbb{S}),\ \theta_{\ell+1},\ \ 0,\ 0,\ \ 0,\ 0,\ \ \eta_{\ell+1}\ )_{\mathbb{B}_{d+1}}$,

return 0 if $e(\boldsymbol{b}_{0,1}, \boldsymbol{s}_0^*) = 1$,

return 1 if $\prod_{i=0}^{\ell+1} e(\boldsymbol{c}_i, \boldsymbol{s}_i^*) = 1$, return 0 otherwise.

[Correctness]

$$\begin{aligned}\prod_{i=0}^{\ell+1} e(\boldsymbol{c}_i, \boldsymbol{s}_i^*) &= e(\boldsymbol{c}_0, \boldsymbol{k}_0^*)^{\xi} \cdot \prod_{i \in I} e(\boldsymbol{c}_i, \boldsymbol{k}_t^*)^{\gamma_i \xi} \cdot \prod_{i=1}^{\ell} \prod_{\iota=1}^{2} e(\boldsymbol{c}_i, \boldsymbol{b}_{t,\iota}^*)^{y_{i,\iota}} \cdot e(\boldsymbol{c}_{\ell+1}, \boldsymbol{s}_{\ell+1}^*) \\ &= g_T^{\xi\delta(-s_0 - s_{\ell+1})} \cdot \prod_{i \in I} g_T^{\xi\delta\alpha_i s_i} \cdot \prod_{i=1}^{\ell} g_T^{\beta_i s_i} \cdot g_T^{\xi\delta s_{\ell+1}} \\ &= g_T^{\xi\delta(-s_0 - s_{\ell+1})} \cdot g_T^{\xi\delta s_0} \cdot g_T^{\xi\delta s_{\ell+1}} = 1.\end{aligned}$$

4.3 Security

Theorem 1 *The proposed ABS scheme is perfectly private.*

Theorem 2 *The proposed ABS scheme is unforgeable (adaptive-predicate unforgeable) under the DLIN assumption and the existence of collision resistant hash functions.*

For any adversary $\mathcal{A}$, there exist probabilistic machines $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_{3\text{-}1}, \ldots, \mathcal{E}_{3\text{-}4}, \mathcal{E}_5, \mathcal{E}_6$, whose running times are essentially the same as that of $\mathcal{A}$, such that for any security parameter λ,

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{ABS},\mathsf{UF}}(\lambda) \leq\ & \textstyle\sum_{i=1}^{2} \left(\mathsf{Adv}_{\mathcal{E}_{1\text{-}i}}^{\mathsf{DLIN}}(\lambda) + \mathsf{Adv}_{\mathcal{E}_{2\text{-}i}}^{\mathsf{DLIN}}(\lambda)\right) \\ & + \textstyle\sum_{h=1}^{\nu_1} \left(\mathsf{Adv}_{\mathcal{E}_{3\text{-}h\text{-}1}}^{\mathsf{DLIN}}(\lambda) + \sum_{i=1}^{2} \left(\mathsf{Adv}_{\mathcal{E}_{3\text{-}h\text{-}2\text{-}i}}^{\mathsf{DLIN}}(\lambda) + \mathsf{Adv}_{\mathcal{E}_{3\text{-}h\text{-}3\text{-}i}}^{\mathsf{DLIN}}(\lambda)\right) + \mathsf{Adv}_{\mathcal{E}_{3\text{-}h\text{-}4}}^{\mathsf{DLIN}}(\lambda)\right) \\ & + \textstyle\sum_{h=1}^{\nu_2} \left(\mathsf{Adv}_{\mathcal{E}_{5\text{-}h}}^{\mathsf{DLIN}}(\lambda) + \mathsf{Adv}_{\mathcal{E}_{6\text{-}h}}^{\mathsf{H},\mathsf{CR}}(\lambda)\right) + \epsilon,\end{aligned}$$

where $\mathcal{E}_{\iota\text{-}i}(\cdot) := \mathcal{E}_\iota(i, \cdot)$ for $\iota = 1, 2$ ($i = 1, 2$), $\mathcal{E}_{\iota\text{-}h}(\cdot) := \mathcal{E}_\iota(h, \cdot)$ for $\iota = 5, 6$ ($h = 1, \ldots, \nu_2$), $\mathcal{E}_{3\text{-}h\text{-}\iota}(\cdot) := \mathcal{E}_{3\text{-}\iota}(h, \cdot)$ for $\iota = 1, 4$, $\mathcal{E}_{3\text{-}h\text{-}\iota\text{-}i}(\cdot) := \mathcal{E}_{3\text{-}\iota}(h, i, \cdot)$ for $\iota = 2, 3$ ($h = 1, \ldots, \nu_1; i = 1, 2$), ν_1 is the maximum number of $\mathcal{A}$'s reveal key queries, ν_2 is the maximum number of $\mathcal{A}$'s reveal signature queries, and $\epsilon := ((2d + 16)\nu_1 + 8\nu_2 + 2d + 11)/q$.

The proofs of Theorems 1 and 2 (for a general form of our ABS) are given in Appendices D and E, respectively.

4.4 Performance

In this section, we compare the efficiency and security of the proposed ABS scheme with the existing ABS schemes in the standard model (two typical instantiations) [25] as well as the ABS scheme in the generic group model [24] (as a benchmark). Since all of these schemes can be implemented over a *prime order* pairing group, the size of a group element can be around the size of $\mathbb{F}_q$ (e.g., 256 bits). In Table 1, ℓ and r represent the size of the underlying access structure matrix M for a predicate, i.e., $M \in \mathbb{F}_q^{\ell \times r}$. For example, some predicate with 4 AND and 5 OR gates as well as 10 variables may be expressed by a 10×5 matrix, and a predicate with 49 AND and 50 OR gates as well as 100 variables may be expressed by a 100×50 matrix (see the appendix of [21]). λ is the security parameter (e.g., 128).

Table 1: Comparison with the Existing ABS Schemes

	MPR08 [24]	MPR10 [25] (Boneh-Boyen based)	MPR10 [25] (Waters based)	Proposed
Signature size (# of group elts)	$\ell + r + 2$	$51\ell + 2r + 18\lambda\ell$	$36\ell + 2r + 9\lambda + 12$	$9\ell + 11$
Model	generic group model	standard model	standard model	standard model
Security	full	full	full	full
Assumptions	CR hash	q-SDH and DLIN	DLIN	DLIN and CR hash
Predicates	monotone	monotone	monotone	non-monotone
Sig. size example 1 ($\ell = 10, r = 5, \lambda = 128$)	17	23560	1534	101
Sig. size example 2 ($\ell = 100, r = 50, \lambda = 128$)	152	282400	4864	911

5 Multi-Authority ABS (MA-ABS)

5.1 Definitions and Security of MA-ABS

Definition 10 (Multi-Authority ABS : MA-ABS) *A multi-authority ABS scheme consists of the following algorithms/protocols.*

TSetup *This is a randomized algorithm. The signature trustee runs algorithm* $\mathsf{TSetup}(1^\lambda)$ *which outputs trustee public key* tpk *and trustee secret key* tsk.

UserReg *This is a randomized algorithm. When a user with user id* uid *registers with the signature trustee, the trustee runs* $\mathsf{UserReg}(\mathsf{tpk}, \mathsf{tsk}, \mathsf{uid})$ *which outputs public user-token* $\mathsf{token}_{\mathsf{uid}}$. *The trustee gives* $\mathsf{token}_{\mathsf{uid}}$ *to the user.*

ASetup *This is a randomized algorithm. Attribute authority* t $(1 \leq t \leq d)$ *who wishes to issue attributes runs* $\mathsf{ASetup}(\mathsf{tpk})$ *which outputs attribute-authority public key* apk_t *and attribute-authority secret key* ask_t. *The attribute authority,* t, *publishes* apk_t *and stores* ask_t.

AttrGen *This is a randomized algorithm. When attribute authority* t *issues user* uid *a secret key associated with attribute* x_t, *first it obtains (from the user) her user-token* $\mathsf{token}_{\mathsf{uid}}$, *and runs token verification algorithm* $\mathsf{TokenVerify}(\mathsf{tpk}, \mathsf{uid}, \mathsf{token}_{\mathsf{uid}})$. *If the token is verified, then it runs* $\mathsf{AttrGen}(\mathsf{tpk}, t, \mathsf{ask}_t, \mathsf{token}_{\mathsf{uid}}, x_t)$ *that outputs attribute secret key* usk_t. *The attribute authority gives* usk_t *to the user.*

Sig *This is a randomized algorithm. A user signs message* m *with claim-predicate (access structure)* $\mathbb{S} := (M, \rho)$, *only if there is a set of attributes* Γ *such that* $\mathbb{S}$ *accepts* Γ, *the user*

has obtained a set of keys $\{\mathsf{usk}_t \mid (t, x_t) \in \Gamma\}$ *from the attribute authorities. Then signature* σ *can be generated using* $\mathsf{Sig}(\mathsf{tpk}, \mathsf{token}_{\mathsf{uid}}, \{\mathsf{apk}_t, \mathsf{usk}_t \mid (t, x_t) \in \Gamma\}, m, \mathbb{S})$*, where* $\mathsf{usk}_t \xleftarrow{\mathsf{R}} \mathsf{AttrGen}(\mathsf{tpk}, t, \mathsf{ask}_t, \mathsf{token}_{\mathsf{uid}}, x_t)$*.*

Ver *To verify signature* σ *on message* m *with claim-predicate (access structure)* $\mathbb{S}$*, a user runs* $\mathsf{Ver}(\mathsf{tpk}, \{\mathsf{apk}_t\}, m, \mathbb{S}, \sigma)$ *which outputs boolean value* $\mathsf{accept} := 1$ *or* $\mathsf{reject} := 0$*.*

Definition 11 (Perfect Privacy of MA-ABS) *A MA-ABS scheme is perfectly private, if, for all* $(\mathsf{tsk}, \mathsf{tpk}) \xleftarrow{\mathsf{R}} \mathsf{TSetup}(1^\lambda)$*, all* uid_ι $(\iota = 1, 2)$*, all* $\mathsf{token}_{\mathsf{uid}_\iota} \xleftarrow{\mathsf{R}} \mathsf{UserReg}(\mathsf{tpk}, \mathsf{tsk}, \mathsf{uid}_\iota)$ $(\iota = 1, 2)$*, all* $(\mathsf{ask}_t, \mathsf{apk}_t) \xleftarrow{\mathsf{R}} \mathsf{ASetup}(\mathsf{tpk})$ $(1 \leq t \leq d)$*, all messages* m*, all attribute sets* Γ_ι *associated with* uid_ι $(\iota = 1, 2)$*, all signing keys* $\{\mathsf{usk}_{t,\iota} \xleftarrow{\mathsf{R}} \mathsf{AttrGen}(\mathsf{tpk}, t, \mathsf{ask}_t, \mathsf{token}_{\mathsf{uid}_\iota}, x_{t,\iota})\}_{(t,x_{t,\iota})\in\Gamma_\iota}\}$ $(\iota = 1, 2)$*, all access structures* $\mathbb{S}$ *such that* $\mathbb{S}$ *accepts* Γ_1 *and* $\mathbb{S}$ *accepts* Γ_2*, the distributions* $\mathsf{Sig}(\mathsf{tpk}, \mathsf{token}_{\mathsf{uid}_1}, \{\mathsf{apk}_t, \mathsf{usk}_{t,1} \mid (t, x_{t,1}) \in \Gamma_1\}, m, \mathbb{S})$ *and* $\mathsf{Sig}(\mathsf{tpk}, \mathsf{token}_{\mathsf{uid}_2}, \{\mathsf{apk}_t, \mathsf{usk}_{t,2} \mid (t, x_{t,2}) \in \Gamma_2\}, m, \mathbb{S})$ *are equal.*

Let T be the set of authorities. We assume each attribute is assigned to one authority.

Definition 12 (Unforgeability of MA-ABS) *For an adversary, we define* $\mathsf{Adv}^{\mathsf{MA\text{-}ABS,UF}}_{\mathcal{A}}(\lambda)$ *to be the success probability in the following experiment for any security parameter* λ*. A MA-ABS scheme is existentially unforgeable if the success probability of any polynomial-time adversary is negligible:*

1. *The challenger gives a trustee public key* tpk *to adversary* $\mathcal{A}$*, where* $(\mathsf{tpk}, \mathsf{tsk}) \xleftarrow{\mathsf{R}} \mathsf{TSetup}(1^\lambda)$*. Adversary* $\mathcal{A}$ *specifies a set* $\mathcal{T}_{\mathsf{bad}} \subseteq \mathcal{T} := \{1, \ldots, d\}$ *of corrupt attribute authorities (and good (non-corrupt) authorities* $\mathcal{T}_{\mathsf{good}} := \mathcal{T} \setminus \mathcal{T}_{\mathsf{bad}}$*). For good authorities* $t \in \mathcal{T}_{\mathsf{good}}$*, The challenger runs* $(\mathsf{ask}_t, \mathsf{apk}_t) \xleftarrow{\mathsf{R}} \mathsf{ASetup}(\mathsf{tpk})$ *for authority* $t \in \mathcal{T}_{\mathsf{good}}$ *and gives* $\{\mathsf{apk}_t\}_{t\in\mathcal{T}_{\mathsf{good}}}$ *to* $\mathcal{A}$*.*

2. $\mathcal{A}$ *may adaptively makes a polynomial number of queries of the following type:*

   - *[Create and reveal token]* $\mathcal{A}$ *asks the challenger to create a token for user id,* uid*. The challenger creates a token* $\mathsf{token}_{\mathsf{uid}} \xleftarrow{\mathsf{R}} \mathsf{UserReg}(\mathsf{tpk}, \mathsf{tsk}, \mathsf{uid})$ *and gives it to* $\mathcal{A}$*.*
   - *[Create key]* $\mathcal{A}$ *sends a token* $\mathsf{token}_{\mathsf{uid}}$ *that has already been created for* uid *(which is correctly verified), and asks the challenger to create a signing key for an attribute,* (t, x_t)*, for good* $t \in \mathcal{T}_{\mathsf{good}}$*. The challenger creates an attribute secret key* $\mathsf{usk}_{\mathsf{uid},(t,x_t)} \xleftarrow{\mathsf{R}} \mathsf{AttrGen}(\mathsf{tpk}, t, \mathsf{ask}_t, \mathsf{token}_{\mathsf{uid}}, x_t)$ *without giving it to* $\mathcal{A}$*.*
   - *[Create signature]* $\mathcal{A}$ *asks the challenger to perform a signing operation to create a signature for a message* m*, user id* uid*, and an access structure* $\mathbb{S}$ *that accepts* Γ*: For that,* $\mathcal{A}$ *sends a token* $\mathsf{token}_{\mathsf{uid}}$ *and keys* $\mathsf{usk}_{\mathsf{uid},(t,x_t)}$ *that has already been created for* uid *and attributes* (t, x_t) *with good* $t \in \mathcal{T}_{\mathsf{good}} \wedge (t, x_t) \in \Gamma$*, and provides corrupted authority public keys* apk_t *and attribute secret keys* $\mathsf{usk}_{\mathsf{uid},(t,x_t)}$ *for corrupt* $t \in \mathcal{T}_{\mathsf{bad}} \wedge (t, x_t) \in \Gamma$*. Using the above key* $\{\mathsf{usk}_{\mathsf{uid},(t,x_t)}\}_{(t,x_t)\in\Gamma}$*, the challenger computes the signature* $\sigma \xleftarrow{\mathsf{R}} \mathsf{Sig}(\mathsf{tpk}, \mathsf{token}_{\mathsf{uid}}, \{\mathsf{apk}_t, \mathsf{usk}_{\mathsf{uid},(t,x_t)}\}_{(t,x_t)\in\Gamma}, m, \mathbb{S})$ *without giving it to the adversary.*
   - *[Reveal key or signature]* $\mathcal{A}$ *asks the challenger to reveal an already-created key for a user id* uid *and an attribute* (t, x_t)*, or an already-created signature for a user id* uid*, attributes* $\Gamma := \{(t, x_t)\}$ *and an access structure* $\mathbb{S}$ *(where* Γ *satisfies* $\mathbb{S}$*).*

Note that when key or signature creation requests are made, $\mathcal{A}$ does not automatically see the created key or signature. $\mathcal{A}$ sees it only when it makes a reveal query.

3. *At the end, the adversary outputs $(m', \mathbb{S}', \sigma')$ and corrupted authority public keys $\{\mathsf{apk}_t\}_{t\in\mathcal{T}_{\mathsf{bad}}}$.*

Let $\Gamma_{\mathsf{uid}_i} := \{(t, x_t)\}$ $(i = 1, \ldots, \nu_1)$ be attributes under uid_i that have been revealed to the adversary, and $\Gamma_0 := \{(t, *)\}_{t\in\mathcal{T}_{\mathsf{bad}}}$, where $*$ denotes a wild card (an arbitrary value). We say the adversary succeeds, if a signature for $(m', \mathbb{S}')$ was never revealed to the adversary, $\mathbb{S}'$ does not accept $\Gamma_{\mathsf{uid}_i} \cup \Gamma_0$ for any uid_i $(i = 1, \ldots, \nu_1)$, and $\mathsf{Ver}(\mathsf{gparam}, \{\mathsf{apk}_t\}, m', \mathbb{S}', \sigma') = 1$.

5.2 Construction

The key idea of our construction of MA-ABS scheme is to share $G_{\mathsf{uid}} := \delta G_1$ as well as G_0 and G_1 among attribute authorities to generate $\delta \boldsymbol{b}^*_{t,i}$ by each authority t. Hence, G_0 and G_1 are included in tpk and $G_{\mathsf{uid}} := \delta G_1$ is shared with attribute authorities through the user's token $\mathsf{token}_{\mathsf{uid}}$.

For matrix $X := (\chi_{i,j})_{i,j=1,\ldots,N} \in \mathbb{F}_q^{N\times N}$ and element $\boldsymbol{v}$ in N-dimensional $\mathbb{V}$, $\boldsymbol{v}X$ denotes matrix multiplication of $\boldsymbol{v}$ and X (Remark 1 in Section 2.1). It holds then that $e(\boldsymbol{x}X, \boldsymbol{y}(X^{-1})^{\mathrm{T}}) = e(\boldsymbol{x}, \boldsymbol{y})$ for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{V}$.

Moreover, $(\mathsf{G}_{\mathsf{SIG}}, \mathsf{S}, \mathsf{V})$ is a (conventional) unforgeable signature scheme.

$\mathsf{TSetup}(1^\lambda)$: $\mathsf{param}_{\mathbb{G}} := (q, \mathbb{G}, \mathbb{G}_T, G, e) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{bpg}}(1^\lambda)$,
$\mathsf{hk} \xleftarrow{\mathsf{R}} \mathsf{KH}_\lambda$, $(\mathsf{verk}, \mathsf{sigk}) \xleftarrow{\mathsf{R}} \mathsf{G}_{\mathsf{SIG}}(1^\lambda)$ $N_0 := 4$, $N_{d+1} := 7$, $\kappa, \xi \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times$,
for $t = 0, d+1$, $\mathsf{param}_{\mathbb{V}_t} := (q, \mathbb{V}_t, \mathbb{G}_T, \mathbb{A}_t, e) := \mathcal{G}_{\mathsf{dpvs}}(1^\lambda, N_t, \mathsf{param}_{\mathbb{G}})$,
$X_t := (\chi_{t,i,j})_{i,j} \xleftarrow{\mathsf{U}} GL(N_t, \mathbb{F}_q)$, $(\vartheta_{t,i,j})_{i,j} := (X_t^{-1})^{\mathrm{T}}$,
$\boldsymbol{b}_{t,i} := \kappa(\chi_{t,i,1}, \ldots, \chi_{t,i,N_t})_{\mathbb{A}_t}$, $\mathbb{B}_t := (\boldsymbol{b}_{t,1}, \ldots, \boldsymbol{b}_{t,N_t})$,
$\boldsymbol{b}^*_{t,i} := \xi(\vartheta_{t,i,1}, \ldots, \vartheta_{t,i,N_t})_{\mathbb{A}_t}$, $\mathbb{B}^*_t := (\boldsymbol{b}^*_{t,1}, \ldots, \boldsymbol{b}^*_{t,N_t})$,
$G_0 := \kappa G$, $G_1 := \xi G$, $g_T := e(G, G)^{\kappa\xi}$,
$\widehat{\mathbb{B}}_0 := (\boldsymbol{b}_{0,1}, \boldsymbol{b}_{0,4})$, $\widehat{\mathbb{B}}_{d+1} := (\boldsymbol{b}_{d+1,1}, \boldsymbol{b}_{d+1,2}, \boldsymbol{b}_{d+1,7})$,
$\widehat{\mathbb{B}}^*_{d+1} := (\boldsymbol{b}^*_{d+1,1}, \boldsymbol{b}^*_{d+1,2}, \boldsymbol{b}^*_{d+1,5}, \boldsymbol{b}^*_{d+1,6})$,
$\mathsf{tsk} := (\boldsymbol{b}^*_{0,1}, \mathsf{sigk})$,
$\mathsf{tpk} := (1^\lambda, \mathsf{hk}, \{\mathsf{param}_{\mathbb{V}_t}, \widehat{\mathbb{B}}_t\}_{t=0,d+1}, \boldsymbol{b}^*_{0,3}, \widehat{\mathbb{B}}^*_{d+1}, g_T, G_0, G_1, \mathsf{verk})$,
return $(\mathsf{tsk}, \mathsf{tpk})$.

$\mathsf{UserReg}(\mathsf{tpk}, \mathsf{tsk}, \mathsf{uid})$: $\delta \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times$, $\varphi_0, \varphi_{d+1,1,\iota}, \varphi_{d+1,2,\iota} \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $G_{\mathsf{uid}} := \delta G_1$,
$\boldsymbol{k}^*_0 := (\ \delta,\ 0,\ \varphi_0,\ 0\ )_{\mathbb{B}^*_0}$,
$\boldsymbol{k}^*_{d+1,1} := (\ \delta(1,\ 0),\ \ 0,\ 0,\ \ \varphi_{d+1,1,1},\ \varphi_{d+1,1,2},\ \ 0\ )_{\mathbb{B}^*_{d+1}}$,
$\boldsymbol{k}^*_{d+1,2} := (\ \delta(0,\ 1),\ \ 0,\ 0,\ \ \varphi_{d+1,2,1},\ \varphi_{d+1,2,2},\ \ 0\ )_{\mathbb{B}^*_{d+1}}$,
$\mathsf{usk}_0 := (\boldsymbol{k}^*_0, \boldsymbol{k}^*_{d+1,1}, \boldsymbol{k}^*_{d+1,2})$, $\sigma_{\mathsf{uid}} := \mathsf{S}(\mathsf{sigk}, (\mathsf{uid}, G_{\mathsf{uid}}))$,
return $\mathsf{token}_{\mathsf{uid}} := (\mathsf{uid}, G_{\mathsf{uid}}, \sigma_{\mathsf{uid}}, \mathsf{usk}_0)$.

$\mathsf{ASetup}(\mathsf{tpk})$: $\boldsymbol{u}_{j,i} := (0^{i-1},\ G_j,\ 0^{9-i})$ for $j = 0, 1; i = 1, .., 9$, $X_t \xleftarrow{\mathsf{U}} GL(9, \mathbb{F}_q)$,
$\mathbb{B}_t := (\boldsymbol{b}_{t,i})_{i=1,\ldots,9} := (\boldsymbol{u}_{0,1}X_t, \ldots, \boldsymbol{u}_{0,9}X_t)$,
$\mathbb{B}^*_t := (\boldsymbol{b}^*_{t,i})_{i=1,\ldots,9} := (\boldsymbol{u}_{1,1}(X_t^{-1})^{\mathrm{T}}, \ldots, \boldsymbol{u}_{1,9}(X_t^{-1})^{\mathrm{T}})$,
$\widehat{\mathbb{B}}_t := (\boldsymbol{b}_{t,1}, \boldsymbol{b}_{t,2}, \boldsymbol{b}_{t,8}, \boldsymbol{b}_{t,9})$, $\widehat{\mathbb{B}}^*_t := (\boldsymbol{b}^*_{t,1}, \boldsymbol{b}^*_{t,2}, \boldsymbol{b}^*_{t,6}, \boldsymbol{b}^*_{t,7})$,
return $(\mathsf{ask}_t := X_t,\ \mathsf{apk}_t := (\widehat{\mathbb{B}}_t, \widehat{\mathbb{B}}^*_t))$.

$\mathsf{TokenVerify}(\mathsf{tpk}, \mathsf{uid}, \mathsf{token}_{\mathsf{uid}})$ holds iff $\mathsf{V}(\mathsf{verk}, (\mathsf{uid}, G_{\mathsf{uid}}), \sigma_{\mathsf{uid}}) = 1$.

$\mathsf{AttrGen}(\mathsf{tpk}, t, \mathsf{ask}_t, \mathsf{token}_{\mathsf{uid}}, x_t \in \mathbb{F}_q)$: $\varphi_{t,1}, \varphi_{t,2} \xleftarrow{\mathsf{U}} \mathbb{F}_q$,

$\boldsymbol{k}_t^* := (\ G_{\mathsf{uid}},\ x_t G_{\mathsf{uid}},\ \ 0,\ 0,\ 0,\ \ \varphi_{t,1} G_1,\ \varphi_{t,2} G_1,\ \ 0,\ 0\ )(X_t^{-1})^{\mathrm{T}}$,

that is, $\boldsymbol{k}_t^* = (\ \delta,\ \delta x_t,\ \ 0,\ 0,\ 0,\ \ \varphi_{t,1},\ \varphi_{t,2},\ \ 0,\ 0\ )_{\mathbb{B}_t^*}$,

return $\mathsf{usk}_t := \boldsymbol{k}_t^*$.

$\mathsf{Sig}(\mathsf{tpk},\ \mathsf{token}_{\mathsf{uid}}, \{\mathsf{apk}_t, \mathsf{usk}_t \xleftarrow{\mathsf{R}} \mathsf{AttrGen}(\mathsf{tpk}, t, \mathsf{ask}_t, \mathsf{token}_{\mathsf{uid}}, x_t) \mid (t, x_t) \in \Gamma\}, m, \mathbb{S} := (M, \rho))$ and $\mathsf{Ver}(\mathsf{tpk},\ \{\mathsf{apk}_t\}_{t=1,\ldots,d},\ m,\ \mathbb{S} := (M, \rho), \vec{\boldsymbol{s}}^*)$ are essentially the same as those in Section 4.2.

5.3 Security

Theorem 3 *The proposed MA-ABS scheme is perfectly private.*

Theorem 4 *The proposed MA-ABS scheme is unforgeable (adaptive-predicate unforgeable) under the DLIN assumption and the existence of collision resistant hash functions.*

For any adversary $\mathcal{A}$, there exist probabilistic machines $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_{3\text{-}1}, \ldots, \mathcal{E}_{3\text{-}4}, \mathcal{E}_5, \mathcal{E}_6$, whose running times are essentially the same as that of $\mathcal{A}$, such that for any security parameter λ,

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{MA\text{-}ABS,UF}}(\lambda) \le{}& \textstyle\sum_{i=1}^{2} \left(\mathsf{Adv}_{\mathcal{E}_{1\text{-}i}}^{\mathsf{DLIN}}(\lambda) + \mathsf{Adv}_{\mathcal{E}_{2\text{-}i}}^{\mathsf{DLIN}}(\lambda)\right) \\ &+ \textstyle\sum_{h=1}^{\nu_1} \left(\mathsf{Adv}_{\mathcal{E}_{3\text{-}h\text{-}1}}^{\mathsf{DLIN}}(\lambda) + \sum_{i=1}^{2} \left(\mathsf{Adv}_{\mathcal{E}_{3\text{-}h\text{-}2\text{-}i}}^{\mathsf{DLIN}}(\lambda) + \mathsf{Adv}_{\mathcal{E}_{3\text{-}h\text{-}3\text{-}i}}^{\mathsf{DLIN}}(\lambda)\right) + \mathsf{Adv}_{\mathcal{E}_{3\text{-}h\text{-}4}}^{\mathsf{DLIN}}(\lambda)\right) \\ &+ \textstyle\sum_{h=1}^{\nu_2} \left(\mathsf{Adv}_{\mathcal{E}_{5\text{-}h}}^{\mathsf{DLIN}}(\lambda) + \mathsf{Adv}_{\mathcal{E}_{6\text{-}h}}^{\mathsf{H,CR}}(\lambda)\right) + \epsilon,\end{aligned}$$

where $\mathcal{E}_{\iota\text{-}i}(\cdot) := \mathcal{E}_\iota(i, \cdot)$ for $\iota = 1, 2$ ($i = 1, 2$), $\mathcal{E}_{\iota\text{-}h}(\cdot) := \mathcal{E}_\iota(h, \cdot)$ for $\iota = 5, 6$ ($h = 1, \ldots, \nu_2$), $\mathcal{E}_{3\text{-}h\text{-}\iota}(\cdot) := \mathcal{E}_{3\text{-}\iota}(h, \cdot)$ for $\iota = 1, 4$, $\mathcal{E}_{3\text{-}h\text{-}\iota\text{-}i}(\cdot) := \mathcal{E}_{3\text{-}\iota}(h, i, \cdot)$ for $\iota = 2, 3$ ($h = 1, \ldots, \nu_1; i = 1, 2$), ν_1 is the maximum number of $\mathcal{A}$'s token queries, ν_2 is the maximum number of $\mathcal{A}$'s reveal signature queries, and $\epsilon := ((2d + 16)\nu_1 + 8\nu_2 + 2d + 11)/q$.

The proofs of Theorems 3 and 4 are given in Appendix F.

References

[1] S. Agrawal, S. Gorbunov, V. Vaikuntanathan, and H. Wee. Functional encryption: New perspectives and lower bounds. *IACR Cryptology ePrint Archive*, 2012:468, 2012. To appear in CRYPTO 2013.

[2] A. Beimel. Secure schemes for secret sharing and key distribution. *PhD Thesis, Israel Institute of Technology, Technion, Haifa*, 1996.

[3] M. Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss, A. Lysyanskaya, and H. Shacham. Randomizable proofs and delegatable anonymous credentials. In Halevi [14], pages 108–125.

[4] M. Belenkiy, M. Chase, M. Kohlweiss, and A. Lysyanskaya. P-signatures and noninteractive anonymous credentials. In R. Canetti, editor, *TCC*, volume 4948 of *Lecture Notes in Computer Science*, pages 356–374. Springer, 2008.

[5] D. Boneh and M. Hamburg. Generalized identity based and broadcast encryption schemes. In J. Pieprzyk, editor, *ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 455–470. Springer, 2008.

[6] X. Boyen. Mesh signatures. In M. Naor, editor, *EUROCRYPT*, volume 4515 of *Lecture Notes in Computer Science*, pages 210–227. Springer, 2007.

[7] J. Camenisch and T. Groß. Efficient attributes for anonymous credentials. In P. Ning, P. F. Syverson, and S. Jha, editors, *ACM Conference on Computer and Communications Security*, pages 345–356. ACM, 2008.

[8] J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In B. Pfitzmann, editor, *EUROCRYPT*, volume 2045 of *Lecture Notes in Computer Science*, pages 93–118. Springer, 2001.

[9] J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In M. K. Franklin, editor, *CRYPTO*, volume 3152 of *LNCS*, pages 56–72. Springer, 2004.

[10] D. Chaum. Security without identification: Transaction systems to make big brother obsolete. *Commun. ACM*, 28(10):1030–1044, 1985.

[11] A. Escala, J. Herranz, and P. Morillo. Revocable attribute-based signatures with adaptive security in the standard model. In A. Nitaj and D. Pointcheval, editors, *AFRICACRYPT*, volume 6737 of *Lecture Notes in Computer Science*, pages 224–241. Springer, 2011.

[12] J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In Smart [35], pages 415–432.

[13] S. Guo and Y. Zeng. Attribute-based signature scheme. In *ISA*, pages 509–511. IEEE, 2008.

[14] S. Halevi, editor. *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *LNCS*. Springer, 2009.

[15] J. Herranz, F. Laguillaumie, B. Libert, and C. Ràfols. Short attribute-based signatures for threshold predicates. In O. Dunkelman, editor, *CT-RSA*, volume 7178 of *Lecture Notes in Computer Science*, pages 51–67. Springer, 2012.

[16] J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In Smart [35], pages 146–162.

[17] D. Khader. Attribute based group signature with revocation. *IACR Cryptology ePrint Archive*, 2007:241, 2007.

[18] D. Khader. Attribute based group signatures. *IACR Cryptology ePrint Archive*, 2007:159, 2007.

[19] A. Lewko. Functional encryption: New proof techniques and advancing capabilities. *PhD Thesis, The university of Texas at Austin*, May 2012. `http://www.cs.utexas.edu/~alewko/`.

[20] A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In H. Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 62–91. Springer, 2010. Full version is available at `http://eprint.iacr.org/2010/110`.

[21] A. B. Lewko and B. Waters. Decentralizing attribute-based encryption. In K. G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 568–588. Springer, 2011.

[22] J. Li, M. H. Au, W. Susilo, D. Xie, and K. Ren. Attribute-based signature and its applications. In D. Feng, D. A. Basin, and P. Liu, editors, *ASIACCS*, pages 60–69. ACM, 2010.

[23] J. Li and K. Kim. Attribute-based ring signatures. *IACR Cryptology ePrint Archive*, 2008:394, 2008.

[24] H. K. Maji, M. Prabhakaran, and M. Rosulek. Attribute-based signatures: Achieving attribute-privacy and collusion-resistance. *IACR Cryptology ePrint Archive*, 2008:328, 2008.

[25] H. K. Maji, M. Prabhakaran, and M. Rosulek. Attribute-based signatures. In A. Kiayias, editor, *CT-RSA*, volume 6558 of *Lecture Notes in Computer Science*, pages 376–392. Springer, 2011.

[26] T. Okamoto and K. Takashima. Homomorphic encryption and signatures from vector decomposition. In S. D. Galbraith and K. G. Paterson, editors, *Pairing 2008*, volume 5209 of *LNCS*, pages 57–74. Springer, 2008.

[27] T. Okamoto and K. Takashima. Hierarchical predicate encryption for inner-products. In M. Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 214–231. Springer, 2009.

[28] T. Okamoto and K. Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In T. Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 191–208. Springer, 2010. Full version is available at `http://eprint.iacr.org/2010/563`.

[29] T. Okamoto and K. Takashima. Efficient attribute-based signatures for non-monotone predicates in the standard model. In D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, editors, *Public Key Cryptography*, volume 6571 of *LNCS*, pages 35–52. Springer, 2011.

[30] T. Okamoto and K. Takashima. Efficient attribute-based signatures for non-monotone predicates in the standard model. *To appear in IEEE Trans. Cloud Computing*, 2014.

[31] R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In C. Boyd, editor, *ASIACRYPT*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565. Springer, 2001.

[32] S. F. Shahandashti and R. Safavi-Naini. Threshold attribute-based signatures and their application to anonymous credential systems. In B. Preneel, editor, *AFRICACRYPT*, volume 5580 of *Lecture Notes in Computer Science*, pages 198–216. Springer, 2009.

[33] A. Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and D. Chaum, editors, *CRYPTO*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer, 1984.

[34] E. Shi and B. Waters. Delegating capabilities in predicate encryption systems. In L. Aceto, I. Damgård, L. A. Goldberg, M. M. Halldórsson, A. Ingólfsdóttir, and I. Walukiewicz, editors, *ICALP (2) 2008*, volume 5126 of *LNCS*, pages 560–578. Springer, 2008.

[35] N. P. Smart, editor. *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*, volume 4965 of *LNCS*. Springer, 2008.

[36] B. Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In Halevi [14], pages 619–636.

[37] P. Yang, Z. Cao, and X. Dong. Fuzzy identity based signature. *IACR Cryptology ePrint Archive*, 2008:2, 2008.

A Dual Pairing Vector Spaces (DPVS)

A.1 Summary

We now briefly explain our approach, DPVS, constructed on symmetric pairing groups $(q, \mathbb{G}, \mathbb{G}_T, G, e)$, where q is a prime, $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order q, G is a generator of $\mathbb{G}$, $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a non-degenerate bilinear pairing operation, and $e(G, G) \neq 1$. Here we denote the group operation of $\mathbb{G}$ by addition and $\mathbb{G}_T$ by multiplication, respectively. Note that this construction also works on *asymmetric* pairing groups (in this paper, we use symmetric pairing groups for simplicity of description).

Vector space $\mathbb{V}$: $\mathbb{V} := \overbrace{\mathbb{G} \times \cdots \times \mathbb{G}}^{N}$, whose element is expressed by N-dimensional vector, $\boldsymbol{x} := (x_1 G, \ldots, x_N G)$ ($x_i \in \mathbb{F}_q$ for $i = 1, \ldots, N$).

Canonical base $\mathbb{A}$: $\mathbb{A} := (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N)$ of $\mathbb{V}$, where $\boldsymbol{a}_1 := (G, 0, \ldots, 0)$, $\boldsymbol{a}_2 := (0, G, 0, \ldots, 0)$, $\ldots, \boldsymbol{a}_N := (0, \ldots, 0, G)$.

Pairing operation: $e(\boldsymbol{x}, \boldsymbol{y}) := \prod_{i=1}^{N} e(x_i G, y_i G) = e(G, G)^{\sum_{i=1}^{N} x_i y_i} = e(G, G)^{\vec{x} \cdot \vec{y}} \in \mathbb{G}_T$, where $\boldsymbol{x} := (x_1 G, \ldots, x_N G) = x_1 \boldsymbol{a}_1 + \cdots + x_N \boldsymbol{a}_N \in \mathbb{V}$, $\boldsymbol{y} := (y_1 G, \ldots, y_N G) = y_1 \boldsymbol{a}_1 + \cdots + y_N \boldsymbol{a}_N \in \mathbb{V}$, $\vec{x} := (x_1, \ldots, x_N)$ and $\vec{y} := (y_1, \ldots, y_N)$. Here, $\boldsymbol{x}$ and $\boldsymbol{y}$ can be expressed by coefficient vector over basis $\mathbb{A}$ such that $(x_1, \ldots, x_N)_{\mathbb{A}} = (\vec{x})_{\mathbb{A}} := \boldsymbol{x}$ and $(y_1, \ldots, y_N)_{\mathbb{A}} = (\vec{y})_{\mathbb{A}} := \boldsymbol{y}$.

Base change: Canonical basis $\mathbb{A}$ is changed to basis $\mathbb{B} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_N)$ of $\mathbb{V}$ using a uniformly chosen (regular) linear transformation, $X := (\chi_{i,j}) \xleftarrow{\cup} GL(N, \mathbb{F}_q)$, such that $\boldsymbol{b}_i = \sum_{j=1}^{N} \chi_{i,j} \boldsymbol{a}_j$, $(i = 1, \ldots, N)$. $\mathbb{A}$ is also changed to basis $\mathbb{B}^* := (\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_N^*)$ of $\mathbb{V}$, such that $(\vartheta_{i,j}) := (X^T)^{-1}$, $\boldsymbol{b}_i^* = \sum_{j=1}^{N} \vartheta_{i,j} \boldsymbol{a}_j$, $(i = 1, \ldots, N)$. We see that $e(\boldsymbol{b}_i, \boldsymbol{b}_j^*) = e(G, G)^{\delta_{i,j}}$, ($\delta_{i,j} = 1$ if $i = j$, and $\delta_{i,j} = 0$ if $i \neq j$) i.e., $\mathbb{B}$ and $\mathbb{B}^*$ are dual orthonormal bases of $\mathbb{V}$.

Here, $\boldsymbol{x} := x_1 \boldsymbol{b}_1 + \cdots + x_N \boldsymbol{b}_N \in \mathbb{V}$ and $\boldsymbol{y} := y_1 \boldsymbol{b}_1^* + \cdots + y_N \boldsymbol{b}_N^* \in \mathbb{V}$ can be expressed by coefficient vectors over $\mathbb{B}$ and $\mathbb{B}^*$ such that $(x_1, \ldots, x_N)_{\mathbb{B}} = (\vec{x})_{\mathbb{B}} := \boldsymbol{x}$ and $(y_1, \ldots, y_N)_{\mathbb{B}^*} = (\vec{y})_{\mathbb{B}^*} := \boldsymbol{y}$, and $e(\boldsymbol{x}, \boldsymbol{y}) = e(G, G)^{\sum_{i=1}^{N} x_i y_i} = e(G, G)^{\vec{x} \cdot \vec{y}} \in \mathbb{G}_T$.

Intractable problem: One of the most natural decisional problems in this approach is the decisional subspace problem [26]. It is to tell $\boldsymbol{v} := v_{N_2+1} \boldsymbol{b}_{N_2+1} + \cdots + v_{N_1} \boldsymbol{b}_{N_1}$ $(= (0, \ldots, 0, v_{N_2+1}, \ldots, v_{N_1})_{\mathbb{B}})$, from $\boldsymbol{u} := v_1 \boldsymbol{b}_1 + \cdots + v_{N_1} \boldsymbol{b}_{N_1}$ $(= (v_1, \ldots, v_{N_1})_{\mathbb{B}})$, where $(v_1, \ldots, v_{N_1}) \xleftarrow{\cup} {\mathbb{F}_q}^{N_1}$ and $N_2 + 1 < N_1$.

Trapdoor: Although the decisional subspace problem is assumed to be intractable, it can be efficiently solved by using *trapdoor* $\boldsymbol{t}^* \in \mathsf{span}\langle \boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_{N_2}^* \rangle$. Given $\boldsymbol{v} := v_{N_2+1}\boldsymbol{b}_{N_2+1} + \cdots + v_{N_1}\boldsymbol{b}_{N_1}$ or $\boldsymbol{u} := v_1\boldsymbol{b}_1 + \cdots + v_{N_1}\boldsymbol{b}_{N_1}$, we can tell $\boldsymbol{v}$ from $\boldsymbol{u}$ using $\boldsymbol{t}^*$ since $e(\boldsymbol{v}, \boldsymbol{t}^*) = 1$ and $e(\boldsymbol{u}, \boldsymbol{t}^*) \neq 1$ with high probability.

Advantage of this approach: Higher dimensional vector treatment of bilinear pairing groups have been already employed in literature especially in the areas of IBE, ABE and BE (e.g., [5, 12]). For example, in a typical vector treatment, two vector forms of $P := (x_1G, \ldots, x_NG)$ and $Q := (y_1G, \ldots, y_NG)$ are set and pairing for P and Q is operated as $e(P, Q) := \prod_{i=1}^{N} e(x_iG, y_iG)$. Such treatment can be rephrased in this approach such that $P = x_1\boldsymbol{a}_1 + \cdots + x_N\boldsymbol{a}_N$ $(= (x_1, \ldots, x_N)_{\mathbb{A}})$, and $Q = y_1\boldsymbol{a}_1 + \cdots + y_N\boldsymbol{a}_N$ $(= (y_1, \ldots, y_N)_{\mathbb{A}})$ over canonical basis $\mathbb{A}$.

The major drawback of this approach is the easily *decomposable* property over $\mathbb{A}$ (i.e., the decisional subspace problem is easily solved). That is, it is easy to decompose $x_i\boldsymbol{a}_i = (0, \ldots, 0, x_iG, 0, \ldots, 0)$ from $P := x_1\boldsymbol{a}_1 + \cdots x_N\boldsymbol{a}_N = (x_1G, \ldots, x_NG)$.

In contrast, our approach employs basis $\mathbb{B}$, which is linearly transformed from $\mathbb{A}$ using a secret random matrix $X \in {\mathbb{F}_q}^{n \times n}$. A remarkable property over $\mathbb{B}$ is that it seems hard to decompose $x_i\boldsymbol{b}_i$ from $P' := x_1\boldsymbol{b}_1 + \cdots x_N\boldsymbol{b}_N$ (and the decisional subspace problem seems intractable). In addition, the secret matrix X (and the dual orthonormal basis $\mathbb{B}^*$ of $\mathbb{V}$) can be used as a source of the trapdoors to the decomposability (and distinguishability for the decisional subspace problem through the pairing operation over $\mathbb{B}$ and $\mathbb{B}^*$ as mentioned above). The hard decomposability (and indistinguishability) and its trapdoors are ones of the key tricks in this paper. Note that composite order pairing groups are often employed with similar tricks such as hard decomposability (and indistinguishability) of a composite order group to the prime order subgroups and its trapdoors through factoring (e.g., [16, 34]).

A.2 Dual Pairing Vector Spaces by Direct Product of Asymmetric Pairing Groups

Definition 13 *"Asymmetric bilinear pairing groups"* $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, G_1, G_2, e)$ *are a tuple of a prime* q*, cyclic additive groups* $\mathbb{G}_1, \mathbb{G}_2$ *and multiplicative group* $\mathbb{G}_T$ *of order* q*,* $G_1 \neq 0 \in \mathbb{G}_1, G_2 \neq 0 \in \mathbb{G}_2$*, and a polynomial-time computable nondegenerate bilinear pairing* $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ *i.e.,* $e(sG_1, tG_2) = e(G_1, G_2)^{st}$ *and* $e(G_1, G_2) \neq 1$*.*

Let $\mathcal{G}_{\mathsf{bpg}}$ *be an algorithm that takes input* 1^λ *and outputs a description of bilinear pairing groups* $\mathsf{param}_{\mathbb{G}} := (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, G_1, G_2, e)$ *with security parameter* λ*.*

Definition 14 *"Dual pairing vector spaces (DPVS)"* $(q, \mathbb{V}, \mathbb{V}^*, \mathbb{G}_T, \mathbb{A}, \mathbb{A}^*, e)$ *by direct product of asymmetric pairing groups* $\mathsf{param}_{\mathbb{G}} := (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, G_1, G_2, e)$ *are a tuple of a prime* q*, two* N*-dimensional vector spaces* $\mathbb{V} := \overbrace{\mathbb{G}_1 \times \cdots \times \mathbb{G}_1}^{N}$ *and* $\mathbb{V}^* := \overbrace{\mathbb{G}_2 \times \cdots \times \mathbb{G}_2}^{N}$ *over* $\mathbb{F}_q$*, a cyclic group* $\mathbb{G}_T$ *of order* q*, and their canonical bases i.e.,* $\mathbb{A} := (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N)$ *of* $\mathbb{V}$ *and* $\mathbb{A}^* := (\boldsymbol{a}_1^*, \ldots, \boldsymbol{a}_N^*)$ *of* $\mathbb{V}^*$*, where* $\boldsymbol{a}_i := (\overbrace{0, \ldots, 0}^{i-1}, G_1, \overbrace{0, \ldots, 0}^{N-i})$ *and* $\boldsymbol{a}_i^* := (\overbrace{0, \ldots, 0}^{i-1}, G_2, \overbrace{0, \ldots, 0}^{N-i})$*, and pairing* $e : \mathbb{V} \times \mathbb{V}^* \to \mathbb{G}_T$*.*

The pairing is defined by $e(\boldsymbol{x}, \boldsymbol{y}) := \prod_{i=1}^{N} e(D_i, H_i) \in \mathbb{G}_T$ *where* $\boldsymbol{x} := (D_1, \ldots, D_N) \in \mathbb{V}$ *and* $\boldsymbol{y} := (H_1, \ldots, H_N) \in \mathbb{V}^*$*. This is nondegenerate bilinear i.e.,* $e(s\boldsymbol{x}, t\boldsymbol{y}) = e(\boldsymbol{x}, \boldsymbol{y})^{st}$ *and if* $e(\boldsymbol{x}, \boldsymbol{y}) = 1$ *for all* $\boldsymbol{y} \in \mathbb{V}$*, then* $\boldsymbol{x} = \boldsymbol{0}$*. For all* i *and* j*,* $e(\boldsymbol{a}_i, \boldsymbol{a}_j^*) = g_T^{\delta_{i,j}}$ *where* $\delta_{i,j} = 1$ *if* $i = j$*, and* 0 *otherwise, and* $e(G_1, G_2) \neq 1 \in \mathbb{G}_T$*.*

DPVS generation algorithm $\mathcal{G}_{\mathsf{dpvs}}$ *takes input* 1^λ *(*$\lambda \in \mathbb{N}$*),* $N \in \mathbb{N}$ *and a description of bilinear pairing groups* $\mathsf{param}_{\mathbb{G}}$*, and outputs a description of* $\mathsf{param}_{\mathbb{V}} := (q, \mathbb{V}, \mathbb{V}^*, \mathbb{G}_T, \mathbb{A}, \mathbb{A}^*, e)$ *constructed above with security parameter* λ *and* N*-dimensional* $(\mathbb{V}, \mathbb{V}^*)$*.*

Right multiplication by $W \in GL(N, \mathbb{F}_q)$ is defined as in Remark 1 in Section 2.1.

B Anonymous Credentials

The notion of anonymous credentials (ACs) [3, 4, 7, 8, 9, 10] provides a functionality for users to demonstrate anonymously possession of attributes, but the goals of ACs and ABS differ in several points.

First of all, ABS is a class of signatures, which are non-interactive primitives and can be used as transferable digital evidence, while ACs are typically (non-transferable) interactive protocols to prove the possession of credentials. Nevertheless, chosen-message-attack secure signatures can be employed to construct an interactive protocol by signing a random number challenge from a verifier, and non-interactive ACs [4] have been proposed. So, we will focus on the other properties of ABS and ACs rather than the difference in signatures and interactive protocols.

Although the basic ABS is in the single-authority setting, we often consider a multi-authority (MA) setting of ABS (see the last item of Section 1.2 and Section 5), and AC also considers multiple authorities. So in this comparison we will use the MA settings of ABS and AC.

The first difference between ABS and ACs is the number of attributes for which an attribute authority is responsible. In MA-ABS, each authority can issue credentials (or keys) to users for an unbounded number of attributes (e.g., $q = O(2^\lambda)$ many attributes, where λ is the security parameter), and a user reveals only a predicate on the attributes that the user possesses, rather than the individual attributes themselves. In contrast, an authority in ACs is typically considered to be responsible for only a single attribute. Therefore, the public key size increases linearly with the number of attributes in ACs, while the size in MA-ABS increases with the number of authorities. Camenisch and Groß [7] introduce an AC system with an unbounded number of attributes for an authority, but the admissible predicates are limited to a single level of disjunctions or conjunctions of attributes, whereas more general predicates are typically available in ABS.

The second difference is the anonymity when a user registers with multiple authorities (or requests multiple authorities to issue credentials/keys with attributes). In ACs the multiple registrations of a user cannot be linked to each other, while they can be linked in MA-ABS schemes. For example, in the MA-ABS in Section 5, a user provides a token (a kind of identity for a user) to multiple authorities. However, this information in the registration stage is the only information that MA-ABS leaks, and no privacy is revealed after the registration stage, e.g., even colluding authorities cannot identify the user when a user proves some predicate on the credentials in MA-ABS. This provides sufficient anonymity in many applications.

As a summary, ACs and ABS aim at different goals: ACs target very strong anonymity even in the registration phase, whereas under less demanding anonymity requirements in the registration phase, ABS aims to achieve more expressive functionalities, more efficient constructions and new applications. In addition, ABS is a signature scheme and a simpler primitive compared with ACs.

C General Form of the Proposed ABS Scheme

This section provides a general form description of the proposed ABS scheme, while Section 4 describes a simpler form of the ABS scheme.

The security proof of the proposed ABS scheme will be given in this appendix for the general form of the ABS scheme.

In the description of the scheme, we assume that an input vector, $\vec{x}_t := (x_{t,1}, \ldots, x_{t,n_t})$, is normalized such that $x_{t,1} := 1$. (If $\vec{x}_t$ is not normalized, change it to a normalized one by $(1/x_{t,1}) \cdot \vec{x}_t$, assuming that $x_{t,1}$ is non-zero). In addition, we assume that input vector $\vec{v}_i := (v_{i,1}, \ldots, v_{i,n_t})$ satisfies that $v_{i,n_t} \neq 0$. We refer to Section 1.5 for notations on DPVS.

We describe random dual orthonormal basis generator $\mathcal{G}_{\mathsf{ob}}$ below, which is used as a subroutine in the proposed ABS scheme.

$$
\begin{array}{l}
\mathcal{G}_{\mathsf{ob}}(1^\lambda, \vec{n} := (d; n_1, \ldots, n_d)) : \mathsf{param}_{\mathbb{G}} := (q, \mathbb{G}, \mathbb{G}_T, G, e) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{bpg}}(1^\lambda), \quad \psi \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times, \\
\quad N_0 := 4, \quad N_t := 3n_t + 3 \ \text{ for } t = 1, \ldots, d, \quad N_{d+1} := 7, \\
\quad \text{for } t = 0, \ldots, d+1, \quad \mathsf{param}_{\mathbb{V}_t} := (q, \mathbb{V}_t, \mathbb{G}_T, \mathbb{A}_t, e) := \mathcal{G}_{\mathsf{dpvs}}(1^\lambda, N_t, \mathsf{param}_{\mathbb{G}}), \\
\quad\quad X_t := (\chi_{t,i,j})_{i,j} \xleftarrow{\mathsf{U}} GL(N_t, \mathbb{F}_q), \ (\vartheta_{t,i,j})_{i,j} := \psi \cdot (X_t^{\mathrm{T}})^{-1}, \\
\quad\quad \boldsymbol{b}_{t,i} := (\chi_{t,i,1}, \ldots, \chi_{t,i,N_t})_{\mathbb{A}_t} = \sum_{j=1}^{N_t} \chi_{t,i,j} \boldsymbol{a}_{t,j}, \ \mathbb{B}_t := (\boldsymbol{b}_{t,1}, \ldots, \boldsymbol{b}_{t,N_t}), \\
\quad\quad \boldsymbol{b}^*_{t,i} := (\vartheta_{t,i,1}, \ldots, \vartheta_{t,i,N_t})_{\mathbb{A}_t} = \sum_{j=1}^{N_t} \vartheta_{t,i,j} \boldsymbol{a}_{t,j}, \ \mathbb{B}^*_t := (\boldsymbol{b}^*_{t,1}, \ldots, \boldsymbol{b}^*_{t,N_t}), \\
\quad g_T := e(G, G)^\psi, \quad \mathsf{param}_{\vec{n}} := (\{\mathsf{param}_{\mathbb{V}_t}\}_{t=0,\ldots,d+1}, \ g_T) \\
\quad \mathsf{return} \ \ (\mathsf{param}_{\vec{n}}, \{\mathbb{B}_t, \mathbb{B}^*_t\}_{t=0,\ldots,d+1}).
\end{array}
$$

We note that $g_T = e(\boldsymbol{b}_{t,i}, \boldsymbol{b}^*_{t,i})$ for $t = 0, \ldots, d+1; i = 1, \ldots, N_t$.

$$
\begin{array}{l}
\mathsf{Setup}(1^\lambda, \ \vec{n} := (d; n_1, \ldots, n_d)) : \\
\quad \mathsf{hk} \xleftarrow{\mathsf{R}} \mathsf{KH}_\lambda, \quad (\mathsf{param}_{\vec{n}}, \{\mathbb{B}_t, \mathbb{B}^*_t\}_{t=0,\ldots,d+1}) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}(1^\lambda, \vec{n}), \\
\quad \widehat{\mathbb{B}}_0 := (\boldsymbol{b}_{0,1}, \boldsymbol{b}_{0,4}), \ \ \widehat{\mathbb{B}}_t := (\boldsymbol{b}_{t,1}, \ldots, \boldsymbol{b}_{t,n_t}, \boldsymbol{b}_{t,3n_t+2}, \boldsymbol{b}_{t,3n_t+3}) \ \text{ for } t = 1, \ldots, d, \\
\quad \widehat{\mathbb{B}}_{d+1} := (\boldsymbol{b}_{d+1,1}, \boldsymbol{b}_{d+1,2}, \boldsymbol{b}_{d+1,7}), \\
\quad \widehat{\mathbb{B}}^*_t := (\boldsymbol{b}^*_{t,1}, \ldots, \boldsymbol{b}^*_{t,n_t}, \boldsymbol{b}^*_{t,2n_t+2}, \ldots, \boldsymbol{b}^*_{t,3n_t+1}) \ \text{ for } t = 1, \ldots, d, \\
\quad \widehat{\mathbb{B}}^*_{d+1} := (\boldsymbol{b}^*_{d+1,1}, \boldsymbol{b}^*_{d+1,2}, \boldsymbol{b}^*_{d+1,5}, \boldsymbol{b}^*_{d+1,6}), \\
\quad \text{return} \ \ \mathsf{sk} := \boldsymbol{b}^*_{0,1}, \ \mathsf{pk} := (1^\lambda, \mathsf{hk}, \mathsf{param}_{\vec{n}}, \{\widehat{\mathbb{B}}_t\}_{t=0,\ldots,d+1}, \{\widehat{\mathbb{B}}^*_t\}_{t=1,\ldots,d+1}, \boldsymbol{b}^*_{0,3}). \\
\mathsf{KeyGen}(\mathsf{pk}, \ \mathsf{sk}, \ \ \Gamma := \{(t, \vec{x}_t := (x_{t,1}, \ldots, x_{t,n_t}) \in {\mathbb{F}_q}^{n_t}) \mid 1 \leq t \leq d\}) : \\
\quad \delta \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times, \quad \varphi_0, \varphi_{t,\iota}, \varphi_{d+1,1,\iota}, \varphi_{d+1,2,\iota} \xleftarrow{\mathsf{U}} \mathbb{F}_q \ \text{ for } t = 1, \ldots, d; \ \iota = 1, \ldots, n_t; \\
\quad \boldsymbol{k}^*_0 := (\delta, 0, \varphi_0, 0)_{\mathbb{B}^*_0}, \\
\quad \boldsymbol{k}^*_t := \ (\ \overbrace{\delta(x_{t,1}, \ldots, x_{t,n_t})}^{n_t}, \ \overbrace{0^{n_t+1}}^{n_t+1}, \ \overbrace{\varphi_{t,1}, \ldots, \varphi_{t,n_t}}^{n_t}, \ \overbrace{0^2}^{2} \ )_{\mathbb{B}^*_t} \ \text{ for } (t, \vec{x}_t) \in \Gamma, \\
\quad \boldsymbol{k}^*_{d+1,1} := (\delta(1,0), 0, 0, \varphi_{d+1,1,1}, \varphi_{d+1,1,2}, 0)_{\mathbb{B}^*_{d+1}}, \\
\quad \boldsymbol{k}^*_{d+1,2} := (\delta(0,1), 0, 0, \varphi_{d+1,2,1}, \varphi_{d+1,2,2}, 0)_{\mathbb{B}^*_{d+1}}, \\
\quad T := \{0, (d+1, 1), (d+1, 2)\} \cup \{t \ \mid \ 1 \leq t \leq d, \ (t, \vec{x}_t) \in \Gamma\}, \\
\quad \text{return} \ \ \mathsf{sk}_\Gamma := (\Gamma, \{\boldsymbol{k}^*_t\}_{t \in T}). \\
\mathsf{Sig}(\mathsf{pk}, \ \mathsf{sk}_\Gamma, \ m, \ \mathbb{S} := (M, \rho)) : \ \ \text{If } \mathbb{S} := (M, \rho) \text{ accepts } \Gamma := \{(t, \vec{x}_t)\}, \\
\quad \text{then compute } I \text{ and } \{\alpha_i\}_{i \in I} \text{ such that } \ \sum_{i \in I} \alpha_i M_i = \vec{1}, \\
\quad \text{and } I \subseteq \ \{i \in \{1, \ldots, \ell\} \mid \ [\rho(i) = (t, \vec{v}_i) \ \wedge \ (t, \vec{x}_t) \in \Gamma \ \wedge \ \vec{v}_i \cdot \vec{x}_t = 0] \\
\qquad\qquad\qquad\qquad \vee \ \ [\rho(i) = \neg(t, \vec{v}_i) \ \wedge \ (t, \vec{x}_t) \in \Gamma \ \wedge \ \vec{v}_i \cdot \vec{x}_t \neq 0] \ \}, \\
\quad \xi \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times, \ \ (\beta_i) \xleftarrow{\mathsf{U}} \{(\beta_1, \ldots, \beta_\ell) \ \mid \ \sum_{i=1}^{\ell} \beta_i M_i = \vec{0}\},
\end{array}
$$

$\boldsymbol{s}_0^* := \xi \boldsymbol{k}_0^* + \boldsymbol{r}_0^*$, where $\boldsymbol{r}_0^* \xleftarrow{\mathsf{U}} \mathsf{span}\langle \boldsymbol{b}_{0,3}^* \rangle$,
$\boldsymbol{s}_i^* := \gamma_i \cdot \xi \boldsymbol{k}_t^* + \sum_{\iota=1}^{n_t} y_{i,\iota} \cdot \boldsymbol{b}_{t,\iota}^* + \boldsymbol{r}_i^*$, for $1 \leq i \leq \ell$,
where $\boldsymbol{r}_i^* \xleftarrow{\mathsf{U}} \mathsf{span}\langle \boldsymbol{b}_{t,2n_t+2}^*, \ldots, \boldsymbol{b}_{t,3n_t+1}^* \rangle$, and $\gamma_i, \vec{y}_i := (y_{i,1}, \ldots, y_{i,n_t})$ are defined as
if $i \in I \ \wedge \ \rho(i) = (t, \vec{v}_i)$, $\gamma_i := \alpha_i$, $\vec{y}_i \xleftarrow{\mathsf{U}} \{\vec{y}_i \mid \vec{y}_i \cdot \vec{v}_i = 0 \ \wedge \ y_{i,1} = \beta_i\}$,
if $i \in I \ \wedge \ \rho(i) = \neg(t, \vec{v}_i)$, $\gamma_i := \alpha_i / (\vec{v}_i \cdot \vec{x}_t)$, $\vec{y}_i \xleftarrow{\mathsf{U}} \{\vec{y}_i \mid \vec{y}_i \cdot \vec{v}_i = \beta_i\}$,
if $i \notin I \ \wedge \ \rho(i) = (t, \vec{v}_i)$, $\gamma_i := 0$, $\vec{y}_i \xleftarrow{\mathsf{U}} \{\vec{y}_i \mid \vec{y}_i \cdot \vec{v}_i = 0 \ \wedge \ y_{i,1} = \beta_i\}$,
if $i \notin I \ \wedge \ \rho(i) = \neg(t, \vec{v}_i)$, $\gamma_i := 0$, $\vec{y}_i \xleftarrow{\mathsf{U}} \{\vec{y}_i \mid \vec{y}_i \cdot \vec{v}_i = \beta_i\}$,
$\boldsymbol{s}_{\ell+1}^* := \xi(\boldsymbol{k}_{d+1,1}^* + \mathsf{H}_{\mathsf{hk}}^{\lambda,\mathsf{D}}(m \,||\, \mathbb{S}) \cdot \boldsymbol{k}_{d+1,2}^*) + \boldsymbol{r}_{\ell+1}^*$, where $\boldsymbol{r}_{\ell+1}^* \xleftarrow{\mathsf{U}} \mathsf{span}\langle \boldsymbol{b}_{d+1,5}^*, \boldsymbol{b}_{d+1,6}^* \rangle$,
return $\vec{\boldsymbol{s}}^* := (\boldsymbol{s}_0^*, \ldots, \boldsymbol{s}_{\ell+1}^*)$.

$\mathsf{Ver}(\mathsf{pk}, \ m, \ \mathbb{S} := (M, \rho), \vec{\boldsymbol{s}}^*)$:
$\vec{f} \xleftarrow{\mathsf{R}} {\mathbb{F}_q}^r$, $\vec{s}^{\mathrm{T}} := (s_1, \ldots, s_\ell)^{\mathrm{T}} := M \cdot \vec{f}^{\mathrm{T}}$, $s_0 := \vec{1} \cdot \vec{f}^{\mathrm{T}}$, $\eta_0, \eta_{\ell+1}, \theta_{\ell+1}, s_{\ell+1} \xleftarrow{\mathsf{U}} \mathbb{F}_q$,
$\boldsymbol{c}_0 := (-s_0 - s_{\ell+1}, 0, 0, \eta_0)_{\mathbb{B}_0}$,
for $1 \leq i \leq \ell$,
if $\rho(i) = (t, \vec{v}_i := (v_{i,1}, \ldots, v_{i,n_t}) \in {\mathbb{F}_q}^{n_t})$,
return 0 if $\boldsymbol{s}_i^* \notin \mathbb{V}_t$, else $\theta_i, \eta_{i,1}, \eta_{i,2} \xleftarrow{\mathsf{U}} \mathbb{F}_q$,

$$\boldsymbol{c}_i := \Big(\overbrace{s_i + \theta_i v_{i,1}, \theta_i v_{i,2}, \ldots, \theta_i v_{i,n_t}}^{n_t}, \overbrace{0^{n_t+1}}^{n_t+1}, \overbrace{0^{n_t}}^{n_t}, \overbrace{\eta_{i,1}, \eta_{i,2}}^{2} \Big)_{\mathbb{B}_t},$$

if $\rho(i) = \neg(t, \vec{v}_i)$,
return 0 if $\boldsymbol{s}_i^* \notin \mathbb{V}_t$, else $\eta_{i,1}, \eta_{i,2} \xleftarrow{\mathsf{U}} \mathbb{F}_q$,

$$\boldsymbol{c}_i := \Big(\overbrace{s_i(v_{i,1}, \ldots, v_{i,n_t})}^{n_t}, \overbrace{0^{n_t+1}}^{n_t+1}, \overbrace{0^{n_t}}^{n_t}, \overbrace{\eta_{i,1}, \eta_{i,2}}^{2} \Big)_{\mathbb{B}_t},$$

$\boldsymbol{c}_{\ell+1} := (s_{\ell+1} - \theta_{\ell+1} \cdot \mathsf{H}_{\mathsf{hk}}^{\lambda,\mathsf{D}}(m \,||\, \mathbb{S}), \ \theta_{\ell+1}, 0, 0, 0, 0, \eta_{\ell+1})_{\mathbb{B}_{d+1}}$,
return 0 if $e(\boldsymbol{b}_{0,1}, \boldsymbol{s}_0^*) = 1$,
return 1 if $\prod_{i=0}^{\ell+1} e(\boldsymbol{c}_i, \boldsymbol{s}_i^*) = 1$, return 0 otherwise.

[Correctness]

$$\begin{aligned}\prod_{i=0}^{\ell+1} e(\boldsymbol{c}_i, \boldsymbol{s}_i^*) &= e(\boldsymbol{c}_0, \boldsymbol{k}_0^*)^{\xi} \cdot \prod_{i \in I} e(\boldsymbol{c}_i, \boldsymbol{k}_i^*)^{\gamma_i \xi} \cdot \prod_{i=1}^{\ell} \prod_{\iota=1}^{n_t} e(\boldsymbol{c}_i, \boldsymbol{b}_{t,\iota}^*)^{y_{i,\iota}} \cdot e(\boldsymbol{c}_{\ell+1}, \boldsymbol{k}_{\ell+1}^*) \\ &= g_T^{\xi\delta(-s_0+s_{\ell+1})} \cdot \prod_{i \in I} g_T^{\xi\delta\alpha_i s_i} \prod_{i=1}^{\ell} g_T^{\beta_i s_i} \cdot g_T^{-\xi\delta s_{\ell+1}} = g_T^{\xi\delta(-s_0+s_{\ell+1})} \cdot g_T^{\xi\delta s_0} \cdot g_T^{-\xi\delta s_{\ell+1}} = 1.\end{aligned}$$

D Proof of Theorem 1

Theorem 1 *The proposed ABS scheme is perfectly private.*

Proof. Before starting the proof, we first define function AltSig specified in the proposed ABS scheme as follows:

$\mathsf{AltSig}(\mathsf{pk}, \ \mathsf{sk}, \ m, \ \mathbb{S})$: $\widetilde{\delta} \xleftarrow{\mathsf{U}} \mathbb{F}_q^{\times}$, $\sigma_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q$,
if $Z := \{(\zeta_1, \ldots, \zeta_\ell) \mid \sum_{i=1}^{\ell} \zeta_i M_i = \vec{1}\} = \emptyset$, then return $\perp$,
otherwise, $(\zeta_i) \xleftarrow{\mathsf{U}} Z$, $\boldsymbol{s}_0^* := (\widetilde{\delta}, 0, \sigma_0, 0)_{\mathbb{B}_0^*}$,
for $i = 1, \ldots, \ell$,

$$\left.\begin{array}{l}\text{if } \rho(i)=(t,\vec{v}_i),\ \text{then } \vec{z}_i \xleftarrow{\mathsf{U}} \{\vec{z}_i := (z_{i,1},..,z_{i,n_t}) \ \mid\ \vec{z}_i\cdot\vec{v}_i=0,\ z_{i,1}=\widetilde{\delta}\zeta_i\},\\ \text{if } \rho(i)=\neg(t,\vec{v}_i),\ \text{then } \vec{z}_i \xleftarrow{\mathsf{U}} \{\vec{z}_i := (z_{i,1},..,z_{i,n_t}) \ \mid\ \vec{z}_i\cdot\vec{v}_i=\widetilde{\delta}\zeta_i\}.\end{array}\right\} \quad (1)$$

$$\boldsymbol{s}_i^* := (\ \overbrace{z_{i,1},..,z_{i,n_t}}^{n_t},\ \overbrace{0^{n_t+1}}^{n_t+1},\ \overbrace{\sigma_{i,1},..,\sigma_{i,n_t}}^{n_t},\ \overbrace{0^2}^{2}\ )_{\mathbb{B}_t^*} \text{ where } \sigma_{i,\iota} \xleftarrow{\mathsf{U}} \mathbb{F}_q \text{ for } \iota=1,..,n_t,$$

$$\boldsymbol{s}_{\ell+1}^* := (\widetilde{\delta}(1,\mathsf{H}_{\mathsf{hk}}^{\lambda,\mathsf{D}}(m\,||\,\mathbb{S})),0,0,\sigma_{\ell+1,1},\sigma_{\ell+1,2},0)_{\mathbb{B}_{d+1}^*} \text{ where } \sigma_{\ell+1,1},\sigma_{\ell+1,2} \xleftarrow{\mathsf{U}} \mathbb{F}_q,$$

$$\text{return } \ \vec{\boldsymbol{s}}^* := (\boldsymbol{s}_0^*,\ldots,\boldsymbol{s}_{\ell+1}^*).$$

Remark: Even when there exist no attributes Γ that satisfy an access structure $\mathbb{S}$, AltSig taking $\mathbb{S}$ as input can generate a correctly verifiable signature $\vec{\boldsymbol{s}}^*$. The signature $\vec{\boldsymbol{s}}^*$ with no matching Γ cannot be generated in the real world and in the unforgeability definition (Definition 9), hence, is considered as only a virtual one.

We now start the proof. This theorem is true if the following statement is true, where AltSig is defined above:

For all $(\mathsf{sk},\mathsf{pk}) \xleftarrow{\mathsf{R}} \mathsf{Setup}(1^\lambda,\ \vec{n})$, all messages m, all attribute sets Γ, all signing keys $\mathsf{sk}_\Gamma \xleftarrow{\mathsf{R}} \mathsf{KeyGen}(\mathsf{pk},\mathsf{sk},\Gamma)$, all access structures $\mathbb{S}$ such that $\mathbb{S}$ accepts Γ, the distributions of $\mathsf{Sig}(\mathsf{pk},\mathsf{sk}_\Gamma,m,\mathbb{S})$ and $\mathsf{AltSig}(\mathsf{pk},\mathsf{sk},m,\mathbb{S})$ are equal.

In the proposed ABS scheme, $(\boldsymbol{s}_0^*,\ldots,\boldsymbol{s}_{\ell+1}^*) \xleftarrow{\mathsf{R}} \mathsf{Sig}(\mathsf{pk},\mathsf{sk}_\Gamma,m,\mathbb{S})$ are expressed by

$$\boldsymbol{s}_0^* := (\vec{z}_0,0,\sigma_0,0)_{\mathbb{B}_0^*},\quad \boldsymbol{s}_{\ell+1}^* := (\vec{z}_{\ell+1},0,0,\sigma_{\ell+1,1},\sigma_{\ell+1,2},0)_{\mathbb{B}_{d+1}^*}$$

$$\boldsymbol{s}_i^* := (z_{i,1},\ldots,z_{i,n_t},0^{n_t+1},\sigma_{i,1},\ldots,\sigma_{i,n_t},0^2)_{\mathbb{B}_t^*} \quad (i=1,\ldots,\ell),$$

where $\vec{z}_i := (z_{i,1},\ldots,z_{i,n_t})$ and $\vec{z}_0 := (\xi\delta),\ \ \vec{z}_{\ell+1} := \xi\delta(1,\mathsf{H}_{\mathsf{hk}}^{\lambda,\mathsf{D}}(m\,||\,\mathbb{S}))$,

for $1\le i\le\ell$,

if $i\in I\ \wedge\ \rho(i)=(t,\vec{v}_i),\ \ \vec{z}_i=\alpha_i\xi\delta\vec{x}_t+\vec{y}_i$

where $\vec{y}_i \xleftarrow{\mathsf{U}} \{\vec{y}_i \ \mid\ \vec{y}_i\cdot\vec{v}_i=0\ \wedge\ y_{i,1}=\beta_i\}$,

if $i\in I\ \wedge\ \rho(i)=\neg(t,\vec{v}_i),\ \ \vec{z}_i=(\alpha_i/(\vec{v}_i\cdot\vec{x}_t))\xi\delta\vec{x}_t+\vec{y}_i$

where $\vec{y}_i \xleftarrow{\mathsf{U}} \{\vec{y}_i \ \mid\ \vec{y}_i\cdot\vec{v}_i=\beta_i\}$,

if $i\notin I\ \wedge\ \rho(i)=(t,\vec{v}_i),\ \ \vec{z}_i=\vec{y}_i$ where $\vec{y}_i \xleftarrow{\mathsf{U}} \{\vec{y}_i \ \mid\ \vec{y}_i\cdot\vec{v}_i=0\ \wedge\ y_{i,1}=\beta_i\}$,

if $i\notin I\ \wedge\ \rho(i)=\neg(t,\vec{v}_i),\ \ \vec{z}_i=\vec{y}_i$ where $\vec{y}_i \xleftarrow{\mathsf{U}} \{\vec{y}_i \ \mid\ \vec{y}_i\cdot\vec{v}_i=\beta_i\}$.

Let $\vec{\alpha}' := (\alpha'_1,\ldots,\alpha'_{\ell+1})$ such that $\alpha'_i := \alpha_i$ if $i\in I$ and $\alpha'_i := 0$ if $i\notin I$, then it can be rephrased by

$$\vec{z}_0 := (\xi\delta),\ \ \vec{z}_{\ell+1} := \xi\delta(1,\mathsf{H}_{\mathsf{hk}}^{\lambda,\mathsf{D}}(m\,||\,\mathbb{S})),$$

for $1\le i\le\ell$,

$\vec{z}_i \xleftarrow{\mathsf{U}} \{\vec{z}_i \ \mid\ \vec{z}_i\cdot\vec{v}_i=0\ \wedge\ z_{i,1}=\xi\delta\alpha'_i+\beta_i\}$ if $\rho(i)=(t,\vec{v}_i)$,

$\vec{z}_i \xleftarrow{\mathsf{U}} \{\vec{z}_i \ \mid\ \vec{z}_i\cdot\vec{v}_i=\xi\delta\alpha'_i+\beta_i\}$ if $\rho(i)=\neg(t,\vec{v}_i)$,

On the other hand, $(\boldsymbol{s}_0^*,\ldots,\boldsymbol{s}_{\ell+1}^*) \xleftarrow{\mathsf{R}} \mathsf{AltSig}(\mathsf{pk},\mathsf{sk},m,\mathbb{S})$ are expressed by

$$\boldsymbol{s}_i^* := (z_{i,1},\ldots,z_{i,n_t},0^{n_t},\sigma_{i,1},\ldots,\sigma_{i,n_t},0)_{\mathbb{B}_t^*} \quad (i=0,\ldots,\ell+1), \quad \text{where}$$

$\vec{z}_0 := (\widetilde{\delta}),\ \ \vec{z}_{\ell+1} := \widetilde{\delta}(1,\mathsf{H}_{\mathsf{hk}}^{\lambda,\mathsf{D}}(m\,||\,\mathbb{S}))$,

for $1\le i\le\ell$,

$\vec{z}_i \xleftarrow{\mathsf{U}} \{\vec{z}_i \ \mid\ \vec{z}_i\cdot\vec{v}_i=0\ \wedge\ z_{i,1}=\widetilde{\delta}\zeta_i\}$ if $\rho(i)=(t,\vec{v}_i)$,

$\vec{z}_i \xleftarrow{\mathsf{U}} \{\vec{z}_i \ \mid\ \vec{z}_i\cdot\vec{v}_i=\widetilde{\delta}\zeta_i\}$ if $\rho(i)=\neg(t,\vec{v}_i)$,

For any $\{\alpha'_i\}$ such that $\sum_{i=1}^{\ell} \alpha'_i M_i = \vec{1}$, the distributions of

$$(\xi\delta, \xi\delta\alpha'_1 + \beta_1, \ldots, \xi\delta\alpha'_\ell + \beta_\ell) \quad \text{s.t.} \quad \xi, \delta \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times, \quad (\beta_i) \xleftarrow{\mathsf{U}} \{(\beta_i) \mid \textstyle\sum_{i=1}^{\ell} \beta_i M_i = \vec{0}\} \text{ and}$$
$$(\widetilde{\delta}, \widetilde{\delta}\zeta_1, \ldots, \widetilde{\delta}\zeta_\ell) \quad \text{s.t.} \quad \widetilde{\delta} \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times, \quad (\zeta_i) \xleftarrow{\mathsf{U}} \{(\zeta_i) \mid \textstyle\sum_{i=1}^{\ell} \zeta_i M_i = \vec{1}\}$$

are equivalent. Therefore, distributions $\mathsf{Sig}(\mathsf{pk}, \mathsf{sk}_\Gamma, m, \mathbb{S})$ and $\mathsf{AltSig}(\mathsf{pk}, \mathsf{sk}, m, \mathbb{S})$ are equivalent. □

E Proof of Theorem 2

Theorem 2 (for General Form ABS) *The proposed ABS scheme is unforgeable (adaptive-predicate unforgeable) under the DLIN assumption and the existence of collision resistance (CR) hash functions.*

For any adversary $\mathcal{A}$, there exist probabilistic machines $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_{3\text{-}1}, \ldots, \mathcal{E}_{3\text{-}4}, \mathcal{E}_5, \mathcal{E}_6$, whose running times are essentially the same as that of $\mathcal{A}$, such that for any security parameter λ,

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{ABS,UF}}(\lambda) \le{}& \textstyle\sum_{i=1}^{n_{\max}} \left(\mathsf{Adv}_{\mathcal{E}_{1\text{-}i}}^{\mathsf{DLIN}}(\lambda) + \mathsf{Adv}_{\mathcal{E}_{2\text{-}i}}^{\mathsf{DLIN}}(\lambda)\right) \\ &+ \textstyle\sum_{h=1}^{\nu_1} \left(\mathsf{Adv}_{\mathcal{E}_{3\text{-}h\text{-}1}}^{\mathsf{DLIN}}(\lambda) + \sum_{i=1}^{n_{\max}} \left(\mathsf{Adv}_{\mathcal{E}_{3\text{-}h\text{-}2\text{-}i}}^{\mathsf{DLIN}}(\lambda) + \mathsf{Adv}_{\mathcal{E}_{3\text{-}h\text{-}3\text{-}i}}^{\mathsf{DLIN}}(\lambda)\right) + \mathsf{Adv}_{\mathcal{E}_{3\text{-}h\text{-}4}}^{\mathsf{DLIN}}(\lambda)\right) \\ &+ \textstyle\sum_{h=1}^{\nu_2} \left(\mathsf{Adv}_{\mathcal{E}_{5\text{-}h}}^{\mathsf{DLIN}}(\lambda) + \mathsf{Adv}_{\mathcal{E}_{6\text{-}h}}^{\mathsf{H,CR}}(\lambda)\right) + \epsilon,\end{aligned}$$

where $\mathcal{E}_{\iota\text{-}i}(\cdot) := \mathcal{E}_\iota(i, \cdot)$ for $\iota = 1, 2$ ($i = 1, \ldots, n_{\max}$), $\mathcal{E}_{\iota\text{-}h}(\cdot) := \mathcal{E}_\iota(h, \cdot)$ for $\iota = 5, 6$ ($h = 1, \ldots, \nu_2$), $\mathcal{E}_{3\text{-}h\text{-}\iota}(\cdot) := \mathcal{E}_{3\text{-}\iota}(h, \cdot)$ for $\iota = 1, 4$, $\mathcal{E}_{3\text{-}h\text{-}\iota\text{-}i}(\cdot) := \mathcal{E}_{3\text{-}\iota}(h, i, \cdot)$ for $\iota = 2, 3$ ($h = 1, \ldots, \nu_1; i = 1, \ldots, n_{\max}$), $n_{\max}$ is the maximum of dimensions n_t ($t = 1, \ldots, d$), ν_1 is the maximum number of $\mathcal{A}$'s reveal key queries, ν_2 is the maximum number of $\mathcal{A}$'s reveal signature queries, and $\epsilon := ((2d + 16)\nu_1 + 8\nu_2 + 2d + 11)/q$.

E.1 Key Techniques

In the security proof of ABS, we use several key ingredients for removing the limitation for multi-use: special matrix transformation, unbounded randomness injection, and one-dimensional localization of inner-product values.

Let $\vec{v}'_i := (s_i\vec{e}_{t,1} + \theta_i\vec{v}_i,\ 0) \in \mathbb{F}_q^{n_t+1}$ if $\rho(i) = (t, \vec{v}_i)$, $\vec{v}'_i := (s_i\vec{v}_i,\ 0) \in \mathbb{F}_q^{n_t+1}$ if $\rho(i) = \neg(t, \vec{v}_i)$, and $\vec{x}'_t := (\vec{x}_t,\ 0) \in \mathbb{F}_q^{n_t+1}$ where s_i, θ_i are defined in algorithm Ver and $\vec{e}_{t,1} := (1, 0, \ldots, 1) \in \mathbb{F}_q^{n_t}$.

Security of previous functional cryptosystems based on DPVS are proven based on *pairwise independence lemma* (Lemma 3 in [28]), i.e., with random matrix pair $Z_t \xleftarrow{\mathsf{U}} GL(n_t{+}1, \mathbb{F}_q)$, $U_t := (Z_t^{-1})^{\mathrm{T}}$ for randomizing vector pair $(\vec{x}'_t, \vec{v}'_i)$ except for the inner product value $\vec{x}'_t \cdot \vec{v}'_i$. However, the method is not applicable to the general, i.e., *multi-use*, situation when multiple vectors $\{\vec{v}'_i\}_{i=1,2,..}$ are considered for the same vector $\vec{x}'_t$.

Namely, for $\vec{x}' := \vec{x}'_1, \vec{v}'_1$ and $(Z, U) := (Z_1, U_1)$ as above, then $(\vec{x}' \cdot Z, \vec{v}'_1 \cdot U)$ is distributed as a pair of *random* vectors with inner-product equals $\vec{x}' \cdot \vec{v}'_1$. However, if we also consider $\vec{v}'_2 \cdot U$ for $\vec{v}'_2 = 2\vec{v}'_1$, the distribution $(\vec{x}' \cdot Z,\ \vec{v}'_1 \cdot U,\ \vec{v}'_2 \cdot U)$ is *not* equivalent to the uniform one on $T := \{(\vec{r}, \vec{w}_1, \vec{w}_2) \,|\, \vec{r} \cdot \vec{w}_1 = p,\ \vec{r} \cdot \vec{w}_2 = 2p,\ \text{with } p := \vec{x}' \cdot \vec{v}'_1\}$ since $\vec{w}_1$ and $\vec{w}_2$ are not parallel for $(\vec{r}, \vec{w}_1, \vec{w}_2) \in T$ in general, but $\vec{v}'_1 \cdot U$ and $\vec{v}'_2 \cdot U$ are clearly parallel. So, since information except for the inner product values may be leaked, the previous information-theoretical argument does not hold. For overcoming the problem, we introduce a new computational technique using a *special matrix transformation* depending on $\vec{x}'$.

If we insert a special matrix Z such that $\vec{x}' \cdot Z = (0, \ldots, 0, 1)$, then $\vec{v}'_1 \cdot U = (*, \ldots, *, \vec{x}' \cdot \vec{v}'_1)$, $\vec{v}'_2 \cdot U = (*, \ldots, *, \vec{x}' \cdot \vec{v}'_2)$, i.e., the last coordinate values of $\{\vec{v}'_i \cdot U\}_{i=1,2,..}$ are inner product

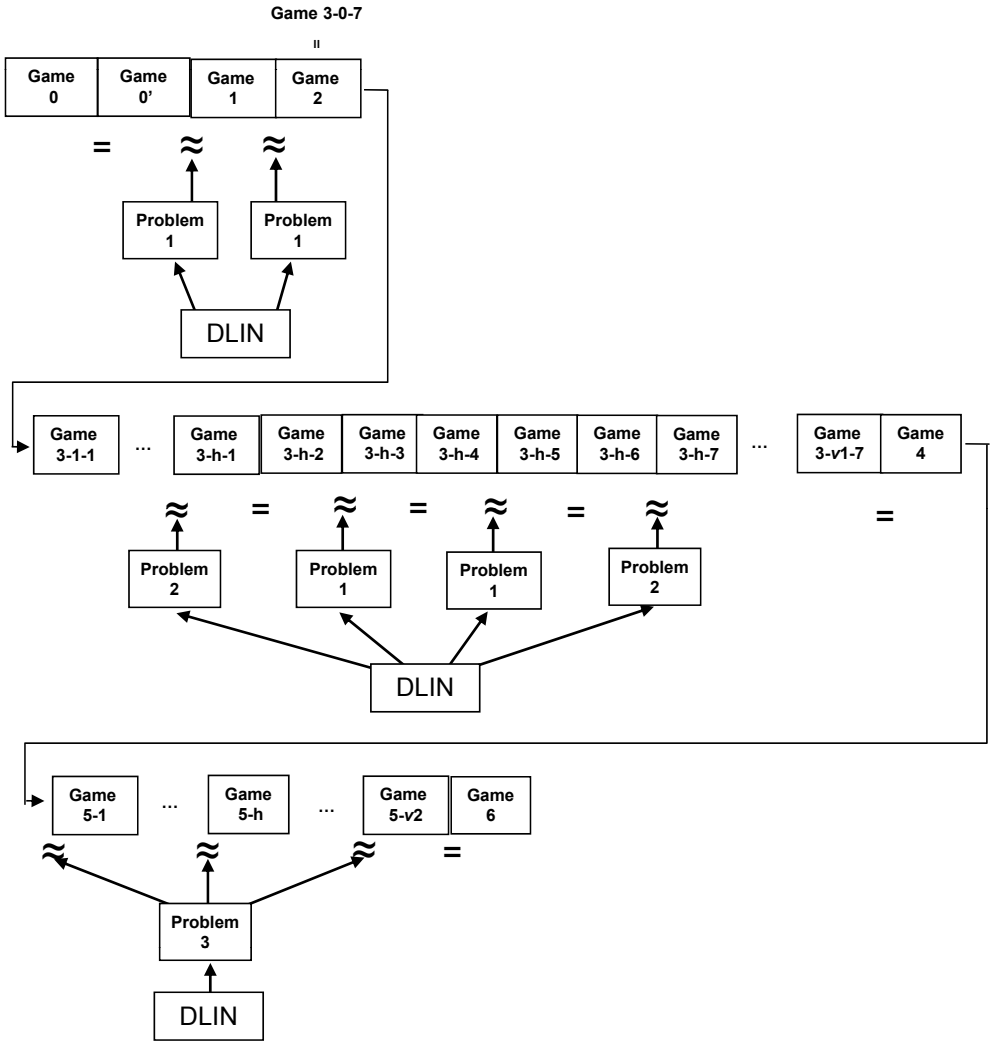

Figure 1: Structure of Reductions

values $\{\vec{x}' \cdot \vec{v}_i'\}_{i=1,2,..}$. While the rest of the first n_t coordinate values of $\{\vec{v}_i' \cdot U\}_{i=1,2,..}$ have additional information, we can *computationally* change these values to uniformly random since the corresponding n_t coordinate values of $\vec{x}' \cdot Z$ are zero, i.e., $\vec{x}' \cdot Z = (0, \ldots, 0, 1)$. The special matrix transformation and the *unbounded* randomness injection technique achieve *one-dimensional localization of inner-product values* in the last one-dimensional subspace without leaking any additional information. While the special matrix Z_t such that $\vec{x}_t' \cdot Z_t = (0, \ldots, 0, 1)$ should be used instead of random Z_t, these matrices are determined when the target key query is issued.

Games 3-h-2 and 3-h-3 ($h = 1, .., \nu_1$) in the proof below reflect the above techniques.

E.2 Proof Outline

As mentioned in Section 4.1, secret signing keys and verification texts in our ABS are the counterparts of secret decryption keys and ciphertexts in CP-FE. Based on this correspondence, we follow the dual system encryption methodology proposed by Waters [36], at the top level of strategy of the unforgeability proof.

Table 2: Forms of verification texts, keys, and signatures

	normal	1-st temp.	2-nd temp.	3-rd temp.	4-th temp.	semi-func.	non-func.
Keys	Eqs. (2), (3), (4)	Eqs. (13), (14), (4)	Eqs. (13), (15), (4)	Eqs. (18), (15), (4)	Eqs. (18), (14), (4)	Eqs. (18), (3), (4)	—
Signatures	Eqs. (5), (6), (7)	—	—	—	—	Eqs. (6), (20)	—
Verification text	Eq. (8)	Eqs. (9), (10), (11)	Eqs. (9), (12), (11)	Eqs. (9), (16), (12), (11)	Eqs. (9), (17), (11)	Eqs. (19), (12), (11)	Eqs. (21), (12), (11)

In the methodology, verification texts (ciphertexts), secret keys and signatures have two forms, *normal* and *semi-functional*. In our proof, we also introduce other forms, *temporary* forms for verification texts and secret keys (Table 2). (Moreover, verification texts have final, *random* form.) The real system uses only normal verification texts, normal secret keys and normal signatures, and semi-functional/temporary forms of verification texts, keys and signatures are used only in a sequence of security games for the unforgeability proof.

To prove this theorem, we employ Game 0 (original unforgeability game) through Game 6. We first Game 0 to Game 0', where queried keys and signatures are calculated and sent to the adversary when corresponding *reveal* queries are issued. In particular, a queried signature is calculated by AltSig in the proof of Theorem 1. When at most ν_1 secret key (KeyGen) reveal queries are issued by an adversary, there are $7\nu_1$ game changes from Game 2 (Game 3-0-7), Game 3-1-1, through Game 3-ν_1-6, Game 3-ν_1-7. When at most ν_2 signature reveal queries are issued by an adversary, there are ν_2 game changes from Game 4 (Game 5-0), Game 5-1 through Game 5-ν_2. The final game, Game 6, is changed from Game 5-ν_2. Since the coefficient of $\boldsymbol{b}^*_{0,1}$ of $\boldsymbol{c}_0$ in the verification text is uniformly randomized in Game 6, the probability that any signature output by an adversary is correctly verified by using the randomized verification text is negligible in Game 6. As usual, we prove that the advantage gaps between neighboring games are negligible.

In Figure 1, an equality between neighboring games indicates that the left-hand game can be conceptually (information-theoretically) changed to the right-hand game. An approximate equality between them indicates that the gap between them is upper-bounded by the advantage of the problem indicated. The DLIN Problem is defined in Definition 3. Problems 1–3 are defined in Definitions 15–17, respectively. We have shown that the intractability of (complicated) Problems 1–3 is reduced to that of the DLIN Problem through several intermediate steps, or intermediate problems, as in [28]. The vertical reductions are also indicated in Figure 1.

A *normal* secret key, $\mathsf{sk}_\Gamma^{\mathsf{norm}}$ (with attribute set Γ), is a correct form of the secret key of the proposed ABS scheme. Similarly, a *normal* verification text is denoted by $\vec{\boldsymbol{c}}_{\mathbb{S}}^{\mathsf{norm}} := (\boldsymbol{c}_0, \ldots, \boldsymbol{c}_{\ell+1})$ (with access structure $\mathbb{S}$), and a *normal* signature is $\vec{\boldsymbol{s}}^{*\,\mathsf{norm}}$. These are given by formulas indicated in Table 2.

A *semi-functional* secret key, $\mathsf{sk}_\Gamma^{\mathsf{semi}}$, a *semi-functional* and *non-functional* verification text, $\vec{\boldsymbol{c}}_{\mathbb{S}}^{\mathsf{semi}}, \vec{\boldsymbol{c}}_{\mathbb{S}}^{\mathsf{non\text{-}f}}$, and temporary forms, $\mathsf{sk}_\Gamma^{\mathsf{temp\text{-}1}}, \ldots, \mathsf{sk}_\Gamma^{\mathsf{temp\text{-}4}}$, $\vec{\boldsymbol{c}}_{\mathbb{S}}^{\mathsf{temp\text{-}1}}, \ldots, \vec{\boldsymbol{c}}_{\mathbb{S}}^{\mathsf{temp\text{-}4}}$, are also given as indicated in Table 2. A *semi-functional* signature is denoted by $\vec{\boldsymbol{s}}^{*\,\mathsf{semi}}$, and is given as in Table 2.

We summarize changes of the forms in Table 3, where shaded parts indicate the verification text or queried key(s), signature(s), which were changed in a game from the previous game.

Table 3: Outline of Game Descriptions

Game	Verification text	Queried keys					Queried sigs		
		1	$\cdots$	h	$\cdots$	ν_1	1	$\cdots$	ν_2
0, 0'	normal	normal							
1	1-st temp.	normal							
2	2-nd temp.	normal							
3-1-1	2-nd temp.	1-st temp.	normal						
3-1-2	3-rd temp.	2-rd temp.	normal						
3-1-3	4-th temp.	2-rd temp.	normal						
3-1-4	4-th temp.	3-rd temp.	normal						
3-1-5	3-rd temp.	3-rd temp.	normal						
3-1-6	2-nd temp.	4-th temp.	normal						
3-1-7	2-nd temp.	semi-func.	normal						
$\vdots$									
3-h-1	2-nd temp.	semi-func.		1-st temp.		normal			
3-h-2	3-rd temp.	semi-func.		2-nd temp.		normal			
3-h-3	4-th temp.	semi-func.		2-nd temp.		normal			
3-h-4	4-th temp.	semi-func.		3-rd temp.		normal			
3-h-5	3-rd temp.	semi-func.		3-rd temp.		normal			
3-h-6	2-nd temp.	semi-func.		4-th temp.		normal			
3-h-7	2-nd temp.	semi-func.		semi-func.		normal			
$\vdots$									
3-ν_1-7	2-nd temp.	semi-func.				semi-func.	normal		
4	semi func.	semi-func.					normal		
5-1	semi-func.	semi-func.					semi-func.		normal
$\vdots$									
5-ν_2	semi-func.	semi-func.							semi func.
6	non-func.	semi func.							

To prove that the advantage gap between Games 0' and 1 is bounded by the advantage of Problem 1 (to guess $\beta \in \{0,1\}$), we construct a simulator of the challenger of Game 0 (or 1) (against an adversary $\mathcal{A}$) by using an instance with $\beta \xleftarrow{\cup} \{0,1\}$ of Problem 1. We then show that the distribution of the secret keys and verification texts replied by the simulator is almost equivalent to those of Game 0 when $\beta = 0$ and Game 1 when $\beta = 1$. That is, the advantage of Problem 1 is almost equivalent to the advantage gap between Games 0 and 1 (Lemma 6). The advantage of Problem 1 is proven to be bounded by that of the DLIN assumption with ignoring a negligible factor (Lemma 1). The advantage gap between Games 1 and 2 is bounded by that of Problem 1 (then DLIN) in a similar manner (Lemma 7).

The advantage gap between Games 3-$(h-1)$-7 and 3-h-1 is similarly shown to be bounded by the advantage of Problem 2 (i.e., of the DLIN assumption) with ignoring a negligible factor (Lemmas 8 and 2). The next two steps are tricky parts in our game transformation. We then show that Game 3-h-1 can be conceptually changed to Game 3-h-2 (Lemma 9), by using the fact

that parts of bases, $(\boldsymbol{b}_{n+1}, \ldots, \boldsymbol{b}_{2n+1})$ and $(\boldsymbol{b}^*_{n+1}, \ldots, \boldsymbol{b}^*_{2n+1})$, are unknown to the adversary. In Game 3-h-2, the semi-functional part of the h-th queried key is encoded as $(0^{n_t},\ \delta')$ with common $\delta' \xleftarrow{\mathsf{U}} \mathbb{F}_q$ for any t. Hence, in the next change, we can mask the first n_t-components in the semi-functional part of the verification text by using Problem 1, and the coefficient vector is given as random $\vec{r}_i \xleftarrow{\mathsf{U}} \mathbb{F}_q^{n_t}$ in the first n_t-dimension and inner-product value $(s'_i\vec{e}_{t,1}+\theta'_i\vec{r}_i)\cdot\vec{x}_t = s'_i+\theta'_i\vec{r}_i\cdot\vec{x}_t$ (or $s'_i\vec{r}_i\cdot\vec{x}_t$) in the last (n_t+1)-th component (Lemma 10). Here, s'_0 in the semi-functional part of verification text $\boldsymbol{c}_0$, is independently distributed from the other variables when $\mathbb{S}$ does not accept Γ (shown in proof of Lemma 11). That is, the joint distribution of h-th queried $\mathsf{sk}_\Gamma^{\mathsf{temp\text{-}2}}$ and $\vec{\boldsymbol{c}}_{\mathbb{S}}^{\mathsf{temp\text{-}4}}$ is equivalent to that of $\mathsf{sk}_\Gamma^{\mathsf{temp\text{-}3}}$ and $\vec{\boldsymbol{c}}_{\mathbb{S}}^{\mathsf{temp\text{-}4}}$ when $\mathbb{S}$ does not accept Γ, which is the distribution in Game 3-h-4.

The advantage gaps between Games 3-h-4 and 3-h-5 is similarly shown to be bounded by the advantage of Problem 1 (i.e., of the DLIN assumption) with ignoring a negligible factor (Lemmas 12 and 1). Then, Game 3-h-5 is conceptually changed to Game 3-h-6 (Lemma 13). And, the gap between Game 3-h-6 and 3-h-7 is bounded by that of Problem 2 as before (Lemma 14). Here, the h-th queried key is in semi-functional form, and game change continues for the next $(h+1)$-th queried key.

After the sequence of games, Game 3-ν_1-7 is conceptually changed to Game 4 (Lemma 15), where the verification text is in semi-functional form.

Then, the advantage gap between Games 5-$(h-1)$ and 5-h is similarly shown to be bounded by the advantage of Problem 3 (i.e., of the DLIN assumption) and the CR hash function with ignoring a negligible factor (Lemmas 16 and 3).

Finally we show that Game 5-ν_2 can be conceptually changed to Game 6 with a negligible error probability (Lemma 17).

E.3 Main Part of the Proof

To prove Theorem 2, we consider the following $(7\nu_1+\nu_2+6)$ games. In Game 0', a part framed by a box indicates coefficients to be changed in a subsequent game. In the other games, a part framed by a box indicates coefficients which were changed in a game from the previous game.

Game 0 : Original game (Definition 9).

Game 0' : Game 0' is the same as Game 0 except the following procedures.

1. When a create key query is issued by $\mathcal{A}$, challenger $\mathcal{C}$ only records the specified attributes, and when a create signature query is issued, $\mathcal{C}$ only records the specified attributes (for key) and access structure. In this step, $\mathcal{C}$ just records, but creates no corresponding keys or signatures.
2. When a reveal key query is issued for attributes Γ which has been already recorded, $\mathcal{C}$ creates the queried key by using $\mathsf{KeyGen}(\mathsf{pk}, \mathsf{sk}, \Gamma)$. And, when a reveal signature query is issued for (attributes Γ and) access structure $\mathbb{S}$ which has been already recorded with Γ satisfying $\mathbb{S}$, $\mathcal{C}$ creates the queried signature by using $\mathsf{AltSig}(\mathsf{pk}, \mathsf{sk}, m, \mathbb{S})$ in the proof of Theorem 1.

That is, the reply to a KeyGen reveal query for $\Gamma := \{(t, \vec{x}_t)\}$ are:

$$\boldsymbol{k}^*_0 := (\ \delta,\ \boxed{0},\ \varphi_0,\ 0\ )_{\mathbb{B}^*_0}, \tag{2}$$

$$\boldsymbol{k}^*_t := (\quad \delta\vec{x}_t,\quad \boxed{0^{n_t+1}},\quad \vec{\varphi}_t,\quad 0^2\ )_{\mathbb{B}^*_t}\ \text{ for } (t,\vec{x}_t)\in\Gamma, \tag{3}$$

$$\left.\begin{aligned} \boldsymbol{k}^*_{d+1,1} &:= (\ \delta(1,0),\ 0^2,\ \vec{\varphi}_{d+1,1},\ 0\ )_{\mathbb{B}^*_{d+1}},\\ \boldsymbol{k}^*_{d+1,2} &:= (\ \delta(0,1),\ 0^2,\ \vec{\varphi}_{d+1,2},\ 0\ )_{\mathbb{B}^*_{d+1}}, \end{aligned}\right\} \tag{4}$$

where $\delta \xleftarrow{\mathsf{U}} \mathbb{F}_q^{\times}$, $\varphi_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $\vec{\varphi}_t \xleftarrow{\mathsf{U}} \mathbb{F}_q^{n_t}$ for $(t, \vec{x}_t) \in \Gamma$, $\vec{\varphi}_{d+1,1}, \vec{\varphi}_{d+1,2} \xleftarrow{\mathsf{U}} \mathbb{F}_q^2$. The reply to an signature (Sig) reveal query for $(m, \mathbb{S})$ with $\mathbb{S} := (M, \rho)$ are:

$$\boldsymbol{s}_0^* := (\ \widetilde{\delta},\ \boxed{0},\ \sigma_0,\ 0\ )_{\mathbb{B}_0^*}, \tag{5}$$

$$\boldsymbol{s}_i^* := (\quad \vec{z}_i,\quad 0^{n_t+1},\quad \vec{\sigma}_i,\quad 0^2\ )_{\mathbb{B}_t^*}\ \text{ for } i = 1, \ldots, \ell, \tag{6}$$

$$\boldsymbol{s}_{\ell+1}^* := (\ \widetilde{\delta}(1, \mathsf{H}_{\mathsf{hk}}^{\lambda,\mathsf{D}}(m \,||\, \mathbb{S})),\ \boxed{0^2},\ \vec{\sigma}_{\ell+1},\ 0\ )_{\mathbb{B}_{d+1}^*}, \tag{7}$$

where, $\widetilde{\delta} \xleftarrow{\mathsf{U}} \mathbb{F}_q^{\times}$, $\sigma_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $\vec{\sigma}_i \xleftarrow{\mathsf{U}} \mathbb{F}_q^{n_t}$, $\vec{\sigma}_{d+1} \xleftarrow{\mathsf{U}} \mathbb{F}_q^2$, $(\zeta_i) \xleftarrow{\mathsf{U}} \{(\zeta_i) \mid \sum_{i=1}^{\ell} \zeta_i M_i = \vec{1}\}$, and for $i = 1, \ldots, \ell$, if $\rho(i) = (t, \vec{v}_i)$, then $\vec{z}_i \xleftarrow{\mathsf{U}} \{\vec{z}_i \mid \vec{z}_i \cdot \vec{v}_i = 0,\ z_{i,1} = \widetilde{\delta}\zeta_i\}$, if $\rho(i) = \neg(t, \vec{v}_i)$, then $\vec{z}_i \xleftarrow{\mathsf{U}} \{\vec{z}_i \mid \vec{z}_i \cdot \vec{v}_i = \widetilde{\delta}\zeta_i\}$.

The components $\boldsymbol{c}_0, \ldots, \boldsymbol{c}_{\ell+1}$ (verification text) for $(m', \mathbb{S}')$ with $\mathbb{S}' := (M, \rho)$ generated in Ver for verifying the output of the adversary are:

$$\left.\begin{array}{l} \boldsymbol{c}_0 := (\ -s_0 - s_{\ell+1},\ \boxed{0},\ 0,\ \eta_0\ )_{\mathbb{B}_0}, \\ \text{for } 1 \le i \le \ell, \\ \quad \text{if } \rho(i) = (t, \vec{v}_i), \quad \boldsymbol{c}_i := (\quad s_i\vec{e}_{t,1} + \theta_i\vec{v}_i,\quad \boxed{0^{n_t+1}},\quad 0^{n_t},\quad \vec{\eta}_i\ )_{\mathbb{B}_t}, \\ \quad \text{if } \rho(i) = \neg(t, \vec{v}_i), \quad \boldsymbol{c}_i := (\quad s_i\vec{v}_i,\quad \boxed{0^{n_t+1}},\quad 0^{n_t},\quad \vec{\eta}_i\ )_{\mathbb{B}_t}, \\ \boldsymbol{c}_{\ell+1} := (\ s_{\ell+1}\vec{e}_1 + \theta_{\ell+1}(-\mathsf{H}_{\mathsf{hk}}^{\lambda,\mathsf{D}}(m' \,||\, \mathbb{S}'),\ 1),\ \boxed{0^2},\ 0^2,\ \eta_{\ell+1}\ )_{\mathbb{B}_{d+1}}, \end{array}\right\} \tag{8}$$

where $\vec{f} \xleftarrow{\mathsf{R}} {\mathbb{F}_q}^r$, $\vec{s}^{\mathrm{T}} := (s_1, \ldots, s_\ell)^{\mathrm{T}} := M \cdot \vec{f}^{\mathrm{T}}$, $s_0 := \vec{1} \cdot \vec{f}^{\mathrm{T}}$, $\vec{e}_{t,1} := (1, 0. \ldots, 0) \in \mathbb{F}_q^{n_t}, \vec{\eta}_i \xleftarrow{\mathsf{U}} \mathbb{F}_q^2, \eta_0, \eta_{\ell+1}, \theta_i, s_{\ell+1} \xleftarrow{\mathsf{U}} \mathbb{F}_q$ $(i = 1, \ldots, \ell+1)$.

Game 1 : Same as Game 0' except that the verification text $(\boldsymbol{c}_0, \ldots, \boldsymbol{c}_{\ell+1})$ for $(m', \mathbb{S}')$ with $\mathbb{S}' := (M, \rho)$ generated in Ver for verifying the output of the adversary are:

$$\boldsymbol{c}_0 := (\ -s_0 - s_{\ell+1},\ \boxed{w_0},\ 0,\ \eta_0\ )_{\mathbb{B}_0}, \tag{9}$$

$$\left.\begin{array}{l} \text{for } 1 \le i \le \ell, \\ \quad \text{if } \rho(i) = (t, \vec{v}_i),\ \boldsymbol{c}_i := (\quad s_i\vec{e}_{t,1} + \theta_i\vec{v}_i,\quad \boxed{\vec{w}_i},\ 0,\quad 0^{n_t},\quad \vec{\eta}_i\ )_{\mathbb{B}_t}, \\ \quad \text{if } \rho(i) = \neg(t, \vec{v}_i),\ \boldsymbol{c}_i := (\quad s_i\vec{v}_i,\quad \boxed{\vec{w}_i},\ 0,\quad 0^{n_t},\quad \vec{\eta}_i\ )_{\mathbb{B}_t}, \end{array}\right\} \tag{10}$$

$$\boldsymbol{c}_{\ell+1} := (\ s_{\ell+1}\vec{e}_1 + \theta_{\ell+1}(-\mathsf{H}_{\mathsf{hk}}^{\lambda,\mathsf{D}}(m' \,||\, \mathbb{S}'),\ 1),\ \boxed{\vec{w}_{\ell+1}},\ 0^2,\ \eta_{\ell+1}\ )_{\mathbb{B}_{d+1}}, \tag{11}$$

where $w_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $\vec{w}_i \xleftarrow{\mathsf{U}} \mathbb{F}_q^{n_t}$ $(i = 1, \ldots, \ell)$, $\vec{w}_{\ell+1} \xleftarrow{\mathsf{U}} \mathbb{F}_q^2$, and all the other variables are generated as in Game 0'.

Game 2 : Same as Game 1 except that the verification text $(\boldsymbol{c}_0, \ldots, \boldsymbol{c}_{\ell+1})$ for $(m', \mathbb{S}')$ with $\mathbb{S}' := (M, \rho)$ generated in Ver for verifying the output of the adversary are:

$$\left.\begin{array}{l} \boldsymbol{c}_0 := (\ -s_0 - s_{\ell+1},\ \boxed{-s_0'},\ 0,\ \eta_0\ )_{\mathbb{B}_0}, \\ \text{for } 1 \le i \le \ell, \\ \quad \text{if } \rho(i) = (t, \vec{v}_i),\ \boldsymbol{c}_i := (\quad s_i\vec{e}_{t,1} + \theta_i\vec{v}_i,\quad \boxed{s_i'\vec{e}_{t,1} + \theta_i'\vec{v}_i},\ 0,\quad 0^{n_t},\quad \vec{\eta}_i\ )_{\mathbb{B}_t}, \\ \quad \text{if } \rho(i) = \neg(t, \vec{v}_i),\ \boldsymbol{c}_i := (\quad s_i\vec{v}_i,\quad \boxed{s_i'\vec{v}_i},\ 0,\quad 0^{n_t},\quad \vec{\eta}_i\ )_{\mathbb{B}_t}, \end{array}\right\} \tag{12}$$

where $\vec{f}' \xleftarrow{\mathsf{U}} {\mathbb{F}_q}^r$, $(\vec{s}')^{\mathrm{T}} := (s_1', \ldots, s_\ell')^{\mathrm{T}} := M \cdot (\vec{f}')^{\mathrm{T}}$, $s_0' := \vec{1} \cdot (\vec{f}')^{\mathrm{T}}$, $\theta_i' \xleftarrow{\mathsf{U}} \mathbb{F}_q$ $(i = 1, \ldots, \ell)$, and all the other variables are generated as in Game 1.

Game 3-h-1 ($h = 1, \ldots, \nu_1$) : Game 3-0-1 is Game 2. Game 3-h-1 is the same as Game 3-$(h-1)$-7 except that $\boldsymbol{k}_t^*$ for $t = 0$ and $(t, \vec{x}_t) \in \Gamma$ of the reply to the h-th KeyGen query is:

$$\boldsymbol{k}_0^* := (\ \delta,\ \boxed{\delta'},\ \varphi_0,\ 0\ )_{\mathbb{B}_0^*}, \tag{13}$$

$$\boldsymbol{k}_t^* := (\quad \delta\vec{x}_t,\quad \boxed{\delta'\vec{x}_t},\ 0,\quad \vec{\varphi}_t,\quad 0^2\ )_{\mathbb{B}_t^*}\ \text{ for } (t, \vec{x}_t) \in \Gamma, \tag{14}$$

where $\delta' \xleftarrow{\cup} \mathbb{F}_q$, and all the other variables are generated as in Game 3-$(h-1)$-7.

Game 3-h-2 ($h = 1, \ldots, \nu_1$) : Game 3-h-2 is the same as Game 3-h-1 except that $\boldsymbol{k}_t^*$ for $(t, \vec{x}_t) \in \Gamma$ of the reply to the h-th KeyGen query and $(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_\ell)$ of the verification text for $(m', \mathbb{S}')$ with $\mathbb{S}' := (M, \rho)$ generated in Ver for verifying the output of the adversary are:

$$\boldsymbol{k}_t^* := (\quad \delta\vec{x}_t,\quad \boxed{0^{n_t},\ \delta'},\quad \vec{\varphi}_t,\quad 0^2\ )_{\mathbb{B}_t^*}\ \text{ for } (t, \vec{x}_t) \in \Gamma, \tag{15}$$

$$\left.\begin{array}{l} \text{for } 1 \le i \le \ell, \\ \quad \text{if } \rho(i) = (t, \vec{v}_i)\ \wedge\ (t, \vec{x}_t) \in \Gamma, \\ \qquad \boldsymbol{c}_i := (\ s_i\vec{e}_{t,1} + \theta_i\vec{v}_i,\ \boxed{(s_i'\vec{e}_{t,1} + \theta_i'\vec{v}_i,\ 0) \cdot Z_t},\ 0^{n_t},\ \vec{\eta}_i\ )_{\mathbb{B}_t}, \\ \quad \text{if } \rho(i) = \neg(t, \vec{v}_i)\ \wedge\ (t, \vec{x}_t) \in \Gamma,\ \boldsymbol{c}_i := (\ s_i\vec{v}_i,\ \boxed{(s_i'\vec{v}_i,\ 0) \cdot Z_t},\ 0^{n_t},\ \vec{\eta}_i\ )_{\mathbb{B}_t}, \end{array}\right\} \tag{16}$$

where $\vec{e}_{t,1} := (1, 0, \ldots, 0) \in \mathbb{F}_q^{n_t}$, $Z_t \xleftarrow{\cup} \{Z_t \in GL(n_t + 1, \mathbb{F}_q) \,|\, (0^{n_t},\ 1) = (\vec{x}_t,\ 0) \cdot (Z_t^{-1})^{\mathrm{T}}\}$, and all the other variables are generated as in Game 3-h-1. We note that the last $((n_t+1)$-th) coordinate of $(s_i'\vec{e}_{t,1} + \theta_i'\vec{v}_i,\ 0) \cdot Z_t \in \mathbb{F}_q^{n_t+1}$ (resp. $(s_i'\vec{v}_i,\ 0) \cdot Z_t \in \mathbb{F}_q^{n_t+1}$) is inner-product value $(s_i'\vec{e}_{t,1} + \theta_i'\vec{v}_i) \cdot \vec{x}_t = s_i' + \theta_i'\vec{v}_i \cdot \vec{x}_t$ (resp. $s_i'\vec{v}_i \cdot \vec{x}_t$).

Game 3-h-3 ($h = 1, \ldots, \nu_1$) : Game 3-h-3 is the same as Game 3-h-2 except that $(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_\ell)$ of the verification text for $(m', \mathbb{S}')$ with $\mathbb{S}' := (M, \rho)$ generated in Ver for verifying the output of the adversary are:

$$\left.\begin{array}{l} \text{for } 1 \le i \le \ell, \\ \quad \text{if } \rho(i) = (t, \vec{v}_i)\ \wedge\ (t, \vec{x}_t) \in \Gamma, \\ \qquad \boldsymbol{c}_i := (\ s_i\vec{e}_{t,1} + \theta_i\vec{v}_i,\ \boxed{\vec{w}_i},\ s_i' + \theta_i'\vec{v}_i \cdot \vec{x}_t,\ 0^{n_t},\ \vec{\eta}_i\ )_{\mathbb{B}_t}, \\ \quad \text{if } \rho(i) = \neg(t, \vec{v}_i)\ \wedge\ (t, \vec{x}_t) \in \Gamma,\ \boldsymbol{c}_i := (\ s_i\vec{v}_i,\ \boxed{\vec{w}_i},\ s_i'\vec{v}_i \cdot \vec{x}_t,\ 0^{n_t},\ \vec{\eta}_i\ )_{\mathbb{B}_t}, \\ \quad \text{if } \rho(i) = (t, \vec{v}_i)\ \wedge\ (t, \vec{x}_t) \notin \Gamma,\ \boldsymbol{c}_i := (\ s_i\vec{e}_{t,1} + \theta_i\vec{v}_i,\ \boxed{\vec{w}_i},\ 0,\ 0^{n_t},\ \vec{\eta}_i\ )_{\mathbb{B}_t}, \\ \quad \text{if } \rho(i) = \neg(t, \vec{v}_i)\ \wedge\ (t, \vec{x}_t) \notin \Gamma,\ \boldsymbol{c}_i := (\ s_i\vec{v}_i,\ \boxed{\vec{w}_i},\ 0,\ 0^{n_t},\ \vec{\eta}_i\ )_{\mathbb{B}_t}, \end{array}\right\} \tag{17}$$

where, $\vec{w}_i \xleftarrow{\cup} \mathbb{F}_q^{n_t}$ for $i = 1, \ldots, \ell$, and all the other variables are generated as in Game 3-h-2.

Game 3-h-4 ($h = 1, \ldots, \nu_1$) : Game 3-h-4 is the same as Game 3-h-3 except that $\boldsymbol{k}_0^*$ of the reply to the h-th KeyGen query is:

$$\boldsymbol{k}_0^* := (\ \delta,\ \boxed{r_0},\ \varphi_0,\ 0\ )_{\mathbb{B}_0^*}, \tag{18}$$

where $r_0 \xleftarrow{\cup} \mathbb{F}_q$, which is independent from $\delta' \xleftarrow{\cup} \mathbb{F}_q$ in Eq. (15), and all the other variables are generated as in Game 3-h-3.

Game 3-h-5 ($h = 1, \ldots, \nu_1$) : Game 3-h-5 is the same as Game 3-h-4 except that $(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_\ell)$ of the verification text for $(m', \mathbb{S}')$ with $\mathbb{S}' := (M, \rho)$ generated in Ver for verifying the output of the adversary are given as in Eq. (16), where $Z_t \xleftarrow{\mathsf{U}} \{Z_t \in GL(n_t+1, \mathbb{F}_q) \,|\, (0^{n_t},\ 1) = (\vec{x}_t,\ 0) \cdot (Z_t^{-1})^{\mathrm{T}}\}$, and all the other variables are generated as in Game 3-h-4.

Game 3-h-6 ($h = 1, \ldots, \nu_1$) : Game 3-h-6 is the same as Game 3-h-5 except that $\boldsymbol{k}_t^*$ for $(t, \vec{x}_t) \in \Gamma$ of the reply to the h-th KeyGen query are given as in Eq. (14) with $\delta' \xleftarrow{\mathsf{U}} \mathbb{F}_q$ (independent from $r_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q$ in Eq. (18)) and $(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_\ell)$ of the verification text for $(m', \mathbb{S}')$ with $\mathbb{S}' := (M, \rho)$ generated in Ver for verifying the output of the adversary are given as in Eq. (12).

Game 3-h-7 ($h = 1, \ldots, \nu_1$) : Game 3-h-7 is the same as Game 3-h-6 except that $\boldsymbol{k}_t^*$ for $(t, \vec{x}_t) \in \Gamma$ of the reply to the h-th KeyGen query are given as in Eq. (3).

Game 4 : Same as Game 3-ν_1-7 except that the verification text $(\boldsymbol{c}_0, \ldots, \boldsymbol{c}_{\ell+1})$ for $(m', \mathbb{S}')$ with $\mathbb{S}' := (M, \rho)$ generated in Ver for verifying the output of the adversary are:

$$\boldsymbol{c}_0 := (\ -s_0 - s_{\ell+1},\ \boxed{w_0},\ 0,\ \eta_0\ )_{\mathbb{B}_0}, \tag{19}$$

where $w_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q$, and all the other variables are generated as in Game 3-ν_1-7.

Game 5-h ($h = 1, \ldots, \nu_2$) : Game 5-0 is Game 4. Game 5-h is the same as Game 5-$(h-1)$ except that $\boldsymbol{s}_0^*, \boldsymbol{s}_{\ell+1}^*$ of the reply to the h-th signature reveal query for $(m, \mathbb{S})$ are:

$$\left.\begin{aligned} \boldsymbol{s}_0^* &:= (\ \widetilde{\delta},\ \boxed{\widetilde{r}_0},\ \sigma_0,\ 0\ )_{\mathbb{B}_0^*}, \\ \boldsymbol{s}_{\ell+1}^* &:= (\ \widetilde{\delta}(1, \mathsf{H}_{\mathsf{hk}}^{\lambda,\mathsf{D}}(m \,||\, \mathbb{S})),\ \boxed{\vec{\widetilde{r}}_{\ell+1}},\ \vec{\sigma}_{\ell+1},\ 0\ )_{\mathbb{B}_{d+1}^*}, \end{aligned}\right\} \tag{20}$$

where $\widetilde{r}_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $\vec{\widetilde{r}}_{\ell+1} \xleftarrow{\mathsf{U}} {\mathbb{F}_q}^2$, and all the other variables are generated as in Game 5-$(h-1)$.

Game 6 : Same as Game 5-ν_2 except that $\boldsymbol{c}_0$ generated in Ver for verifying the output of the adversary is:

$$\boldsymbol{c}_0 := (\ \boxed{\widetilde{s}_0},\ w_0,\ 0,\ \eta_0\ )_{\mathbb{B}_0}, \tag{21}$$

where $\widetilde{s}_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q$ (i.e., independent from all the other variables).

Let $\mathsf{Adv}_{\mathcal{A}}^{(0)}(\lambda)$ ($= \mathsf{Adv}_{\mathcal{A}}^{(0')}(\lambda)$ by Lemma 5) be $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{ABS,UF}}(\lambda)$ in Game 0, and $\mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda), \mathsf{Adv}_{\mathcal{A}}^{(2)}(\lambda)$, $\mathsf{Adv}_{\mathcal{A}}^{(3\text{-}h\text{-}\iota)}(\lambda), \ldots, \mathsf{Adv}_{\mathcal{A}}^{(6)}(\lambda)$ be the advantage of $\mathcal{A}$ in Game $1, 2, 3\text{-}h\text{-}\iota, \ldots, 6$, respectively. It is obtained that $\mathsf{Adv}_{\mathcal{A}}^{(6)}(\lambda) = 1/q$ by Lemma 18.

We will show twelve lemmas (Lemmas 6–17) that evaluate the gaps between pairs of neighboring games. From these lemmas and Lemmas 1–3, we obtain

$$\begin{aligned} \mathsf{Adv}_{\mathcal{A}}^{\mathsf{ABS,UF}}(\lambda) = \mathsf{Adv}_{\mathcal{A}}^{(0)}(\lambda) &\leq \left|\mathsf{Adv}_{\mathcal{A}}^{(0)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda)\right| + \left|\mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(2)}(\lambda)\right| + \\ &\sum_{h=1}^{\nu_1}\sum_{\iota=1}^{7}\left|\mathsf{Adv}_{\mathcal{A}}^{(3\text{-}h\text{-}(\iota-1))}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(3\text{-}h\text{-}\iota)}(\lambda)\right| + \left|\mathsf{Adv}_{\mathcal{A}}^{(3\text{-}\nu_1\text{-}7)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(4)}(\lambda)\right| + \\ &\sum_{h=1}^{\nu_2}\left|\mathsf{Adv}_{\mathcal{A}}^{(5\text{-}(h-1))}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(5\text{-}h)}(\lambda)\right| + \left|\mathsf{Adv}_{\mathcal{A}}^{(5\text{-}\nu_2)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(6)}(\lambda)\right| + \mathsf{Adv}_{\mathcal{A}}^{(6)}(\lambda) \\ &\leq \sum_{i=1}^{n_{\max}}\left(\mathsf{Adv}_{\mathcal{E}_{1\text{-}i}}^{\mathsf{DLIN}}(\lambda) + \mathsf{Adv}_{\mathcal{E}_{2\text{-}i}}^{\mathsf{DLIN}}(\lambda)\right) \\ &\quad + \sum_{h=1}^{\nu_1}\left(\mathsf{Adv}_{\mathcal{E}_{3\text{-}h\text{-}1}}^{\mathsf{DLIN}}(\lambda) + \sum_{i=1}^{n_{\max}}\left(\mathsf{Adv}_{\mathcal{E}_{3\text{-}h\text{-}2\text{-}i}}^{\mathsf{DLIN}}(\lambda) + \mathsf{Adv}_{\mathcal{E}_{3\text{-}h\text{-}3\text{-}i}}^{\mathsf{DLIN}}(\lambda)\right) + \mathsf{Adv}_{\mathcal{E}_{3\text{-}h\text{-}4}}^{\mathsf{DLIN}}(\lambda)\right) \\ &\quad + \sum_{h=1}^{\nu_2}\left(\mathsf{Adv}_{\mathcal{E}_{5\text{-}h}}^{\mathsf{DLIN}}(\lambda) + \mathsf{Adv}_{\mathcal{E}_{6\text{-}h}}^{\mathsf{H,CR}}(\lambda)\right) + \epsilon, \end{aligned}$$

where Game 3-h-0 is Game 3-$(h-1)$-7. $\epsilon := ((2d+22)\nu_1 + 8\nu_2 + 2d + 15)/q$. This completes the proof of Theorem 2. □

E.4 Problems 1–3 and Their Security

Definition 15 (Problem 1) *Problem 1 is to guess $\beta \in \{0,1\}$, given* $(\mathsf{param}_{\vec{n}}, \{\mathbb{B}_t, \widehat{\mathbb{B}}^*_t\}_{t=0,\ldots,d+1}, \boldsymbol{f}_{\beta,0}, \{\boldsymbol{e}_{\beta,t,i}\}_{t=1,\ldots,d;i=1,\ldots,n_t}, \boldsymbol{f}_{\beta,d+1}, \boldsymbol{f}_{d+1,2}) \xleftarrow{\mathsf{R}} \mathcal{G}^{\mathsf{P1}}_\beta(1^\lambda, \vec{n})$, *where*

$$
\begin{aligned}
&\mathcal{G}^{\mathsf{P1}}_\beta(1^\lambda, \vec{n}) : \quad (\mathsf{param}_{\vec{n}}, \{\mathbb{B}_t, \mathbb{B}^*_t\}_{t=0,\ldots,d+1}) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}(1^\lambda, \vec{n}), \quad \widehat{\mathbb{B}}^*_0 := (\boldsymbol{b}^*_{0,1}, \boldsymbol{b}^*_{0,3}, \boldsymbol{b}^*_{0,4}),\\
&\widehat{\mathbb{B}}^*_t := (\boldsymbol{b}^*_{t,1}, \ldots, \boldsymbol{b}^*_{t,n_t}, \boldsymbol{b}^*_{t,2n_t+1}, \ldots, \boldsymbol{b}^*_{t,3n_t+3}) \quad \text{for } t = 1, \ldots, d,\\
&\widehat{\mathbb{B}}^*_{d+1} := (\boldsymbol{b}^*_{d+1,1}, \boldsymbol{b}^*_{d+1,2}, \boldsymbol{b}^*_{d+1,5}, \ldots, \boldsymbol{b}^*_{d+1,7}),\\
&\omega, z_0, \gamma_0, \gamma_{d+1} \xleftarrow{\mathsf{U}} \mathbb{F}_q, \quad \vec{z}_{d+1} \xleftarrow{\mathsf{U}} \mathbb{F}_q^2, \quad \boldsymbol{f}_{0,0} := (\omega, 0, 0, \gamma_0)_{\mathbb{B}_0}, \quad \boldsymbol{f}_{1,0} := (\omega, z_0, 0, \gamma_0)_{\mathbb{B}_0},\\
&\boldsymbol{f}_{0,d+1} := (\omega, 0, 0^2, 0^2, \gamma_{d+1})_{\mathbb{B}_{d+1}}, \quad \boldsymbol{f}_{1,d+1} := (\omega, 0, \vec{z}_{d+1}, 0^2, \gamma_{d+1})_{\mathbb{B}_{d+1}},\\
&\text{for } t = 1, \ldots, d;\ i = 1, \ldots, n_t; \quad \vec{e}_{t,i} := (0^{i-1},\ 1,\ 0^{n_t-i}) \in \mathbb{F}_q^{n_t}, \quad z_{t,i}, \gamma_{t,i,1}, \gamma_{t,i,2} \xleftarrow{\mathsf{U}} \mathbb{F}_q,\\
&\qquad\qquad \overbrace{\phantom{0^{n_t}}}^{n_t} \quad \overbrace{\phantom{0^{n_t+1}}}^{n_t+1} \quad \overbrace{\phantom{0^{n_t}}}^{n_t} \quad \overbrace{}^{2}\\
&\boldsymbol{e}_{0,t,i} := \ (\ 0^{n_t},\ 0^{n_t+1},\ 0^{n_t},\ \gamma_{t,i,1}, \gamma_{t,i,2}\ )_{\mathbb{B}_t},\\
&\boldsymbol{e}_{1,t,i} := \ (\ 0^{n_t},\ z_{t,i}\vec{e}_{t,i},\ 0,\ 0^{n_t},\ \gamma_{t,i,1}, \gamma_{t,i,2}\ )_{\mathbb{B}_t},\\
&\boldsymbol{f}_{t,i} := \omega \boldsymbol{b}_{t,i},\\
&\text{return} \quad (\mathsf{param}_{\vec{n}}, \{\mathbb{B}_t, \widehat{\mathbb{B}}^*_t\}_{t=0,\ldots,d+1}, \boldsymbol{f}_{\beta,0}, \{\boldsymbol{e}_{\beta,t,i}, \boldsymbol{f}_{t,i}\}_{t=1,\ldots,d;i=1,\ldots,n_t}, \boldsymbol{f}_{\beta,d+1}).
\end{aligned}
$$

for $\beta \xleftarrow{\mathsf{U}} \{0,1\}$. For a probabilistic machine $\mathcal{B}$, we define the advantage of $\mathcal{B}$ as the quantity

$$
\mathsf{Adv}^{\mathsf{P1}}_{\mathcal{B}}(\lambda) := \left| \mathsf{Pr}\left[\mathcal{B}(1^\lambda, \varrho) \to 1 \;\middle|\; \varrho \xleftarrow{\mathsf{R}} \mathcal{G}^{\mathsf{P1}}_0(1^\lambda, \vec{n})\right] - \mathsf{Pr}\left[\mathcal{B}(1^\lambda, \varrho) \to 1 \;\middle|\; \varrho \xleftarrow{\mathsf{R}} \mathcal{G}^{\mathsf{P1}}_1(1^\lambda, \vec{n})\right] \right|.
$$

Lemma 1 *For any adversary $\mathcal{B}$, there are probabilistic machine $\mathcal{E}_i$, whose running time is essentially the same as that of $\mathcal{B}$, such that for any security parameter λ, $\mathsf{Adv}^{\mathsf{P1}}_{\mathcal{B}}(\lambda) \le \sum_{i=1}^{n_{\max}} \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}_i}(\lambda) + 5n_{\max}/q$, where $n_{\max}$ is the maximum of dimensions n_t for $t = 1, \ldots, d$.*

Proof. We first define hybrid experiments $\mathsf{Exp1}$-j $(j = 0, \ldots, n_{\max})$ as the adversary $\mathcal{B}$ is given

an instance $(\mathsf{param}_{\vec{n}}, \{\mathbb{B}_t, \widehat{\mathbb{B}}^*_t\}_{t=0,\ldots,d+1}, \boldsymbol{f}_{0,0}, \{\boldsymbol{e}_{0,t,i}, \boldsymbol{f}_{t,i}\}_{t=1,\ldots,d;i=1,\ldots,n_t}, \boldsymbol{f}_{0,d+1})$ if $j = 0$, or

an instance $(\mathsf{param}_{\vec{n}}, \{\mathbb{B}_t, \widehat{\mathbb{B}}^*_t\}_{t=0,\ldots,d+1}, \boldsymbol{f}_{1,0}, \{\boldsymbol{e}_{1,t,i}\}_{t=1,\ldots,d;i=1,\ldots,j}, \{\boldsymbol{e}_{0,t,i}\}_{t=1,\ldots,d;i=j+1,\ldots,n_t},$
$\{\boldsymbol{f}_{t,i}\}_{t=1,\ldots,d;i=1,\ldots,n_t}, \boldsymbol{f}_{1,d+1})$ if $j > 0$,

where all elements are generated as in the Problem 1 generator $\mathcal{G}^{\mathsf{P1}}_\beta$, and $\mathcal{B}$ outputs a bit β'. Then, an instance in $\mathsf{Exp1}$-0 (resp. $\mathsf{Exp1}$-$n_{\max}$) has the same distribution of output of $\mathcal{G}^{\mathsf{P1}}_0(1^\lambda, \vec{n})$ (resp. $\mathcal{G}^{\mathsf{P1}}_1(1^\lambda, \vec{n})$), i.e., $\mathsf{Adv}^{\mathsf{P1}}_{\mathcal{B}}(\lambda) = \left|\mathsf{Pr}\left[\mathsf{Exp1}^0_{\mathcal{B}}(1^\lambda) \to 1\right] - \mathsf{Pr}\left[\mathsf{Exp1}^{n_{\max}}_{\mathcal{B}}(1^\lambda) \to 1\right]\right|$, where $\mathsf{Exp1}^j_{\mathcal{B}}(1^\lambda)$ is the output of $\mathcal{B}$ in the experiment $\mathsf{Exp1}$-j for $j = 0, \ldots, n_{\max}$. Therefore,

$$
\mathsf{Adv}^{\mathsf{P1}}_{\mathcal{B}}(\lambda) \le \sum_{j=1}^{n_{\max}} \left| \mathsf{Pr}\left[\mathsf{Exp1}^{j-1}_{\mathcal{B}}(1^\lambda) \to 1\right] - \mathsf{Pr}\left[\mathsf{Exp1}^{j}_{\mathcal{B}}(1^\lambda) \to 1\right] \right|.
$$

From Claim 1, we have $\mathsf{Adv}^{\mathsf{P1}}_{\mathcal{B}}(\lambda) \le \sum_{i=1}^{n_{\max}} \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}_i}(\lambda) + 5n_{\max}/q$. □

Claim 1 *There are probabilistic machines $\mathcal{E}_i$, whose running time are essentially the same as that of $\mathcal{B}$, such that for any security parameter λ, $\left|\mathsf{Pr}\left[\mathsf{Exp1}^{j-1}_{\mathcal{B}}(1^\lambda) \to 1\right] - \mathsf{Pr}\left[\mathsf{Exp1}^{j}_{\mathcal{B}}(1^\lambda) \to 1\right]\right| \le \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}_j}(\lambda) + 5/q$ for $j = 1, \ldots, n_{\max}$.*

Proof. First, we show the relation of Claim 1 and Basic Problem 1 in [28]. The problem asks to distinguish two sets of d' vectors $\{\boldsymbol{f}'_{0,t} := \delta \boldsymbol{d}_{t,1} + \sigma \boldsymbol{d}_{t,3}\}_{t=0,\ldots,d'}$ and $\{\boldsymbol{f}'_{1,t} := \delta \boldsymbol{d}_{t,1} + \rho \boldsymbol{d}_{t,2} + \sigma \boldsymbol{d}_{t,3}\}_{t=0,\ldots,d'}$ with three dimensional basis $\{\boldsymbol{d}_{t,1}, \boldsymbol{d}_{t,2}, \boldsymbol{d}_{t,3}\} \subset \mathbb{B}_t$ as well as several auxiliary elements. We denote the problem of distinguishing an instance of $\mathsf{Exp1}^{j-1}$ and that of $\mathsf{Exp1}^{j}$ given in Claim 1 by Prob^j. Prob^j has a similar form as Basic Problem 1 (BP1) for $j = 1, \ldots, n_{\max}$.

Namely, for $j = 1$, Prob^1 is to distinguish

$$(\boldsymbol{f}_{0,0} := \omega \boldsymbol{b}_{0,1} + \gamma_0 \boldsymbol{b}_{0,4}, \{\boldsymbol{e}_{0,t,1} := \gamma_{t,1,1} \boldsymbol{b}_{t,3n_t+2} + \gamma_{t,1,2} \boldsymbol{b}_{t,3n_t+3}\}_{t=1,\ldots,d}, \\ \boldsymbol{f}_{0,d+1} := \omega \boldsymbol{b}_{d+1,1} + \gamma_{d+1} \boldsymbol{b}_{d+1,7})$$

and

$$(\boldsymbol{f}_{1,0} := \omega \boldsymbol{b}_{0,1} + z_0 \boldsymbol{b}_{0,2} + \gamma_0 \boldsymbol{b}_{0,4}, \{\boldsymbol{e}_{1,t,1} := \gamma_{t,1,1} \boldsymbol{b}_{t,3n_t+2} + z_{t,1} \boldsymbol{b}_{t,n_t+1} + \gamma_{t,1,2} \boldsymbol{b}_{t,3n_t+3}\}_{t=1,\ldots,d}, \\ \boldsymbol{f}_{1,d+1} := \omega \boldsymbol{b}_{d+1,1} + z_{d+1,1} \boldsymbol{b}_{d+1,3} + z_{d+1,2} \boldsymbol{b}_{d+1,4} + \gamma_{d+1} \boldsymbol{b}_{d+1,7})$$

with other auxiliary elements. The elements $(\boldsymbol{f}_{\beta,0}, \{\boldsymbol{e}_{\beta,t,1}\}_{t=1,\ldots,d}, \boldsymbol{f}_{\beta,d+1})$ have the similar form as the above three dimensional subspace problem instance of BP1 except that the first coefficient $\gamma_{t,1,1}$ in $\boldsymbol{e}_{0,t,1}$ (and $\boldsymbol{e}_{1,t,1}$) is not ω, which is commonly used in $\boldsymbol{f}_{0,0}$ (and $\boldsymbol{f}_{1,0}$) and $\boldsymbol{f}_{0,d+1}$ (and $\boldsymbol{f}_{1,d+1}$), that $\boldsymbol{f}_{1,d+1}$ has two dimensional semi-functional part $z_{d+1,1} \boldsymbol{b}_{d+1,3} + z_{d+1,2} \boldsymbol{b}_{d+1,4}$, and that all semi-functional coefficients, i.e., $z_0, \{z_{t,1}\}_{t=1,\ldots,d}, z_{d+1,1}, z_{d+1,2}$, are different from each other.

Based on the similarity, we construct a (probabilistic) machine $\mathcal{C}_1$ against BP1 (on the same spaces $\mathsf{span}\langle \mathbb{B}_t \rangle_{t=0,\ldots,d+1}$ of Prob^1 and three dimensional subspaces $\{\boldsymbol{d}_{t,1}, \boldsymbol{d}_{t,2}, \boldsymbol{d}_{t,3}\}$ are defined in the same manner as in Prob^1, that is, $\boldsymbol{d}_{0,1} := \boldsymbol{b}_{0,1}, \boldsymbol{d}_{0,2} := \boldsymbol{b}_{0,2}, \boldsymbol{d}_{0,3} := \boldsymbol{b}_{0,4}$, $\boldsymbol{d}_{t,1} := \boldsymbol{b}_{t,3n_t+2}, \boldsymbol{d}_{t,2} := \boldsymbol{b}_{t,n_t+1}, \boldsymbol{d}_{t,3} := \boldsymbol{b}_{t,3n_t+3}$ for $t = 1, \ldots, d$, and $\boldsymbol{d}_{d+1,1} := \boldsymbol{b}_{d+1,1}, \boldsymbol{d}_{d+1,2} := \boldsymbol{b}_{d+1,3}$, $\boldsymbol{d}_{d+1,3} := \boldsymbol{b}_{d+1,7}$) by using an adversary $\mathcal{B}$ against Prob^1. $\mathcal{C}_1$ takes $\{\boldsymbol{f}'_{\beta,t}\}_{t=0,\ldots,d+1}$ and other auxiliary elements as input of BP1. Then, $\mathcal{C}_1$ calculates

$$\widetilde{\boldsymbol{f}}_{\beta,0} := \boldsymbol{f}'_{\beta,0} + \widetilde{\gamma}_0 \boldsymbol{b}_{0,4}, \{\widetilde{\boldsymbol{e}}_{\beta,t,1} := \boldsymbol{f}'_{\beta,t} + \widetilde{\gamma}_{t,1,1} \boldsymbol{b}_{t,3n_t+2} + \widetilde{\gamma}_{t,1,2} \boldsymbol{b}_{t,3n_t+3}\}_{t=1,\ldots,d}, \\ \widetilde{\boldsymbol{f}}_{\beta,d+1} := \boldsymbol{f}'_{\beta,d+1} + \widetilde{\gamma}_{d+1} \boldsymbol{b}_{d+1,7}$$

where $\widetilde{\gamma}_0, \widetilde{\gamma}_{t,1,1}, \widetilde{\gamma}_{t,1,2}, \widetilde{\gamma}_{d+1} \xleftarrow{\mathsf{U}} \mathbb{F}_q$ and other elements $\{\boldsymbol{e}_{0,t,i}\}_{t=1,\ldots,d;i=2,\ldots,n_t}, \{\boldsymbol{f}_{t,i}\}_{t=1,\ldots,d;i=1,\ldots,n_t}$ are generated using $\{\mathbb{B}_t\}_{t=1,\ldots,d}$, $\boldsymbol{f}_{t,i}$ in BP1, and freshly generated $\{\gamma_{t,i,1}, \gamma_{t,i,2} \xleftarrow{\mathsf{U}} \mathbb{F}_q\}_{t=1,\ldots,d;i=2,\ldots,n_t}$. $\mathcal{C}_1$ also generates a part of semi-functional space basis as: $\widetilde{\boldsymbol{b}}_{0,2} := r_0 \boldsymbol{b}_{0,2}, \widetilde{\boldsymbol{b}}_{t,n_t+1} := r_t \boldsymbol{b}_{t,n_t+1}$ for $t = 1, \ldots, d$, $(\widetilde{\boldsymbol{b}}_{d+1,3}, \widetilde{\boldsymbol{b}}_{d+1,4}) := (\boldsymbol{b}_{d+1,3}, \boldsymbol{b}_{d+1,4}) \cdot R_{d+1}$ where $r_t \xleftarrow{\mathsf{U}} \mathbb{F}_q$ for $t = 0, \ldots, d$ and $R_{d+1} \xleftarrow{\mathsf{U}} GL(2, \mathbb{F}_q)$. $\mathcal{C}_1$ then sets new bases $\widetilde{\mathbb{B}}_0 := (\boldsymbol{b}_{0,1}, \widetilde{\boldsymbol{b}}_{0,2}, \boldsymbol{b}_{0,3}, \boldsymbol{b}_{0,4}), \widetilde{\mathbb{B}}_t := (\boldsymbol{b}_{t,1}, \ldots, \boldsymbol{b}_{t,n_t}, \widetilde{\boldsymbol{b}}_{t,n_t+1}, \boldsymbol{b}_{t,n_t+2}, \ldots, \boldsymbol{b}_{t,3n_t+3})$ for $t = 1, \ldots, d$, and $\widetilde{\mathbb{B}}_{d+1} := (\boldsymbol{b}_{d+1,1}, \boldsymbol{b}_{d+1,2}, \widetilde{\boldsymbol{b}}_{d+1,3}, \widetilde{\boldsymbol{b}}_{d+1,4}, \boldsymbol{b}_{d+1,5}, \ldots, \boldsymbol{b}_{d+1,7})$. Note that new basis $\widetilde{\mathbb{B}}_t$ is compatible with $\widehat{\mathbb{B}}^*_t$ for $t = 0, \ldots, d+1$, since $\widehat{\mathbb{B}}^*_t$ has no basis vectors for the semi-functional part. $\mathcal{C}_1$ gives a Prob^1 instance $\varrho := (\mathsf{param}_{\vec{n}}, \{\widetilde{\mathbb{B}}_t, \widehat{\mathbb{B}}^*_t\}_{t=0,\ldots,d+1}, \widetilde{\boldsymbol{f}}_{\beta,0}, \{\widetilde{\boldsymbol{e}}_{\beta,t,1}, \boldsymbol{e}_{0,t,i}\}_{t=1,\ldots,d;i=2,\ldots,n_t}, \{\boldsymbol{f}_{t,i}\}_{t=1,\ldots,d;i=1,\ldots,n_t}, \widetilde{\boldsymbol{f}}_{\beta,d+1})$ to $\mathcal{B}$. If $\mathcal{B}$ finally outputs β', then $\mathcal{C}_1$ outputs β'. Since the distribution of ϱ is equivalent to that of the output of $\mathsf{Exp1}^0$ (resp. $\mathsf{Exp1}^1$) when $\beta = 0$ (resp. $\beta = 1$), we have $\mathsf{Adv}^{\mathsf{Prob}^1}_{\mathcal{B}}(\lambda) \leq \mathsf{Adv}^{\mathsf{BP1}}_{\mathcal{C}_1}(\lambda)$.

For $j = 2, \ldots, n_{\max}$, Prob^j is to distinguish

$$\boldsymbol{e}_{0,t,j} := \gamma_{t,j,1} \boldsymbol{b}_{t,3n_t+2} + \gamma_{t,j,2} \boldsymbol{b}_{t,3n_t+3} \text{ for } t \in \{1, \ldots, d\} \text{ such that } j \leq n_t,$$

and

$$\boldsymbol{e}_{1,t,j} := \gamma_{t,j,1} \boldsymbol{b}_{t,3n_t+2} + z_{t,j} \boldsymbol{b}_{t,n_t+j} + \gamma_{t,j,2} \boldsymbol{b}_{t,3n_t+3} \text{ for } t \in \{1, \ldots, d\} \text{ such that } j \leq n_t,$$

with other auxiliary elements. The elements $\{\boldsymbol{e}_{\beta,t,j}\}_{t\in\{1,\ldots,d\}\text{ s.t. }j\le n_t}$ have the similar form as the above three dimensional subspace problem instance of $\mathsf{BP1}$ except that the coefficients $\gamma_{t,j,1},\gamma_{t,j,2},z_{t,j}$ in $\boldsymbol{e}_{0,t,j}$ (and $\boldsymbol{e}_{1,t,j}$) are independently and uniformly distributed.

Based on the similarity, we construct a (probabilistic) machine $\mathcal{C}_j$ against $\mathsf{BP1}$ (on the same spaces $\mathsf{span}\langle\mathbb{B}_t\rangle_{t=0,\ldots,d+1}$ of Prob^j and three dimensional subspaces $\{\boldsymbol{d}_{t,1},\boldsymbol{d}_{t,2},\boldsymbol{d}_{t,3}\}$ are defined in the same manner as in Prob^j, that is, $\boldsymbol{d}_{t,1}:=\boldsymbol{b}_{t,3n_t+2},\boldsymbol{d}_{t,2}:=\boldsymbol{b}_{t,n_t+j},\boldsymbol{d}_{t,3}:=\boldsymbol{b}_{t,3n_t+3}$ for $t=1,\ldots,d$) by using an adversary $\mathcal{B}$ against Prob^j. $\mathcal{C}_j$ takes $\{\boldsymbol{f}'_{\beta,t}\}_{t=0,\ldots,d+1}$ and other auxiliary elements as input of $\mathsf{BP1}$. Then, $\mathcal{C}_j$ calculates

$$\widetilde{\boldsymbol{e}}_{\beta,t,j}:=\boldsymbol{f}'_{\beta,t}+\widetilde{\gamma}_{t,j,1}\boldsymbol{b}_{t,3n_t+2}+\widetilde{\gamma}_{t,j,2}\boldsymbol{b}_{t,3n_t+3}\text{ for }t\text{ such that }j\le n_t,$$

where $\widetilde{\gamma}_{t,j,1},\widetilde{\gamma}_{t,j,2}\xleftarrow{\mathsf{U}}\mathbb{F}_q$ and other elements $\boldsymbol{f}_{1,0},\{\boldsymbol{e}_{1,t,i}\}_{t=1,\ldots,d;i=1,\ldots,j-1},\{\boldsymbol{e}_{0,t,i}\}_{t=1,\ldots,d;i=j+1,\ldots,n_t}$, $\{\boldsymbol{f}_{t,i}\}_{t=1,\ldots,d;i=1,\ldots,n_t},\boldsymbol{f}_{1,d+1}$ are generated using $\{\mathbb{B}_t\}_{t=0,\ldots,d}$ and freshly generated $\omega,z_0,\gamma_0\xleftarrow{\mathsf{U}}\mathbb{F}_q,\{z_{t,i},\gamma_{t,i,1},\gamma_{t,i,2}\xleftarrow{\mathsf{U}}\mathbb{F}_q\}_{t=1,\ldots,d;i=1,\ldots,j-1},\{\gamma_{t,i,1},\gamma_{t,i,2}\xleftarrow{\mathsf{U}}\mathbb{F}_q\}_{t=1,\ldots,d;i=j+1,\ldots,n_t},\vec{z}_{d+1}\xleftarrow{\mathsf{U}}\mathbb{F}_q^2,\gamma_{d+1}\xleftarrow{\mathsf{U}}\mathbb{F}_q$. $\mathcal{C}_j$ also generates a part of semi-functional space basis as: $\widetilde{\boldsymbol{b}}_{t,n_t+j}:=r_t\boldsymbol{b}_{t,n_t+j}$ for $t=1,\ldots,d$ where $r_t\xleftarrow{\mathsf{U}}\mathbb{F}_q$ for $t=1,\ldots,d$. $\mathcal{C}_j$ then sets new bases $\widetilde{\mathbb{B}}_t:=(\boldsymbol{b}_{t,1},\ldots,\boldsymbol{b}_{t,n_t+j-1},\widetilde{\boldsymbol{b}}_{t,n_t+j},\boldsymbol{b}_{t,n_t+j+1},\ldots,\boldsymbol{b}_{t,3n_t+3})$ for $t=1,\ldots,d$. Note that new basis $\widetilde{\mathbb{B}}_t$ is compatible with $\widehat{\mathbb{B}}^*_t$ for $t=1,\ldots,d$, since $\widehat{\mathbb{B}}^*_t$ has no basis vectors for the semi-functional part. $\mathcal{C}_j$ gives a Prob^j instance $\varrho:=(\mathsf{param}_{\vec{n}},\{\mathbb{B}_t,\widehat{\mathbb{B}}^*_t\}_{t=0,d+1},\{\widetilde{\mathbb{B}}_t,\widehat{\mathbb{B}}^*_t\}_{t=1,\ldots,d},\boldsymbol{f}_{1,0},\{\boldsymbol{e}_{1,t,i}\}_{t=1,\ldots,d;i=1,\ldots,j-1},\widetilde{\boldsymbol{e}}_{\beta,t,j},\{\boldsymbol{e}_{0,t,i}\}_{t=1,\ldots,d;i=j+1,\ldots,n_t}$, $\{\boldsymbol{f}_{t,i}\}_{t=1,\ldots,d;i=1,\ldots,n_t},\boldsymbol{f}_{1,d+1})$ to $\mathcal{B}$. If $\mathcal{B}$ finally outputs β', then $\mathcal{C}_j$ outputs β'. Since the distribution of ϱ is equivalent to that of the output of Expl^{j-1} (resp. Expl^j) when $\beta=0$ (resp. $\beta=1$), we have $\mathsf{Adv}^{\mathsf{Prob}^j}_{\mathcal{B}}(\lambda)\le\mathsf{Adv}^{\mathsf{BP1}}_{\mathcal{C}_j}(\lambda)$.

By combining Lemmas 15 and 16 in [28], we obtain that, for any $\mathcal{C}_j$, there exists $\mathcal{E}_j$ such that $\mathsf{Adv}^{\mathsf{BP1}}_{\mathcal{C}_j}(\lambda)\le\mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}_j}(\lambda)+5/q$. Therefore, there are probabilistic machines $\mathcal{E}_j$, whose running times are essentially the same as that of $\mathcal{B}$, such that for any security parameter λ, $\mathsf{Adv}^{\mathsf{Prob}^j}_{\mathcal{B}}(\lambda):=\left|\mathsf{Pr}\left[\mathsf{Expl}^{j-1}_{\mathcal{B}}(1^\lambda)\to1\right]-\mathsf{Pr}\left[\mathsf{Expl}^{j}_{\mathcal{B}}(1^\lambda)\to1\right]\right|\le\mathsf{Adv}^{\mathsf{BP1}}_{\mathcal{C}_j}(\lambda)\le\mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}_j}(\lambda)+5/q$ for $j=1,\ldots,n_{\max}$. This completes the proof of Claim 1. □

Definition 16 (Problem 2) *Problem 2 is to guess $\beta\in\{0,1\}$, given $(\mathsf{param}_{\vec{n}},\{\widehat{\mathbb{B}}_t,\mathbb{B}^*_t\}_{t=0,..,d},\mathbb{B}_{d+1},\mathbb{B}^*_{d+1},\boldsymbol{h}^*_{\beta,0},\boldsymbol{e}_0,\{\boldsymbol{h}^*_{\beta,t,i},\boldsymbol{e}_{t,i}\}_{t=1,..,d;i=1,..,n_t},\{\boldsymbol{h}^*_{d+1,i}\}_{i=1,2})\xleftarrow{\mathsf{R}}\mathcal{G}^{\mathsf{P2}}_\beta(1^\lambda,\vec{n})$, where*

$$\begin{aligned}
&\mathcal{G}^{\mathsf{P2}}_\beta(1^\lambda,\vec{n}):\quad(\mathsf{param}_{\vec{n}},\{\mathbb{B}_t,\mathbb{B}^*_t\}_{t=0,\ldots,d+1})\xleftarrow{\mathsf{R}}\mathcal{G}_{\mathsf{ob}}(1^\lambda,\vec{n}),\\
&\widehat{\mathbb{B}}_0:=(\boldsymbol{b}_{0,1},\boldsymbol{b}_{0,3},\boldsymbol{b}_{0,4}),\quad\widehat{\mathbb{B}}_t:=(\boldsymbol{b}_{t,1},\ldots,\boldsymbol{b}_{t,n_t},\boldsymbol{b}_{t,2n_t+1},\ldots,\boldsymbol{b}_{t,3n_t+2})\ \text{ for }t=1,\ldots,d,\\
&\sigma,\tau\xleftarrow{\mathsf{U}}\mathbb{F}_q^\times,\omega,\delta,\delta_0\xleftarrow{\mathsf{U}}\mathbb{F}_q,\quad\boldsymbol{h}^*_{0,0}:=(\delta,0,\delta_0,0)_{\mathbb{B}^*_0},\ \boldsymbol{h}^*_{1,0}:=(\delta,\sigma,\delta_0,0)_{\mathbb{B}^*_0},\ \boldsymbol{e}_0:=(\omega,\tau,0,0)_{\mathbb{B}_0},\\
&\boldsymbol{h}^*_{d+1,i}:=\delta\boldsymbol{b}^*_{d+1,i}\ \text{ for }i=1,2,\\
&\text{for }t=1,\ldots,d;\ i=1,\ldots,n_t;\quad\vec{e}_{t,i}:=(0^{i-1},\ 1,\ 0^{n_t-i})\in\mathbb{F}_q^{n_t},\quad\vec{\delta}_{t,i}\xleftarrow{\mathsf{U}}\mathbb{F}_q^{n_t},\\
&\qquad\qquad\overbrace{\qquad}^{n_t}\ \overbrace{\qquad\quad}^{n_t+1}\ \overbrace{\qquad}^{n_t}\ \overbrace{\quad}^{2}\\
&\boldsymbol{h}^*_{0,t,i}:=\ (\ \delta\vec{e}_{t,i},\ 0^{n_t+1},\ \vec{\delta}_{t,i},\ 0^2\ )_{\mathbb{B}^*_t},\\
&\boldsymbol{h}^*_{1,t,i}:=\ (\ \delta\vec{e}_{t,i},\ \sigma\vec{e}_{t,i},\ 0,\ \vec{\delta}_{t,i},\ 0^2\ )_{\mathbb{B}^*_t},\\
&\boldsymbol{e}_{t,i}:=\ (\ \omega\vec{e}_{t,i},\ \tau\vec{e}_{t,i},\ 0,\ 0^{n_t},\ 0^2\ )_{\mathbb{B}_t}\\
&\text{return }(\mathsf{param}_{\vec{n}},\{\widehat{\mathbb{B}}_t,\mathbb{B}^*_t\}_{t=0,..,d},\mathbb{B}_{d+1},\mathbb{B}^*_{d+1},\boldsymbol{h}^*_{\beta,0},\boldsymbol{e}_0,\{\boldsymbol{h}^*_{\beta,t,i},\boldsymbol{e}_{t,i}\}_{t=1,..,d;i=1,..,n_t},\{\boldsymbol{h}^*_{d+1,i}\}_{i=1,2}).
\end{aligned}$$

for $\beta\xleftarrow{\mathsf{U}}\{0,1\}$. For a probabilistic machine $\mathcal{B}$, the advantage of $\mathcal{B}$ for Problem 2, $\mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}}(\lambda)$, is similarly defined as in Definition 15.

Lemma 2 *For any adversary $\mathcal{B}$, there exists a probabilistic machine $\mathcal{E}$, whose running time is essentially the same as that of $\mathcal{B}$, such that for any security parameter λ, $\mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}}(\lambda) \leq \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}}(\lambda) + 5/q$.*

Proof. Lemma 2 is proven in a similar manner to the combination of Lemmas 15 and 18 in [28]. These lemmas prove the security of Basic Problem 2 in [28], i.e., for any adversary $\mathcal{B}$, there exists a probabilistic machine $\mathcal{E}$, whose running time is essentially the same as that of $\mathcal{B}$, such that for any security parameter λ, $\mathsf{Adv}^{\mathsf{BP2}}_{\mathcal{B}}(\lambda) \leq \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}}(\lambda) + 5/q$. Basic Problem 2 in [28] is the same as Problem 2 in this paper except for the total dimensions of the spaces and forms of vectors in randomness space $\mathsf{span}\langle \boldsymbol{b}^*_{t,2n_t+2}, \ldots, \boldsymbol{b}^*_{t,3n_t+1}\rangle$ in $\boldsymbol{h}^*_{\beta,t,i}$ (i.e., random coefficient vectors $\vec{\delta}_{t,i}$ are used in $\boldsymbol{h}^*_{\beta,t,i}$ of Problem 2 while $\delta_0\vec{e}_{t,i}$ are used in $\boldsymbol{y}^*_{\beta,t,i}$ of Basic Problem 2), which are not essentially related to the proofs of Lemmas 15 and 18 in [28]. This implies that the proof of Lemma 2 is given as the same manner as the combination of the proofs of Lemmas 15 and 18 in [28]. □

Definition 17 (Problem 3) *Problem 3 is to guess $\beta \in \{0,1\}$, given $(\mathsf{param}_{\vec{n}}, \{\widehat{\mathbb{B}}_t, \mathbb{B}^*_t\}_{t=0,d+1}, \{\mathbb{B}_t, \mathbb{B}^*_t\}_{t=1,..,d}, \boldsymbol{h}^*_{\beta,0}, \boldsymbol{e}_0, \{\boldsymbol{h}^*_{t,i}\}_{t=1,..,d;i=1,..,n_t}, \{\boldsymbol{h}^*_{\beta,d+1,i}, \boldsymbol{e}_{d+1,i}\}_{i=1,2}) \xleftarrow{\mathsf{R}} \mathcal{G}^{\mathsf{P3}}_{\beta}(1^\lambda, \vec{n})$, where*

$$
\begin{aligned}
&\mathcal{G}^{\mathsf{P3}}_{\beta}(1^\lambda, \vec{n}): \quad (\mathsf{param}_{\vec{n}}, \{\mathbb{B}_t, \mathbb{B}^*_t\}_{t=0,\ldots,d+1}) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}(1^\lambda, \vec{n}),\\
&\widehat{\mathbb{B}}_0 := (\boldsymbol{b}_{0,1}, \boldsymbol{b}_{0,3}, \boldsymbol{b}_{0,4}), \quad \widehat{\mathbb{B}}_{d+1} := (\boldsymbol{b}_{d+1,1}, \boldsymbol{b}_{d+1,2}, \boldsymbol{b}_{d+1,5}, \ldots, \boldsymbol{b}_{d+1,7}),\\
&\sigma, \tau \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times, \ \omega, \delta, \delta_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q, \quad \boldsymbol{h}^*_{0,0} := (\delta, 0, \delta_0, 0)_{\mathbb{B}^*_0}, \ \boldsymbol{h}^*_{1,0} := (\delta, \sigma, \delta_0, 0)_{\mathbb{B}^*_0}, \ \boldsymbol{e}_0 := (\omega, \tau, 0, 0)_{\mathbb{B}_0},\\
&\boldsymbol{h}^*_{t,i} := \delta \boldsymbol{b}^*_{t,i} \ \text{ for } t = 1, \ldots, d; \ i = 1, \ldots, n_t, \quad U_{d+1} \xleftarrow{\mathsf{U}} GL(2, \mathbb{F}_q), \quad Z_{d+1} := (U^{-1}_{d+1})^{\mathrm{T}},\\
&\text{for } i = 1, 2; \quad \vec{e}_{d+1,i} := (0^{i-1}, \ 1, \ 0^{2-i}), \quad \vec{\delta}_{d+1,i} \xleftarrow{\mathsf{U}} \mathbb{F}_q^2,\\
&\quad \boldsymbol{h}^*_{0,d+1,i} := \ (\ \delta\vec{e}_{d+1,i}, \quad 0^2 \quad \vec{\delta}_{d+1,i}, \quad 0 \quad)_{\mathbb{B}^*_{d+1}},\\
&\quad \boldsymbol{h}^*_{1,d+1,i} := \ (\ \delta\vec{e}_{d+1,i}, \quad \sigma\vec{e}_{d+1,i}U_{d+1}, \quad \vec{\delta}_{d+1,i}, \quad 0 \quad)_{\mathbb{B}^*_{d+1}},\\
&\quad \boldsymbol{e}_{d+1,i} := \ (\ \omega\vec{e}_{d+1,i}, \quad \tau\vec{e}_{d+1,i}Z_{d+1}, \quad 0^2, \quad 0 \quad)_{\mathbb{B}_{d+1}},\\
&\text{return} \ \ (\mathsf{param}_{\vec{n}}, \{\widehat{\mathbb{B}}_t, \mathbb{B}^*_t\}_{t=0,d+1}, \{\mathbb{B}_t, \mathbb{B}^*_t\}_{t=1,..,d},\\
&\qquad \boldsymbol{h}^*_{\beta,0}, \boldsymbol{e}_0, \{\boldsymbol{h}^*_{t,i}\}_{t=1,..,d;i=1,..,n_t}, \{\boldsymbol{h}^*_{\beta,d+1,i}, \boldsymbol{e}_{d+1,i}\}_{i=1,2}),
\end{aligned}
$$

for $\beta \xleftarrow{\mathsf{U}} \{0,1\}$. For a probabilistic machine $\mathcal{B}$, the advantage of $\mathcal{B}$ for Problem 3, $\mathsf{Adv}^{\mathsf{P3}}_{\mathcal{B}}(\lambda)$, is similarly defined as in Definition 15.

Lemma 3 *For any adversary $\mathcal{B}$, there is a probabilistic machine $\mathcal{E}$, whose running time is essentially the same as that of $\mathcal{B}$, such that for any security parameter λ, $\mathsf{Adv}^{\mathsf{P3}}_{\mathcal{B}}(\lambda) \leq \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}}(\lambda) + 5/q$.*

Proof. Lemma 3 is proven in a similar manner to Lemmas 2 in [28]. The lemma proves the security of Problem 2 in [28], i.e., for any adversary $\mathcal{B}$, there exists a probabilistic machine $\mathcal{E}$, whose running time is essentially the same as that of $\mathcal{B}$, such that for any security parameter λ, $\mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}}(\lambda) \leq \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}}(\lambda) + 5/q$. Problem 2 in [28] is the same as Problem 3 in this paper except for the total dimensions of the spaces and the number of vector elements, which are not essentially related to the proofs of Lemma 2 in [28]. This implies that the proof of Lemma 3 is given as the same manner as the proof of Lemmas 2 in [28]. □

Lemma 4 (Lemma 3 in [28]) *For $p \in \mathbb{F}_q$, let $C_p := \{(\vec{x}, \vec{v}) | \vec{x} \cdot \vec{v} = p\} \subset V \times V^*$ where V is n-dimensional vector space ${\mathbb{F}_q}^n$, and V^* its dual. For all $(\vec{x}, \vec{v}) \in C_p$, for all $(\vec{r}, \vec{w}) \in C_p$,*

$$\Pr_{Z \xleftarrow{\mathsf{U}} GL(n, \mathbb{F}_q),} [\vec{x}U = \vec{r} \ \wedge \ \vec{v}Z = \vec{w}] = \frac{1}{\sharp C_p},$$

where $U := (Z^{-1})^{\mathrm{T}}$.

E.5 Lemmas for Evaluating Advantage Gaps

Lemma 5 *For any adversary $\mathcal{A}$, $\mathsf{Adv}^{(0)}_{\mathcal{A}}(\lambda) = \mathsf{Adv}^{(0')}_{\mathcal{A}}(\lambda)$.*

Lemma 5 follows from Theorem 1. □

Lemma 6 *For any adversary $\mathcal{A}$, there exists a probabilistic machine $\mathcal{B}_1$, whose running time is essentially the same as that of $\mathcal{A}$, such that for any security parameter λ, $|\mathsf{Adv}^{(0')}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(1)}_{\mathcal{A}}(\lambda)| \leq \mathsf{Adv}^{\mathsf{P1}}_{\mathcal{B}_1}(\lambda) + (d+1)/q$.*

Proof. In order to prove Lemma 6, we construct a probabilistic machine $\mathcal{B}_1$ against Problem 1 by using any adversary $\mathcal{A}$ in a security game (Game 0' or 1) as a black box as follows:

1. $\mathcal{B}_1$ is given Problem 1 instance $(\mathsf{param}_{\vec{n}}, \{\mathbb{B}_t, \widehat{\mathbb{B}}^*_t\}_{t=0,\ldots,d+1}, \boldsymbol{f}_{\beta,0}, \{\boldsymbol{e}_{\beta,t,i}, \boldsymbol{f}_{t,i}\}_{t=1,\ldots,d;i=1,\ldots,n_t}, \boldsymbol{f}_{\beta,d+1})$.
2. $\mathcal{B}_1$ plays a role of the challenger in the security game against adversary $\mathcal{A}$.
3. At the first step of the game, $\mathcal{B}_1$ sets $\widehat{\mathbb{B}}_0 := (\boldsymbol{b}_{0,1}, \boldsymbol{b}_{0,4})$, $\widehat{\mathbb{B}}_t := (\boldsymbol{b}_{t,1}, \ldots, \boldsymbol{b}_{t,n_t}, \boldsymbol{b}_{t,3n_t+2}, \boldsymbol{b}_{t,3n_t+3})$ for $t = 1, \ldots, d$, $\widehat{\mathbb{B}}_{d+1} := (\boldsymbol{b}_{d+1,1}, \boldsymbol{b}_{d+1,2}, \boldsymbol{b}_{d+1,7})$, $\widehat{\mathbb{B}}'^*_t := (\boldsymbol{b}^*_{t,1}, \ldots, \boldsymbol{b}^*_{t,n_t}, \boldsymbol{b}^*_{t,2n_t+2}, \ldots, \boldsymbol{b}^*_{t,3n_t+1})$ for $t = 1, \ldots, d$, $\widehat{\mathbb{B}}^*_{d+1} := (\boldsymbol{b}^*_{d+1,1}, \boldsymbol{b}^*_{d+1,2}, \boldsymbol{b}^*_{d+1,5}, \boldsymbol{b}^*_{d+1,6})$. $\mathcal{B}_1$ obtains $\widehat{\mathbb{B}}_t$ and $\widehat{\mathbb{B}}'^*_t$ from $\mathbb{B}_t$ and $\widehat{\mathbb{B}}^*_t$ in the Problem 1 instance, and returns $\mathsf{pk} := (1^\lambda, \mathsf{hk}, \mathsf{param}_{\vec{n}}, \{\widehat{\mathbb{B}}_t\}_{t=0,..,d+1}, \{\widehat{\mathbb{B}}^*_t\}_{t=1,..,d+1}, \boldsymbol{b}^*_{0,3})$ to $\mathcal{A}$, where $\mathsf{hk} \xleftarrow{\mathsf{R}} \mathsf{KH}_\lambda$.
4. When a KeyGen reveal query is issued for attribute set Γ, $\mathcal{B}_1$ answers a correct secret key (Eqs. (2), (3), (4)) computed by using $\{\widehat{\mathbb{B}}^*_t\}_{t=0,..,d+1}$, i.e., normal key. When a signature reveal query is issued for access structure $\mathbb{S}$, $\mathcal{B}_1$ answers a correct signature (Eqs. (5), (6), (7)) computed by using $\{\widehat{\mathbb{B}}^*_t\}_{t=0,..,d+1}$, i.e., normal signature.
5. When $\mathcal{B}_1$ receives an output $(m', \mathbb{S}', \vec{\boldsymbol{s}}'^*)$ from $\mathcal{A}$ (where $\mathbb{S}' := (M, \rho)$), $\mathcal{B}_1$ calculates verification text $(\boldsymbol{c}_0, \ldots, \boldsymbol{c}_{\ell+1})$ as follows:

$$\begin{aligned}
&\boldsymbol{c}_0 := (-s_0 - s_{\ell+1})\boldsymbol{f}_{\beta,0} + \eta_0 \boldsymbol{b}_{0,4},\\
&\text{for } i = 1, \ldots, \ell,\\
&\quad \text{if } \rho(i) = (t, \vec{v}_i),\ \boldsymbol{c}_i := s_i \boldsymbol{f}_{t,1} + \theta_i \textstyle\sum_{j=1}^{n_t} v_{i,j} \boldsymbol{b}_{t,j} + \sum_{j=1}^{n_t} \pi_{i,j} \boldsymbol{e}_{\beta,t,j} + \sum_{j=1}^{2} \eta_{i,j} \boldsymbol{b}_{t,3n+1+j},\\
&\quad \text{if } \rho(i) = \neg(t, \vec{v}_i),\ \ \boldsymbol{c}_i := s_i \textstyle\sum_{j=1}^{n_t} v_{i,j} \boldsymbol{f}_{t,j} + \sum_{j=1}^{n_t} \pi_{i,j} \boldsymbol{e}_{\beta,t,j} + \sum_{j=1}^{2} \eta_{i,j} \boldsymbol{b}_{t,3n+1+j},\\
&\boldsymbol{c}_{\ell+1} := s_{\ell+1} \boldsymbol{f}_{\beta,d+1} + \theta_{\ell+1} \left(-\mathsf{H}^{\lambda,\mathsf{D}}_{\mathsf{hk}}(m' \,||\, \mathbb{S}') \boldsymbol{b}_{d+1,1} + \boldsymbol{b}_{d+1,2}\right) + \eta_{\ell+1} \boldsymbol{b}_{d+1,7},
\end{aligned}$$

where $\vec{f} \xleftarrow{\mathsf{R}} {\mathbb{F}_q}^r$, $(s_1, \ldots, s_\ell)^{\mathrm{T}} := M \cdot \vec{f}^{\mathrm{T}}$, $s_0 := \vec{1} \cdot \vec{f}^{\mathrm{T}}$, $\eta_0, \theta_i \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times, \vec{\eta}_{\ell+1} \xleftarrow{\mathsf{U}} {\mathbb{F}_q}^2 \setminus \vec{0}, \{\pi_{i,j} \xleftarrow{\mathsf{U}} \mathbb{F}_q\}_{j=1,\ldots,n_t}, \{\eta_{i,j} \xleftarrow{\mathsf{U}} \mathbb{F}_q\}_{j=1,2}$ for $i = 1, \ldots, \ell+1$, and $\boldsymbol{f}_{\beta,0}, \boldsymbol{e}_{\beta,t,j}, \boldsymbol{f}_{t,j}, \boldsymbol{f}_{\beta,d+1}$ $(j = 1, \ldots, n_t)$ are from the Problem 1 instance. $\mathcal{B}_1$ verifies the signature $(m', \mathbb{S}', \vec{\boldsymbol{s}}'^*)$ using Ver with the above $(\boldsymbol{c}_0, \ldots, \boldsymbol{c}_{\ell+1})$, and outputs $\beta' := 1$ if the verification succeeds, $\beta' := 0$ otherwise.

When $\beta = 0$, it is straightforward that the distribution by $\mathcal{B}_1$'s simulation is equivalent to that in Game 0'. When $\beta = 1$, the distribution by $\mathcal{B}_1$'s simulation is equivalent to that in Game 1 except for the case that $\omega = 0$ or $(\vec{\gamma}_{i,j})_{i,j} \in \mathbb{F}_q^{2\times 2} \setminus GL(2, \mathbb{F}_q)$ for some $i = 1, \ldots, d$, i.e., except with probability $(d+1)/q$. □

Lemma 7 *For any adversary $\mathcal{A}$, there exists a probabilistic machine $\mathcal{B}_2$, whose running time is essentially the same as that of $\mathcal{A}$, such that for any security parameter λ, $|\mathsf{Adv}^{(1)}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(2)}_{\mathcal{A}}(\lambda)| \leq \mathsf{Adv}^{\mathsf{P1}}_{\mathcal{B}_2}(\lambda) + (d+1)/q$.*

Proof. In order to prove Lemma 7, we construct a probabilistic machine $\mathcal{B}_2$ against Problem 1 by using any adversary $\mathcal{A}$ in a security game (Game 1 or 2) as a black box.

The simulation is very similar to that of a machine $\mathcal{B}_1$ in the proof of Lemma 6. Steps 1 – 4 of $\mathcal{B}_2$ are the same as those of $\mathcal{B}_1$. The last step is:

5. When $\mathcal{B}_2$ receives an output $(m', \mathbb{S}', \vec{\boldsymbol{s}}'^*)$ from $\mathcal{A}$ (where $\mathbb{S}' := (M, \rho)$), $\mathcal{B}_2$ calculates verification text $(\boldsymbol{c}_0, \ldots, \boldsymbol{c}_{\ell+1})$ as follows:

$$\begin{aligned}
&\boldsymbol{c}_0 := (\ -s_0 - s_{\ell+1},\ -s'_0,\ 0,\ \eta_0\ )_{\mathbb{B}_0},\\
&\text{for } i = 1, \ldots, \ell,\\
&\quad \text{if } \rho(i) = (t, \vec{v}_i),\quad \boldsymbol{c}_i := (\ s_i\vec{e}_{t,1} + \theta_i\vec{v}_i,\ s'_i\vec{e}_{t,1} + \theta'_i\vec{v}_i,\ 0,\ 0^{n_t},\ \vec{\eta}_i\ )_{\mathbb{B}_t} + \textstyle\sum_{j=1}^{n_t} \pi_{i,j}\boldsymbol{e}_{\beta,t,j},\\
&\quad \text{if } \rho(i) = \neg(t, \vec{v}_i),\quad \boldsymbol{c}_i := (\ s_i\vec{v}_i,\ s'_i\vec{v}_i,\ 0,\ 0^{n_t},\ \vec{\eta}_i\ )_{\mathbb{B}_t} + \textstyle\sum_{j=1}^{n_t} \pi_{i,j}\boldsymbol{e}_{\beta,t,j},\\
&\boldsymbol{c}_{\ell+1} := (\ s_{\ell+1} - \theta_{\ell+1}\mathsf{H}^{\lambda,\mathsf{D}}_{\mathsf{hk}}(m' \,||\, \mathbb{S}'),\ \theta_{\ell+1},\ \vec{w}_{\ell+1},\ 0^2,\ \eta_{\ell+1}\ )_{\mathbb{B}_{d+1}},
\end{aligned}$$

where $\vec{f}, \vec{f}' \xleftarrow{\mathsf{R}} {\mathbb{F}_q}^r$, $(s_1, \ldots, s_\ell)^{\mathrm{T}} := M \cdot \vec{f}^{\mathrm{T}}$, $s_0 := \vec{1} \cdot \vec{f}^{\mathrm{T}}$, $(s'_1, \ldots, s'_\ell)^{\mathrm{T}} := M \cdot \vec{f}'^{\mathrm{T}}$, $s'_0 := \vec{1} \cdot \vec{f}'^{\mathrm{T}}$, $s_{\ell+1}, \theta_i, \theta'_i, \theta_{\ell+1}, \pi_{i,j} \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $\vec{\eta}_i, \vec{w}_{\ell+1} \xleftarrow{\mathsf{U}} \mathbb{F}_q^2$, $\eta_{\ell+1} \xleftarrow{\mathsf{U}} \mathbb{F}_q$ for $i = 1, \ldots, \ell; j = 1, \ldots, n_t$, and $\boldsymbol{e}_{\beta,t,j}$ $(j = 1, \ldots, n_t)$ are from the Problem 1 instance. $\mathcal{B}_2$ verifies the signature $(m', \mathbb{S}', \vec{\boldsymbol{s}}'^*)$ using Ver with the above $(\boldsymbol{c}_0, \ldots, \boldsymbol{c}_{\ell+1})$, and outputs $\beta' := 0$ if the verification succeeds, $\beta' := 1$ otherwise.

We note that $\boldsymbol{c}_0$ in Game 1 is conceptually changed to that in Game 2 since $\boldsymbol{b}^*_{0,2}$ is hidden from the adversary (in both games). As in the proof of 6, when $\beta = 0$, it is straightforward that the distribution by $\mathcal{B}_2$'s simulation is equivalent to that in Game 1. When $\beta = 1$, the distribution by $\mathcal{B}_2$'s simulation is equivalent to that in Game 2 except for the case that $\omega = 0$ or $(\vec{\gamma}_{i,j})_{i,j} \in \mathbb{F}_q^2 \setminus GL(2, \mathbb{F}_q)$ for some $i = 1, \ldots, d$, i.e., except with probability $(d+1)/q$. □

Lemma 8 *For any adversary $\mathcal{A}$, there exists a probabilistic machine $\mathcal{B}_{3\text{-}1}$, whose running time is essentially the same as that of $\mathcal{A}$, such that for any security parameter λ, $|\mathsf{Adv}^{(3\text{-}(h-1)\text{-}7)}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(3\text{-}h\text{-}1)}_{\mathcal{A}}(\lambda)| \leq \mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}_{3\text{-}h\text{-}1}}(\lambda) + 2/q$, where $\mathcal{B}_{3\text{-}h\text{-}1}(\cdot) := \mathcal{B}_{3\text{-}1}(h, \cdot)$.*

Proof. In order to prove Lemma 8, we construct a probabilistic machine $\mathcal{B}_{3\text{-}1}$ against Problem 2 by using an adversary $\mathcal{A}$ in a security game (Game 3-$(h-1)$-7 or 3-h-1) as a black box.

The simulation is very similar to that of a machine $\mathcal{B}_2$ in [28]. We summarize below:

First, $\mathcal{B}_{3\text{-}1}$ is given an integer h and a Problem 2 instance. $\mathcal{B}_{3\text{-}1}$ plays a role of the challenger in the security game against adversary $\mathcal{A}$. At the first step of the game, $\mathcal{B}_{3\text{-}1}$ provides $\mathcal{A}$ a public key pk of Game 3-$(h-1)$-7 (and 3-h-1), which is calculated from the Problem 2 instance.

When the ι-th key query is issued (for attribute $\Gamma := \{(t, \vec{x}_t := (x_{t,1}, \ldots, x_{t,n_t}))\}$), $\mathcal{B}_{3\text{-}1}$ answers a semi-functional key as in Eqs. (18), (3), (4) when $1 \leq \iota < h$, the following key

$$\begin{aligned}
&\boldsymbol{k}^*_0 := \boldsymbol{h}^*_{\beta,0},\quad \boldsymbol{k}^*_t := \textstyle\sum_{j=1}^{n_t} x_{t,j}\boldsymbol{h}^*_{\beta,t,j} \quad \text{for } (t, \vec{x}_t) \in \Gamma,\\
&\boldsymbol{k}^*_{d+1,j} := \boldsymbol{h}^*_{d+1,j} + \boldsymbol{r}^*_{d+1,j} \quad \text{where } \boldsymbol{r}^*_{d+1,j} \xleftarrow{\mathsf{U}} \mathsf{span}\langle \boldsymbol{b}^*_{d+1,5}, \boldsymbol{b}^*_{d+1,6}\rangle \quad \text{for } j = 1, 2,
\end{aligned}$$

when $\iota = h$, a normal key as in Eqs. (2), (3), (4) when $h \leq \iota$.

When a signature reveal query is issued, $\mathcal{B}_{3\text{-}1}$ calculates a normal signature as in Eqs. (5), (6), (7).

When $\mathcal{B}_{3\text{-}1}$ receives an output $(m', \mathbb{S}', \vec{\boldsymbol{s}}'^*)$ from $\mathcal{A}$ (where $\mathbb{S}' := (M, \rho)$), $\mathcal{B}_{3\text{-}1}$ calculates a 2-nd temporary form of verification text $(\boldsymbol{c}_0, \ldots, \boldsymbol{c}_{\ell+1})$ as in Eqs. (9), (12), (11) using the Problem 2 instance. $\mathcal{B}_{3\text{-}1}$ verifies the signature $(m', \mathbb{S}', \vec{\boldsymbol{s}}'^*)$ using Ver with the above $(\boldsymbol{c}_0, \ldots, \boldsymbol{c}_{\ell+1})$, and outputs $\beta' := 1$ if the verification succeeds, $\beta' := 0$ otherwise, except when $\omega = 0$ or $\delta = 0$, i.e., except with probability $2/q$. □

Lemma 9 *For any adversary $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}^{(3\text{-}h\text{-}1)}(\lambda) = \mathsf{Adv}_{\mathcal{A}}^{(3\text{-}h\text{-}2)}(\lambda)$.*

Proof. To prove Lemma 9, we will show distribution $(\mathsf{param}_{\vec{n}}, \{\widehat{\mathbb{B}}_t\}_{t=0,..,d}, \{\mathsf{sk}_\Gamma^{(j)*}\}_{j=1,..,\nu_1}, \vec{\boldsymbol{c}})$ in Game 3-h-1 and that in Game 3-h-2 are equivalent, where $\mathsf{sk}_\Gamma^{(j)*}$ is the answer to the j-th key query, and $\vec{\boldsymbol{c}}$ is the verification text $(\boldsymbol{c}_0, \ldots, \boldsymbol{c}_{\ell+1})$. By the definition of these games, we only need to consider elements in $\{\mathbb{V}_t\}_{t=1,..,d}$. We define new dual orthonormal bases $\mathbb{D}_t$ and $\mathbb{D}_t^*$ of $\mathbb{V}_t$ as follows: We generate $Z_t \xleftarrow{\mathsf{U}} \{Z_t \in GL(n_t, \mathbb{F}_q) \,|\, (0^{n_t-1}, 1) = \vec{x}_t^{(h)} \cdot (Z_t^{-1})^{\mathrm{T}}\}$ for $(t, \vec{x}_t^{(h)}) \in \Gamma^{(h)}$ with the h-th key query $\Gamma^{(h)}$, and set

$$\begin{pmatrix} \boldsymbol{d}_{t,n_t+1} \\ \vdots \\ \boldsymbol{d}_{t,2n_t+1} \end{pmatrix} := Z_t^{-1} \cdot \begin{pmatrix} \boldsymbol{b}_{t,n_t+1} \\ \vdots \\ \boldsymbol{b}_{t,2n_t+1} \end{pmatrix}, \quad \begin{pmatrix} \boldsymbol{d}^*_{t,n_t+1} \\ \vdots \\ \boldsymbol{d}^*_{t,2n_t+1} \end{pmatrix} := Z_t^{\mathrm{T}} \cdot \begin{pmatrix} \boldsymbol{b}^*_{t,n_t+1} \\ \vdots \\ \boldsymbol{b}^*_{t,2n_t+1} \end{pmatrix},$$

$$\mathbb{D}_t := (\boldsymbol{b}_{t,1}, \ldots, \boldsymbol{b}_{t,n_t}, \boldsymbol{d}_{t,n_t+1}, \ldots, \boldsymbol{d}_{t,2n_t+1}, \boldsymbol{b}_{t,2n_t+2}, \ldots, \boldsymbol{b}_{t,3n_t+3}),$$

$$\mathbb{D}_t^* := (\boldsymbol{b}^*_{t,1}, \ldots, \boldsymbol{b}^*_{t,n_t}, \boldsymbol{d}^*_{t,n_t+1}, \ldots, \boldsymbol{d}^*_{t,2n_t+1}, \boldsymbol{b}^*_{t,2n_t+2}, \ldots, \boldsymbol{b}^*_{t,3n_t+3}) \text{ for } (t, \vec{x}_t^{(h)}) \in \Gamma^{(h)}.$$

Then, $\mathbb{D}_t$ and $\mathbb{D}_t^*$ are dual orthonormal, and are distributed the same as the original bases, $\mathbb{B}_t$ and $\mathbb{B}_t^*$.

The j-th queried keys $\{\boldsymbol{k}_t^{(j)*}\}_{j=1,\ldots,\nu_1;\ (t,\vec{x}_t^{(j)})\in\Gamma^{(j)}}$, and verification text $\{\boldsymbol{c}_i\}_{i=1,\ldots,\ell}$ in Game 3-h-1 are expressed over bases $\mathbb{B}_t$ and $\mathbb{B}_t^*$ as

$$\begin{aligned}
\text{if } j \neq h,\ \boldsymbol{k}_t^{(j)*} &= (\ \delta^{(j)}\vec{x}_t^{(j)},\ 0^{n_t+1},\ \vec{\varphi}^{(j)},\ 0^2\ )_{\mathbb{B}_t^*} \\
&= (\ \delta^{(j)}\vec{x}_t^{(j)},\ 0^{n_t+1},\ \vec{\varphi}^{(j)},\ 0^2\ )_{\mathbb{D}_t^*} \text{ for } (t, \vec{x}_t^{(j)}) \in \Gamma^{(j)}, \\
\text{if } j = h,\ \boldsymbol{k}_t^{(h)*} &= (\ \delta^{(h)}\vec{x}_t^{(h)},\ \delta'^{(h)}\vec{x}_t^{(h)},\ 0,\ \vec{\varphi}^{(h)},\ 0^2\ )_{\mathbb{B}_t^*} \\
&= (\ \delta^{(h)}\vec{x}_t^{(h)},\ (\delta'^{(h)}\vec{x}_t^{(h)},\ 0)\cdot(Z_t^{-1})^{\mathrm{T}},\ \vec{\varphi}^{(h)},\ 0^2\ )_{\mathbb{B}_t^*} \\
&= (\ \delta^{(h)}\vec{x}_t^{(h)},\ 0^{n_t},\ \delta'^{(h)},\ \vec{\varphi}^{(h)},\ 0^2\ )_{\mathbb{D}_t^*} \text{ for } (t, \vec{x}_t^{(h)}) \in \Gamma^{(h)},
\end{aligned}$$

since $(0^{n_t},\ 1) = (\vec{x}_t^{(h)},\ 0)\cdot(Z_t^{-1})^{\mathrm{T}}$, and

$$\begin{aligned}
\text{if } \rho(i) = (t, \vec{v}_i) \wedge (t, \vec{x}_t^{(h)}) \in \Gamma^{(h)},\ \boldsymbol{c}_t &= (\ s_i\vec{e}_{t,1} + \theta_i\vec{v}_i,\ s'_i\vec{e}_{t,1} + \theta'_i\vec{v}_i,\ 0,\ 0^{n_t},\ \vec{\eta}_i\ )_{\mathbb{B}_t} \\
&= (\ s_i\vec{e}_{t,1} + \theta_i\vec{v}_i,\ (s'_i\vec{e}_{t,1} + \theta'_i\vec{v}_i,\ 0)\cdot Z_t,\ 0^{n_t},\ \vec{\eta}_i\ )_{\mathbb{D}_t} \\
\text{if } \rho(i) = \neg(t, \vec{v}_i) \wedge (t, \vec{x}_t^{(h)}) \in \Gamma^{(h)},\ \boldsymbol{c}_t &= (\ s_i\vec{v}_i,\ s'_i\vec{v}_i,\ 0,\ 0^{n_t},\ \vec{\eta}_i\ )_{\mathbb{B}_t} \\
&= (\ s_i\vec{v}_i,\ (s'_i\vec{v}_i,\ 0)\cdot Z_t,\ 0^{n_t},\ \vec{\eta}_i\ )_{\mathbb{D}_t}, \\
\text{if } \rho(i) = (t, \vec{v}_i) \wedge (t, \vec{x}_t^{(h)}) \notin \Gamma^{(h)},\ \boldsymbol{c}_t &= (\ s_i\vec{e}_{t,1} + \theta_i\vec{v}_i,\ s'_i\vec{e}_{t,1} + \theta'_i\vec{v}_i,\ 0,\ 0^{n_t},\ \vec{\eta}_i\ )_{\mathbb{B}_t} \\
\text{if } \rho(i) = \neg(t, \vec{v}_i) \wedge (t, \vec{x}_t^{(h)}) \notin \Gamma^{(h)},\ \boldsymbol{c}_t &= (\ s_i\vec{v}_i,\ s'_i\vec{v}_i,\ 0,\ 0^{n_t},\ \vec{\eta}_i\ )_{\mathbb{B}_t}.
\end{aligned}$$

In the light of the adversary's view, both $(\mathbb{B}_t, \mathbb{B}_t^*)$ and $(\mathbb{D}_t, \mathbb{D}_t^*)$ are consistent with public key $\mathsf{pk} := (1^\lambda, \mathsf{param}_{\vec{n}}, \{\widehat{\mathbb{B}}_t\}_{t=0,\ldots,d})$. Therefore, $\{\mathsf{sk}_\Gamma^{(j)*}\}_{j=1,\ldots,\nu_1}$ and $\vec{\boldsymbol{c}}$ can be expressed as keys, and

verification text in two ways, in Game 3-h-1 over bases $\{\mathbb{B}_t, \mathbb{B}_t^*\}_{t=0,\ldots,d}$ and in Game 3-h-2 over bases $\mathbb{B}_0, \mathbb{B}_0^*, \{\mathbb{D}_t, \mathbb{D}_t^*\}_{(t,\vec{x}^{(h)})\in\Gamma^{(h)}}, \{\mathbb{B}_t, \mathbb{B}_t^*\}_{(t,\vec{x}^{(h)})\notin\Gamma^{(h)}}$. Thus, Game 3-$h$-1 can be conceptually changed to Game 3-h-2. □

Lemma 10 *For any adversary $\mathcal{A}$, there exists a probabilistic machine $\mathcal{B}_{3\text{-}2}$, whose running time is essentially the same as that of $\mathcal{A}$, such that for any security parameter λ, $|\mathsf{Adv}_{\mathcal{A}}^{(3\text{-}h\text{-}2)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(3\text{-}h\text{-}3)}(\lambda)| \leq \mathsf{Adv}_{\mathcal{B}_{3\text{-}h\text{-}2}}^{\mathsf{P1}}(\lambda) + d/q$, where $\mathcal{B}_{3\text{-}h\text{-}2}(\cdot) := \mathcal{B}_{3\text{-}2}(h,\cdot)$.*

Proof. In order to prove Lemma 10, we construct a probabilistic machine $\mathcal{B}_{3\text{-}2}$ against Problem 1 by using any adversary $\mathcal{A}$ in a security game (Game 3-h-2 or 3-h-3) as a black box as follows:

1. $\mathcal{B}_{3\text{-}2}$ is given an integer h and a Problem 1 instance $(\mathsf{param}_{\vec{n}}, \{\mathbb{B}_t, \widehat{\mathbb{B}}_t^*\}_{t=0,\ldots,d+1}, \boldsymbol{f}_{\beta,0}, \{\boldsymbol{e}_{\beta,t,i}, \boldsymbol{f}_{t,i}\}_{t=1,\ldots,d;i=1,\ldots,n_t}, \boldsymbol{f}_{\beta,d+1})$.
2. $\mathcal{B}_{3\text{-}2}$ plays a role of the challenger in the security game against adversary $\mathcal{A}$.
3. At the first step of the game, $\mathcal{B}_{3\text{-}2}$ sets $\widehat{\mathbb{B}}_0 := (\boldsymbol{b}_{0,1}, \boldsymbol{b}_{0,3}, \boldsymbol{b}_{0,5})$, $\widehat{\mathbb{B}}_t := (\boldsymbol{b}_{t,1}, \ldots, \boldsymbol{b}_{t,n_t}, \boldsymbol{b}_{t,3n_t+2}, \boldsymbol{b}_{t,3n_t+3})$ for $t = 1, \ldots, d$. $\mathcal{B}_1$ obtains $\widehat{\mathbb{B}}_t$ from $\mathbb{B}_t$ in the Problem 3 instance, and returns $\mathsf{pk} := (1^\lambda, \mathsf{param}_{\vec{n}}, \{\widehat{\mathbb{B}}_t\}_{t=0,..,d})$ to $\mathcal{A}$.
4. When the ι-th KeyGen query is issued for attribute set Γ, $\mathcal{B}_{3\text{-}2}$ answers as follows:
   (a) when $1 \leq \iota < h$, $\mathcal{B}_{3\text{-}2}$ calculates a semi-functional secret key (Eqs. (18), (3)) by using $\mathbb{B}_0^*, \{\widehat{\mathbb{B}}_t^*\}_{t=1,..,d}$.
   (b) when $\iota = h$, $\mathcal{B}_{3\text{-}2}$ calculates a pre-semi-functional secret key (Eqs. (13), (15)) by using $\mathbb{B}_0^*, \{\widehat{\mathbb{B}}_t^*\}_{t=1,..,d}$.
   (c) when $h < \iota$, $\mathcal{B}_{3\text{-}2}$ calculates a normal secret key (Eqs. (2), (3)) by using $\mathbb{B}_0^*, \{\widehat{\mathbb{B}}_t^*\}_{t=1,..,d}$.
5. When $\mathcal{B}_{3\text{-}2}$ receives an output $(m', \mathbb{S}', \vec{\boldsymbol{s}}'^*)$ from $\mathcal{A}$ (where $\mathbb{S}' := (M, \rho)$), $\mathcal{B}_{3\text{-}2}$ calculates verification text $(\boldsymbol{c}_0, \ldots, \boldsymbol{c}_{\ell+1})$ as follows:

$$
\begin{aligned}
&\boldsymbol{c}_0 := (\ -s_0,\ -s_0',\ \zeta,\ 0,\ \eta_0\ )_{\mathbb{B}_0},\\
&\text{for } i = 1, \ldots, \ell,\\
&\quad \text{if } \rho(i) = (t, \vec{v}_i)\ \wedge\ (t, \vec{x}_t) \in \Gamma,\\
&\qquad \boldsymbol{c}_i := (\ s_i\vec{e}_{t,1} + \theta_i\vec{v}_i,\ (s_i'\vec{e}_{t,1} + \theta_i'\vec{v}_i,\ 0)\cdot Z_t,\ 0^{n_t},\ \vec{\eta}_i\ )_{\mathbb{B}_t} + \textstyle\sum_{j=1}^{n_t} \pi_{i,j}\boldsymbol{e}_{\beta,t,j},\\
&\quad \text{if } \rho(i) = \neg(t, \vec{v}_i)\ \wedge\ (t, \vec{x}_t) \in \Gamma,\\
&\qquad \boldsymbol{c}_i := (\ s_i\vec{v}_i,\ (s_i'\vec{v}_i,\ 0)\cdot Z_t,\ 0^{n_t},\ \vec{\eta}_i\ )_{\mathbb{B}_t} + \textstyle\sum_{j=1}^{n_t} \pi_{i,j}\boldsymbol{e}_{\beta,t,j},\\
&\quad \text{if } \rho(i) = (t, \vec{v}_i)\ \wedge\ (t, \vec{x}_t) \notin \Gamma,\\
&\qquad \boldsymbol{c}_i := (\ s_i\vec{e}_{t,1} + \theta_i\vec{v}_i,\ s_i'\vec{e}_{t,1} + \theta_i'\vec{v}_i,\ 0,\ 0^{n_t},\ \vec{\eta}_i\ )_{\mathbb{B}_t} + \textstyle\sum_{j=1}^{n_t} \pi_{i,j}\boldsymbol{e}_{\beta,t,j},\\
&\quad \text{if } \rho(i) = \neg(t, \vec{v}_i)\ \wedge\ (t, \vec{x}_t) \notin \Gamma,\\
&\qquad \boldsymbol{c}_i := (\ s_i\vec{v}_i,\ s_i'\vec{v}_i,\ 0,\ 0^{n_t},\ \vec{\eta}_i\ )_{\mathbb{B}_t} + \textstyle\sum_{j=1}^{n_t} \pi_{i,j}\boldsymbol{e}_{\beta,t,j},\\
&\boldsymbol{c}_{\ell+1} := (\ s_{\ell+1} - \theta_{\ell+1}\mathsf{H}_{\mathsf{hk}}^{\lambda,\mathsf{D}}(m'\,||\,\mathbb{S}'),\ \theta_{\ell+1},\ \vec{w}_{\ell+1},\ 0^2,\ \eta_{\ell+1}\ )_{\mathbb{B}_{d+1}},
\end{aligned}
$$

where $\vec{f}, \vec{f}' \xleftarrow{\mathsf{R}} {\mathbb{F}_q}^r$, $(s_1, \ldots, s_\ell)^{\mathrm{T}} := M \cdot \vec{f}^{\mathrm{T}}$, $s_0 := \vec{1} \cdot \vec{f}^{\mathrm{T}}$, $(s_1', \ldots, s_\ell')^{\mathrm{T}} := M \cdot \vec{f}'^{\mathrm{T}}$, $s_0' := \vec{1} \cdot \vec{f}'^{\mathrm{T}}$, $\zeta, \theta_i, \theta_i', \pi_{i,j} \xleftarrow{\mathsf{U}} \mathbb{F}_q$ for $i = 1, \ldots, \ell; j = 1, \ldots, n_t$, $b \xleftarrow{\mathsf{U}} \{0,1\}$, $Z_t \xleftarrow{\mathsf{U}} \{Z_t \in GL(n_t + 1, \mathbb{F}_q)\,|\,(0^{n_t}, 1) = (\vec{x}_t, 0)\cdot(Z_t^{-1})^{\mathrm{T}}\}$ for $(t, \vec{x}_t) \in \Gamma$ with the h-th queried Γ and $\boldsymbol{e}_{\beta,t,j}$ ($j = 1, \ldots, n_t$) are from the Problem 1 instance. $\mathcal{B}_{3\text{-}2}$ verifies the signature $(m', \mathbb{S}', \vec{\boldsymbol{s}}'^*)$ using Ver with the above $(\boldsymbol{c}_0, \ldots, \boldsymbol{c}_{\ell+1})$, and outputs $\beta' := 1$ if the verification succeeds, $\beta' := 0$ otherwise.

When $\beta = 0$, it is straightforward that the distribution by $\mathcal{B}_{3\text{-}2}$'s simulation is equivalent to that in Game 3-h-2. When $\beta = 1$, the distribution by $\mathcal{B}_{3\text{-}2}$'s simulation is equivalent to that in Game 3-h-3 except for the case that $(\vec{\gamma}_{i,j})_{i,j} \in \mathbb{F}_q^2 \setminus GL(2, \mathbb{F}_q)$ for some $i = 1, \ldots, d$, i.e., except with probability d/q. □

Lemma 11 *For any adversary $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}^{(3\text{-}h\text{-}3)}(\lambda) = \mathsf{Adv}_{\mathcal{A}}^{(3\text{-}h\text{-}4)}(\lambda)$.*

Proof. It is clear that the distribution of the public-key and the ι-th key query's answer for $\iota \neq h$ in Game 3-h-3 and Game 3-h-4 are exactly the same. Therefore, to prove this lemma we will show that the joint distribution of the h-th key query's answer and the challenge ciphertext in Game 3-h-3 and Game 3-h-4 are equivalent.

Therefore, we will show that s'_0 in Eq. (9) is uniformly and independently distributed from the other variables in the joint distribution of adversary $\mathcal{A}$'s view. $s'_0 := \vec{1} \cdot \vec{f'}^{\mathrm{T}}$ is only related to $(s'_1, \ldots, s'_\ell)^{\mathrm{T}} := M \cdot \vec{f'}^{\mathrm{T}}$ in Eq. (17). With respect to the joint distribution of these variables, there are five cases for each $i \in \{1, \ldots, \ell\}$. Map $\gamma(i)$ is defined in Definition 4.

1. $\gamma(i) = 1$ and $[\rho(i) = (t, \vec{v}_i) \ \wedge \ (t, \vec{x}_t) \in \Gamma \ \wedge \ \vec{v}_i \cdot \vec{x}_t = 0]$. Then, $s'_i + \theta'_i \vec{v}_i \cdot \vec{x}_t = s'_i$.
2. $\gamma(i) = 1$ and $[\rho(i) = \neg(t, \vec{v}_i) \ \wedge \ (t, \vec{x}_t) \in \Gamma \ \wedge \ \vec{v}_i \cdot \vec{x}_t \neq 0]$. Then, $s'_i \vec{v}_i \cdot \vec{x}_t$ with $\vec{v}_i \cdot \vec{x}_t \neq 0$.
3. $\gamma(i) = 0$ and $[\rho(i) = (t, \vec{v}_i) \ \wedge \ (t, \vec{x}_t) \in \Gamma]$ (i.e., $\vec{v}_i \cdot \vec{x}_t \neq 0$). Then, $s'_i + \theta'_i \vec{v}_i \cdot \vec{x}_t \in \mathbb{F}_q$ is uniformly and independently distributed from the other variables.
4. $\gamma(i) = 0$ and $[\rho(i) = \neg(t, \vec{v}_i) \ \wedge \ (t, \vec{x}_t) \in \Gamma]$ (i.e., $\vec{v}_i \cdot \vec{x}_t = 0$). Then, $s'_i \vec{v}_i \cdot \vec{x}_t = 0$.
5. $[\rho(i) = (t, \vec{v}_i) \ \wedge \ (t, \vec{x}_t) \notin \Gamma]$ or $[\rho(i) = \neg(t, \vec{v}_i) \ \wedge \ (t, \vec{x}_t) \notin \Gamma]$. Then, no s'_i.

We then observe the joint distribution (or relation) of s'_0 and the above inner-product values. Those in cases 3-5 are obviously independent from s'_0. Due to the restriction of adversary $\mathcal{A}$'s key queries, $\vec{1} \notin \mathsf{span}\langle (M_i)_{\gamma(i)=1} \rangle$. Therefore, $s'_0 := \vec{1} \cdot \vec{f'}^{\mathrm{T}}$ is independent from the joint distribution of $\{s'_i := M_i \cdot \vec{f'}^{\mathrm{T}} \mid \gamma(i) = 1\}$ (over the random selection of $\vec{f'}$), which can be given in case 1 and in case 2. Thus, s'_0 is uniformly and independently distributed from the other variables in the joint distribution. Therefore, the view of adversary $\mathcal{A}$ in the Game 3-h-3 is the same as that in Game 3-h-4.

This completes the proof of Lemma 11. □

Lemma 12 *For any adversary $\mathcal{A}$, there exists a probabilistic machine $\mathcal{B}_{3\text{-}3}$, whose running time is essentially the same as that of $\mathcal{A}$, such that for any security parameter λ, $|\mathsf{Adv}_{\mathcal{A}}^{(3\text{-}h\text{-}4)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(3\text{-}h\text{-}5)}(\lambda)| \leq \mathsf{Adv}^{\mathsf{P1}}_{\mathcal{B}_{3\text{-}h\text{-}3}}(\lambda) + d/q$, where $\mathcal{B}_{3\text{-}h\text{-}3}(\cdot) := \mathcal{B}_{3\text{-}3}(h, \cdot)$.*

Proof. Lemma 12 is proven in a similar manner to that of Lemma 10. □

Lemma 13 *For any adversary $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}^{(3\text{-}h\text{-}5)}(\lambda) = \mathsf{Adv}_{\mathcal{A}}^{(3\text{-}h\text{-}6)}(\lambda)$.*

Proof. Lemma 13 is proven in a similar manner to that of Lemma 9. □

Lemma 14 *For any adversary $\mathcal{A}$, there exists a probabilistic machine $\mathcal{B}_{3\text{-}4}$, whose running time is essentially the same as that of $\mathcal{A}$, such that for any security parameter λ, $|\mathsf{Adv}_{\mathcal{A}}^{(3\text{-}h\text{-}6)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(3\text{-}h\text{-}7)}(\lambda)| \leq \mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}_{3\text{-}h\text{-}4}}(\lambda) + 2/q$, where $\mathcal{B}_{3\text{-}h\text{-}4}(\cdot) := \mathcal{B}_{3\text{-}4}(h, \cdot)$.*

Proof. Lemma 14 is proven in a similar manner to that of Lemma 8. □

Lemma 15 *For any adversary $\mathcal{A}$, $|\mathsf{Adv}^{(3\text{-}\nu_1\text{-}7)}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(4)}_{\mathcal{A}}(\lambda)| \leq 1/q$.*

Proof. To prove Lemma 15, we will show distribution $(\mathsf{pk}, \{\mathsf{sk}^{(j)}_{\Gamma}\}_{j=1,..,\nu_1}, \{\vec{\boldsymbol{s}}^{(j)*}_{\mathbb{S}}\}_{j=1,..,\nu_2}, \vec{\boldsymbol{c}})$ in Game 3-ν_1-7 and that in Game 4 are equivalent, where $\mathsf{sk}^{(j)}_{\Gamma}$ is the answer to the j-th key query, $\vec{\boldsymbol{s}}^{(j)*}_{\mathbb{S}}$ is the answer to the j-th signature query, and $\vec{\boldsymbol{c}}$ is the verification text $(\boldsymbol{c}_0, \ldots, \boldsymbol{c}_{\ell+1})$. By the definition of these games, we only need to consider elements in $\mathbb{V}_0$. We define new dual orthonormal bases $\mathbb{D}_0$ and $\mathbb{D}^*_0$ of $\mathbb{V}_0$ as follows: We generate $\theta \xleftarrow{\mathsf{U}} \mathbb{F}^{\times}_q$, and set $\boldsymbol{d}_{0,2} := \theta \boldsymbol{b}_{0,2}$, $\boldsymbol{d}^*_{0,2} := \theta^{-1}\boldsymbol{b}^*_{0,2}$ Then, $\mathbb{D}_0 := (\boldsymbol{b}_{0,1}, \boldsymbol{d}_{0,2}, \boldsymbol{b}_{0,3}, \boldsymbol{b}_{0,4})$ and $\mathbb{D}^*_0 := (\boldsymbol{b}^*_{0,1}, \boldsymbol{d}^*_{0,2}, \boldsymbol{b}^*_{0,3}, \boldsymbol{b}^*_{0,4})$ are dual orthonormal, and are distributed the same as the original bases, $\mathbb{B}_0$ and $\mathbb{B}^*_0$.

We note that, since all queried signatures $\vec{\boldsymbol{s}}^{(j)*}_{\mathbb{S}}$ are of normal form, they are not affected by this base change.

The $\mathbb{V}_0$ components $\{\boldsymbol{k}^{(j)*}_0\}_{j=1,\ldots,\nu_1}$ in keys, and challenge ciphertext $\boldsymbol{c}_0$ in Game 3-ν_1-7 are expressed over bases $\mathbb{B}_0$ and $\mathbb{B}^*_0$ as $\boldsymbol{k}^{(j)*}_0 = (\delta^{(j)}, r^{(j)}_0, \varphi^{(j)}_0, 0)_{\mathbb{B}^*_0}$ and $\boldsymbol{c}_0 = (-s_0 - s_{\ell+1}, -s'_0, 0, \eta_0)_{\mathbb{B}_0}$. Then,

$$\boldsymbol{k}^{(j)*}_0 = (\delta^{(j)}, r^{(j)}_0, \varphi^{(j)}_0, 0)_{\mathbb{B}^*_0} = (\delta^{(j)}, r^{(j)}_0\theta, \varphi^{(j)}_0, 0)_{\mathbb{D}^*_0} = (\delta^{(j)}, \widetilde{r}^{(j)}, \varphi^{(j)}_0, 0)_{\mathbb{D}^*_0},$$

where $\widetilde{r}^{(j)} := r^{(j)}_0\theta$ which are uniformly, independently distributed since $r^{(j)}_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q$, and

$$\boldsymbol{c}_0 = (-s_0 - s_{\ell+1}, -s'_0, 0, \eta_0)_{\mathbb{B}_0} = (-s_0 - s_{\ell+1}, -s'_0\theta^{-1}, 0, \eta_0)_{\mathbb{D}_0} = (-s_0 - s_{\ell+1}, \widetilde{s}'_0, 0, \eta_0)_{\mathbb{D}_0}$$

where $\widetilde{s}'_0 := -s'_0\theta^{-1}$ which is uniformly, independently distributed since $\theta \xleftarrow{\mathsf{U}} \mathbb{F}^{\times}_q$ if $s'_0 \neq 0$.

In the light of the adversary's view, both $(\mathbb{B}_0, \mathbb{B}^*_0)$ and $(\mathbb{D}_0, \mathbb{D}^*_0)$ are consistent with public key pk. Therefore, $\{\mathsf{sk}^{(j)}_{\Gamma}\}_{j=1,\ldots,\nu_1}$ and $\vec{\boldsymbol{c}}$ can be expressed as keys and verification text in two ways, in Game 3-ν_1-7 over bases $\{\mathbb{B}_t, \mathbb{B}^*_t\}_{t=0,\ldots,d+1}$ and in Game 4 over bases $\mathbb{D}_0, \mathbb{D}^*_0, \{\mathbb{B}_t, \mathbb{B}^*_t\}_{t=1,\ldots,d+1}$. Thus, Game 3-$\nu_1$-7 can be conceptually changed to Game 4 if $s'_0 \neq 0$, i.e., except with probability $1/q$. □

Lemma 16 *For any adversary $\mathcal{A}$, there exist probabilistic machines $\mathcal{B}_5$ and $\mathcal{E}_6$, whose running time are essentially the same as that of $\mathcal{A}$, such that for any security parameter λ, $|\mathsf{Adv}^{(5\text{-}(h-1))}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(5\text{-}h)}_{\mathcal{A}}(\lambda)| \leq \mathsf{Adv}^{\mathsf{P3}}_{\mathcal{B}_{5\text{-}h}}(\lambda) + \mathsf{Adv}^{\mathsf{H},\mathsf{CR}}_{\mathcal{E}_{6\text{-}h}}(\lambda) + 3/q$, where $\mathcal{B}_{5\text{-}h}(\cdot) := \mathcal{B}_5(h, \cdot)$ and $\mathcal{E}_{6\text{-}h}(\cdot) := \mathcal{E}_6(h, \cdot)$.*

Proof. In order to prove Lemma 16, we construct a probabilistic machine $\mathcal{B}_5$ against Problem 3 by using any adversary $\mathcal{A}$ in a security game (Game 5-$(h-1)$ or 5-h) as a black box as follows:

1. $\mathcal{B}_5$ is given an integer h and a Problem 3 instance,
   $(\mathsf{param}_{\vec{n}}, \{\widehat{\mathbb{B}}_t, \mathbb{B}^*_t\}_{t=0,d+1}, \{\mathbb{B}_t, \mathbb{B}^*_t\}_{t=1,..,d}, \boldsymbol{h}^*_{\beta,0}, \boldsymbol{e}_0, \{\boldsymbol{h}^*_{t,j}\}_{t=1,..,d;j=1,..,n_t}, \{\boldsymbol{h}^*_{\beta,d+1,j}, \boldsymbol{e}_{d+1,j}\}_{j=1,2})$.
2. $\mathcal{B}_5$ plays a role of the challenger in the security game against adversary $\mathcal{A}$.
3. At the first step of the game, $\mathcal{B}_5$ provides $\mathcal{A}$ a public key $\mathsf{pk} := (1^{\lambda}, \mathsf{hk}, \mathsf{param}_{\vec{n}}, \{\widehat{\mathbb{B}}'_t\}_{t=0,\ldots,d+1}$, $\{\widehat{\mathbb{B}}^*_t\}_{t=1,\ldots,d+1}, \boldsymbol{b}^*_{0,3})$ of Game 5-$(h-1)$ (and 5-h), where $\mathsf{hk} \xleftarrow{\mathsf{R}} \mathsf{KH}_{\lambda}$, $\widehat{\mathbb{B}}'_t := (\boldsymbol{b}_{t,1}, \ldots, \boldsymbol{b}_{t,n_t}$, $\boldsymbol{b}_{t,3n_t+1})$, and $\widehat{\mathbb{B}}^*_t := (\boldsymbol{b}^*_{t,1}, \ldots, \boldsymbol{b}^*_{t,n_t}, \boldsymbol{b}^*_{t,2n_t+1}, \ldots, \boldsymbol{b}^*_{t,3n_t})$, that are obtained from the Problem 3 instance.
4. When KeyGen query is issued for attribute $\Gamma := \{(t, \vec{x}_t)\}$, $\mathcal{B}_5$ answers semi-functional key $\{\boldsymbol{k}^*_t\}_{t\in T}$ where $T := \{0, (d+1, 1), (d+1, 2)\} \cup \{t \mid 1 \leq t \leq d,\ (t, \vec{x}_t) \in \Gamma\}$, with Eqs. (18), (3), (4), that is computed by using $\{\mathbb{B}^*_t\}_{t=0,\ldots,d+1}$ of the Problem 3 instance.

5. When the ι-th signature reveal query is issued for attribute $\mathbb{S} := (M, \rho)$, $\mathcal{B}_5$ answers as follows:

   (a) When $1 \le \iota \le h-1$, $\mathcal{B}_5$ answers semi-functional signature $\vec{\boldsymbol{s}}^*$ with Eqs. (6) and (20), that is computed by using $\{\mathbb{B}^*_t\}_{t=0,\ldots,d+1}$ of the Problem 3 instance.

   (b) When $\iota = h$, $\mathcal{B}_5$ calculates $\vec{\boldsymbol{s}}^* := (\boldsymbol{s}^*_0, .., \boldsymbol{s}^*_{\ell+1})$ by using $\{\widehat{\mathbb{B}}^*_t\}_{t=0,..,d+1}, \boldsymbol{h}^*_{\beta,0}$, $\{\boldsymbol{h}^*_{t,j}\}_{t=1,..,d;j=1,..,n_t}, \{\boldsymbol{h}^*_{\beta,d+1,j}\}_{j=1,2}$ of the Problem 3 instance as follows:

$$\begin{aligned}
&\boldsymbol{s}^*_0 := \boldsymbol{h}^*_{\beta,0}, \quad \boldsymbol{s}^*_i := \textstyle\sum_{j=1}^{n} z_j \boldsymbol{h}^*_{t,j} + \boldsymbol{r}^*_i \ \text{ for } i = 1, \ldots, \ell,\\
&\boldsymbol{s}^*_{\ell+1} := \boldsymbol{h}^*_{\beta,d+1,1} + \mathsf{H}^{\lambda,\mathsf{D}}_{\mathsf{hk}}(m \,||\, \mathbb{S}) \cdot \boldsymbol{h}^*_{\beta,d+1,2},
\end{aligned}$$

   where $(\zeta_i) \xleftarrow{\mathsf{U}} \{(\zeta_i) \mid \sum_{i=1}^{\ell} \zeta_i M_i = \vec{1}\}$, and if $\rho(i) = (t, \vec{v}_i)$, then $\vec{z}_i \xleftarrow{\mathsf{U}} \{\vec{z}_i \mid \vec{z}_i \cdot \vec{v}_i = 0,\ z_{i,1} = \zeta_i\}$, if $\rho(i) = \neg(t, \vec{v}_i)$, then $\vec{z}_i \xleftarrow{\mathsf{U}} \{\vec{z}_i \mid \vec{z}_i \cdot \vec{v}_i = \zeta_i\}$, and $\boldsymbol{r}^*_i \xleftarrow{\mathsf{U}} \mathsf{span}\langle \boldsymbol{b}^*_{t,2n_t+1}, \ldots, \boldsymbol{b}^*_{t,3n_t}\rangle$ with $t := \widetilde{\rho}(i)$ for $i = 1, \ldots, \ell$.

   (c) When $\iota \ge h+1$, $\mathcal{B}_5$ answers normal signature $\vec{\boldsymbol{s}}^*$ with Eqs. (5), (6), and (7), that is computed by using $\{\mathbb{B}^*_t\}_{t=0,\ldots,\ell+1}$ of the Problem 3 instance.

6. When $\mathcal{B}_5$ receives an output $(m', \mathbb{S}', \vec{\boldsymbol{s}}'^*)$ from $\mathcal{A}$, $\mathcal{B}_5$ calculates semi-functional verification text $\vec{\boldsymbol{c}} := (\boldsymbol{c}_0, \ldots, \boldsymbol{c}_{\ell+1})$ with Eqs. (19), (12), (11) as follows: $\boldsymbol{c}_i$ for $i = 1, \ldots, \ell$ are calculated as Eq. (12) by using bases $\{\mathbb{B}_t\}_{t=1,\ldots,d}$, and using the coefficient $s_0 := \sum_{k=1}^{r} f_k$,

$$\begin{aligned}
&\alpha_l, \widetilde{\alpha}_l \xleftarrow{\mathsf{U}} \mathbb{F}_q \ \text{ for } l = 1, 2, \quad \widetilde{\boldsymbol{f}}_0 := \widetilde{\alpha}_1 \boldsymbol{e}_0 + \widetilde{\alpha}_2 \boldsymbol{b}_{0,1},\\
&\boldsymbol{f}_{d+1,j} := \alpha_1 \boldsymbol{e}_{d+1,j} + \alpha_2 \boldsymbol{b}_{d+1,j}, \quad \widetilde{\boldsymbol{f}}_{d+1,j} := \widetilde{\alpha}_1 \boldsymbol{e}_{d+1,j} + \widetilde{\alpha}_2 \boldsymbol{b}_{d+1,j} \quad \text{for } j = 1, 2;\\
&\boldsymbol{c}_0 := -s_0 \boldsymbol{b}_{0,1} - \widetilde{\boldsymbol{f}}_0 + \boldsymbol{q}_0, \quad \boldsymbol{c}_{\ell+1} := \widetilde{\boldsymbol{f}}_{d+1,1} - \mathsf{H}^{\lambda,\mathsf{D}}_{\mathsf{hk}}(m' \,||\, \mathbb{S}') \cdot \boldsymbol{f}_{d+1,1} + \boldsymbol{f}_{d+1,2} + \boldsymbol{q}_{\ell+1},
\end{aligned}$$

   where $\boldsymbol{q}_0 \xleftarrow{\mathsf{U}} \mathsf{span}\langle \boldsymbol{b}_{0,4}\rangle$, $\boldsymbol{q}_{\ell+1} \xleftarrow{\mathsf{U}} \mathsf{span}\langle \boldsymbol{b}_{d+1,7}\rangle$, and $\boldsymbol{b}_{0,1}, \boldsymbol{e}_0, \boldsymbol{b}_{d+1,j}, \boldsymbol{e}_{d+1,j}$ for $j = 1, 2$ are from the Problem 3 instance. $\mathcal{B}_5$ verifies the signature $(m', \mathbb{S}', \vec{\boldsymbol{s}}'^*)$ using Ver with the above $(\boldsymbol{c}_0, \ldots, \boldsymbol{c}_{\ell+1})$, and outputs $\beta' := 1$ if the verification succeeds, $\beta' := 0$ otherwise.

Claim 2 *The pair of signature $\vec{\boldsymbol{s}}^*$ generated in case (b) of step 5 and verification text $\vec{\boldsymbol{c}}$ generated in step 6 has the same distribution as that in Game 5-$(h-1)$ (resp. Game 5-h) when $\beta = 0$ (resp. $\beta = 1$) except with probability $1/q$ (resp. $\mathsf{Adv}^{\mathsf{H},\mathsf{CR}}_{\mathcal{E}_{6\text{-}h}}(\lambda) + 2/q$ for a probabilistic machine $\mathcal{E}_6$ with essentially same running time as that of $\mathcal{A}$, where $\mathcal{E}_{6\text{-}h}(\cdot) := \mathcal{E}_6(h, \cdot)$).*

Proof. We consider the joint distribution of $\vec{\boldsymbol{c}}$ and $\vec{\boldsymbol{s}}^*$. Clearly, a part of verification text, $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_\ell$, and a part of signature, $\boldsymbol{s}^*_1, \ldots, \boldsymbol{s}^*_\ell$, are the same as those in Game 5-$(h-1)$ and Game 5-h. Hence, we only consider $\boldsymbol{c}_0, \boldsymbol{c}_{\ell+1}, \boldsymbol{s}^*_0$, and $\boldsymbol{s}^*_{\ell+1}$.

When $\beta = 0$, it is straightforward the joint distribution of $\boldsymbol{c}_0, \boldsymbol{c}_{\ell+1}, \boldsymbol{s}^*_0$, and $\boldsymbol{s}^*_{\ell+1}$ are the same as those in Game 5-$(h-1)$ except that δ defined in Problem 3 is zero, i.e., except with probability $1/q$.

When $\beta = 1$, we need to check that w_0 in $\boldsymbol{c}_0$ (given in Eq. (19)), $\vec{w}_{\ell+1}$ in $\boldsymbol{c}_{\ell+1}$ (given in Eq. (11)), $\widetilde{r}_0$ in $\boldsymbol{s}^*_0$ and $\vec{\widetilde{r}}_{\ell+1}$ in $\boldsymbol{s}^*_{\ell+1}$ (given in Eq. (20)) are distributed as in those in Game 5-h, i.e., these are uniformly and independently distributed (with negligible probability). These are given as

$$\begin{aligned}
&w_0 = -u_0^{-1} \widetilde{s}_{\ell+1}, \quad \vec{w}_{\ell+1} = \left(\widetilde{s}_{\ell+1} - \widetilde{\theta}_{\ell+1} \cdot \mathsf{H}^{\lambda,\mathsf{D}}_{\mathsf{hk}}(m' \,||\, \mathbb{S}'), \widetilde{\theta}_{\ell+1}\right) \cdot Z_{d+1},\\
&\widetilde{r}_0 = u_0, \quad \vec{\widetilde{r}}_{\ell+1} = \left(1, \mathsf{H}^{\lambda,\mathsf{D}}_{\mathsf{hk}}(m \,||\, \mathbb{S})\right) \cdot U_{d+1},
\end{aligned}$$

where $u_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times$, $\widetilde{\theta}_{\ell+1}, \widetilde{s}_{\ell+1} \xleftarrow{\mathsf{U}} \mathbb{F}_q$, which are independent from all the other variables and $U_{d+1} \xleftarrow{\mathsf{U}} GL(2, \mathbb{F}_q), Z_{d+1} := (U_{d+1}^{-1})^{\mathrm{T}}$. Since $(m, \mathbb{S}) \neq (m', \mathbb{S}')$, $\vec{w}_{\ell+1} \cdot \vec{\widetilde{r}}_{\ell+1} = \alpha\widetilde{\theta}_{\ell+1} + \widetilde{s}_{\ell+1}$ with nonzero $\alpha \left(:= \mathsf{H}_{\mathsf{hk}}^{\lambda,\mathsf{D}}(m \,||\, \mathbb{S}) - \mathsf{H}_{\mathsf{hk}}^{\lambda,\mathsf{D}}(m' \,||\, \mathbb{S}')\right)$ except with probability $\mathsf{Adv}_{\mathcal{E}_{6\text{-}h}}^{\mathsf{H},\mathsf{CR}}(\lambda)$ for a probabilistic machine $\mathcal{E}_{6\text{-}h}$ with essentially same running time as that of $\mathcal{A}$.

Then, coefficients u_0 and $\widetilde{r}_0$ are uniformly and independently distributed, which are independent from $\vec{w}_{\ell+1} \cdot \vec{\widetilde{r}}_{\ell+1} = \alpha\widetilde{\theta}_{\ell+1} + \widetilde{s}_{\ell+1}$ since $u_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times$, $\widetilde{s}_{\ell+1}, \widetilde{\theta}_{\ell+1} \xleftarrow{\mathsf{U}} \mathbb{F}_q$ and $\alpha \neq 0$. Moreover, from Lemma 4, pair $(\vec{\widetilde{r}}_{\ell+1}, \vec{w}_{\ell+1})$ is uniformly distributed in $C_{\vec{w}_{\ell+1} \cdot \vec{\widetilde{r}}_{\ell+1}} = C_{\alpha\widetilde{\theta}_{\ell+1} + \widetilde{s}_{\ell+1}}$. Therefore, the joint distribution of $\boldsymbol{c}_0, \boldsymbol{c}_{\ell+1}, \boldsymbol{s}_0^*$, and $\boldsymbol{s}_{\ell+1}^*$ are the same as those in Game 5-h except that δ defined in Problem 3 is zero or $\vec{w}_{\ell+1} \cdot \vec{\widetilde{r}}_{\ell+1} = 0$ i.e., except with probability $\mathsf{Adv}_{\mathcal{E}_{6\text{-}h}}^{\mathsf{H},\mathsf{CR}}(\lambda) + 2/q$. This completes the proof of Claim 2.

Therefore, $|\mathsf{Adv}_{\mathcal{A}}^{(5\text{-}(h-1))}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(5\text{-}h)}(\lambda)| \leq \mathsf{Adv}_{\mathcal{B}_{5\text{-}h}}^{\mathsf{P3}}(\lambda) + \mathsf{Adv}_{\mathcal{E}_{6\text{-}h}}^{\mathsf{H},\mathsf{CR}}(\lambda) + 1/q + 2/q = \mathsf{Adv}_{\mathcal{B}_{5\text{-}h}}^{\mathsf{P3}}(\lambda) + \mathsf{Adv}_{\mathcal{E}_{6\text{-}h}}^{\mathsf{H},\mathsf{CR}}(\lambda) + 3/q$ from Shoup's difference lemma. This completes the proof of Lemma 16. □

Lemma 17 *For any adversary $\mathcal{A}$, $|\mathsf{Adv}_{\mathcal{A}}^{(5\text{-}\nu_2)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(6)}(\lambda)| \leq 1/q$.*

Proof. To prove Lemma 17, we will show distribution $(\mathsf{param}_{\vec{n}}, \{\widehat{\mathbb{B}}_t\}_{t=0,..,d+1}, \{\widehat{\mathbb{B}}_t^*\}_{t=1,..,d+1}, \boldsymbol{b}_{0,3}^*, \{\mathsf{sk}_\Gamma^{(j)}\}_{j=1,..,\nu_1}, \{\vec{\boldsymbol{s}}^{(j)*}\}_{j=1,..,\nu_2}, \boldsymbol{c})$ in Game 5-ν_2 and that in Game 6 are equivalent, where $\mathsf{sk}_\Gamma^{(j)}$ is the answer to the j-th key query, $\vec{\boldsymbol{s}}^{(j)*}$ is that to the j-th signature query, and $\vec{\boldsymbol{c}}$ is the verification text $(\boldsymbol{c}_0, \ldots, \boldsymbol{c}_{\ell+1})$. By the definition of these games, we only need to consider elements in $\mathbb{V}_0$. We define new dual orthonormal bases $\mathbb{D}_0$ and $\mathbb{D}_0^*$ of $\mathbb{V}_0$ as follows: We generate $\theta \xleftarrow{\mathsf{U}} \mathbb{F}_q$, and set

$$\boldsymbol{d}_{0,2} := (\theta, 1, 0, 0)_{\mathbb{B}} = \theta\boldsymbol{b}_{0,1} + \boldsymbol{b}_{0,2}, \quad \boldsymbol{d}_{0,1}^* := (1, -\theta, 0, 0)_{\mathbb{B}} = \boldsymbol{b}_{0,1}^* - \theta\boldsymbol{b}_{0,2}^*.$$

Let $\mathbb{D}_0 := (\boldsymbol{b}_{0,1}, \boldsymbol{d}_{0,2}, \boldsymbol{b}_{0,3}, \boldsymbol{b}_{0,4})$ and $\mathbb{D}_0^* := (\boldsymbol{d}_{0,1}^*, \boldsymbol{b}_{0,2}^*, \boldsymbol{b}_{0,3}^*, \boldsymbol{b}_{0,4}^*)$. Then, $\mathbb{D}_0$ and $\mathbb{D}_0^*$ are dual orthonormal, and are distributed the same as the original bases, $\mathbb{B}_0$ and $\mathbb{B}_0^*$.

The $\mathbb{V}_0$ components $\{\boldsymbol{k}_0^{(j)*}\}_{j=1,\ldots,\nu_1}$ in keys, $\{\boldsymbol{s}_0^{(j)*}\}_{j=1,\ldots,\nu_2}$ in signatures, and verification text $\boldsymbol{c}_0$ in Game 5-ν_2 are expressed over bases $\mathbb{B}_0$ and $\mathbb{B}_0^*$ as $\boldsymbol{k}_0^{(j)*} = (\delta^{(j)}, r_0^{(j)}, \varphi_0^{(j)}, 0)_{\mathbb{B}_0^*}$, $\boldsymbol{s}_0^{(j)*} = (\widetilde{\delta}^{(j)}, \widetilde{r}_0^{(j)}, \sigma_0^{(j)}, 0)_{\mathbb{B}_0^*}$ and $\boldsymbol{c}_0 = (-s_0 - s_{\ell+1}, w_0, 0, \eta_0)_{\mathbb{B}_0}$. Then,

$$\boldsymbol{k}_0^{(j)*} = (\delta^{(j)}, r_0^{(j)}, \varphi_0^{(j)}, 0)_{\mathbb{B}_0^*} = (\delta^{(j)}, r_0^{(j)} + \theta\delta^{(j)}, \varphi_0^{(j)}, 0)_{\mathbb{D}_0^*} = (\delta^{(j)}, \vartheta^{(j)}, \varphi_0^{(j)}, 0)_{\mathbb{D}_0^*},$$

where $\vartheta^{(j)} := r_0^{(j)} + \theta\delta^{(j)}$ which are uniformly, independently distributed since $r_0^{(j)} \xleftarrow{\mathsf{U}} \mathbb{F}_q$,

$$\boldsymbol{s}_0^{(j)*} = (\widetilde{\delta}^{(j)}, \widetilde{r}_0^{(j)}, \sigma_0^{(j)}, 0)_{\mathbb{B}_0^*} = (\widetilde{\delta}^{(j)}, \widetilde{r}_0^{(j)} + \theta\widetilde{\delta}^{(j)}, \sigma_0^{(j)}, 0)_{\mathbb{D}_0^*} = (\widetilde{\delta}^{(j)}, \widetilde{\vartheta}^{(j)}, \sigma_0^{(j)}, 0)_{\mathbb{D}_0^*}$$

where $\widetilde{\vartheta}^{(j)} := \widetilde{r}_0^{(j)} + \theta\widetilde{\delta}^{(j)}$ which are uniformly, independently distributed since $\widetilde{r}_0^{(j)} \xleftarrow{\mathsf{U}} \mathbb{F}_q$, and

$$\boldsymbol{c}_0 = (-s_0 - s_{\ell+1}, w_0, 0, \eta_0)_{\mathbb{B}_0} = (-s_0 - s_{\ell+1} - \theta w_0, w_0, 0, \eta_0)_{\mathbb{D}_0} = (\widetilde{s}_0, w_0, 0, \eta_0)_{\mathbb{D}_0}$$

where $\widetilde{s}_0 := -s_0 - s_{\ell+1} - \theta w_0$ which is uniformly, independently distributed since $\theta \xleftarrow{\mathsf{U}} \mathbb{F}_q$ if $w_0 \neq 0$.

In the light of the adversary's view, both $(\mathbb{B}_0, \mathbb{B}_0^*)$ and $(\mathbb{D}_0, \mathbb{D}_0^*)$ are consistent with public key $\mathsf{pk} := (1^\lambda, \mathsf{param}_{\vec{n}}, \{\widehat{\mathbb{B}}_t\}_{t=0,\ldots,d+1}, \{\widehat{\mathbb{B}}_t^*\}_{t=1,\ldots,d+1}, \boldsymbol{b}_{0,3}^*)$. Therefore, $\{\mathsf{sk}_\Gamma^{(j)}\}_{j=1,\ldots,\nu_1}$, $\{\vec{\boldsymbol{s}}^{(j)*}\}_{j=1,\ldots,\nu_2}$, and $\vec{\boldsymbol{c}}$ can be expressed as keys, signatures, and verification text in two ways, in Game 5-ν_2 over bases $\{\mathbb{B}_t, \mathbb{B}_t^*\}_{t=0,\ldots,d+1}$ and in Game 6 over bases $\mathbb{D}_0, \mathbb{D}_0^*, \{\mathbb{B}_t, \mathbb{B}_t^*\}_{t=1,\ldots,d+1}$. Thus, Game 5-$\nu_2$ can be conceptually changed to Game 6 if $w_0 \neq 0$, i.e., except with probability $1/q$. □

Lemma 18 *For any adversary $\mathcal{A}$, $\mathsf{Adv}^{(6)}_{\mathcal{A}}(\lambda) = 1/q$.*

Proof. Let $(\boldsymbol{s}'^*_0, \ldots, \boldsymbol{s}'^*_{\ell+1})$ be signature $\mathcal{A}$ outputs. If $e(\boldsymbol{b}_{0,1}, \boldsymbol{s}'^*_0) = 1$, the verification fails by the definition of Ver. Otherwise, the verification fails except with negligible probability regardless of the output of the adversary since coefficient $\widetilde{s}_0$ of $\boldsymbol{b}_{0,1}$ in $\boldsymbol{c}_0$ (Eq. (21)) is uniform and independent from all the other variables, and coefficient of $\boldsymbol{b}^*_{0,1}$ in $\boldsymbol{s}'^*_0$ is nonzero. Hence, $\mathsf{Adv}^{(6)}_{\mathcal{A}}(\lambda) = 1/q$. □

F Proofs of Theorems 3 and 4

Theorem 3 *The proposed MA-ABS scheme is perfectly private.*

The proof is essentially equivalent to that for Theorem 1. □

We will give a proof sketch for Theorem 4 for general form MA-ABS similarly to Theorem 2.

Theorem 4 (for General Form MA-ABS) *The proposed MA-ABS scheme is unforgeable (adaptive-predicate unforgeable) under the DLIN assumption and the existence of collision resistance hash functions.*

For any adversary $\mathcal{A}$, there exist probabilistic machines $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_{3\text{-}1}, \ldots, \mathcal{E}_{3\text{-}4}, \mathcal{E}_5, \mathcal{E}_6$, whose running times are essentially the same as that of $\mathcal{A}$, such that for any security parameter λ,

$$\begin{aligned}\mathsf{Adv}^{\mathsf{MA\text{-}ABS},\mathsf{UF}}_{\mathcal{A}}(\lambda) \leq{}& \textstyle\sum_{i=1}^{n_{\max}} \left(\mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}_{1\text{-}i}}(\lambda) + \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}_{2\text{-}i}}(\lambda)\right) \\ &+ \textstyle\sum_{h=1}^{\nu_1} \left(\mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}_{3\text{-}h\text{-}1}}(\lambda) + \sum_{i=1}^{n_{\max}} \left(\mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}_{3\text{-}h\text{-}2\text{-}i}}(\lambda) + \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}_{3\text{-}h\text{-}3\text{-}i}}(\lambda)\right) + \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}_{3\text{-}h\text{-}4}}(\lambda)\right) \\ &+ \textstyle\sum_{h=1}^{\nu_2} \left(\mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}_{5\text{-}h}}(\lambda) + \mathsf{Adv}^{\mathsf{H,CR}}_{\mathcal{E}_{6\text{-}h}}(\lambda)\right) + \epsilon, \end{aligned} \tag{22}$$

where $\mathcal{E}_{\iota\text{-}i}(\cdot) := \mathcal{E}_\iota(i, \cdot)$ for $\iota = 1, 2$ ($i = 1, \ldots, n_{\max}$), $\mathcal{E}_{\iota\text{-}h}(\cdot) := \mathcal{E}_\iota(h, \cdot)$ for $\iota = 5, 6$ ($h = 1, \ldots, \nu_2$), $\mathcal{E}_{3\text{-}h\text{-}\iota}(\cdot) := \mathcal{E}_{3\text{-}\iota}(h, \cdot)$ for $\iota = 1, 4$, $\mathcal{E}_{3\text{-}h\text{-}\iota\text{-}i}(\cdot) := \mathcal{E}_{3\text{-}\iota}(h, i, \cdot)$ for $\iota = 2, 3$ ($h = 1, \ldots, \nu_1; i = 1, \ldots, n_{\max}$), $n_{\max}$ is the maximum of dimensions n_t ($t = 1, \ldots, d$), ν_1 is the maximum number of $\mathcal{A}$'s token queries, ν_2 is the maximum number of $\mathcal{A}$'s reveal signature queries, and $\epsilon := ((2d + 16)\nu_1 + 8\nu_2 + 2d + 11)/q$.

Proof. (Sketch) The proof of this theorem is equivalent to that of Theorem 2 except the following:

1. Components of keys or verification texts for corrupt authority $t \in \mathcal{T}_{\mathsf{bad}}$ cannot be used for simulation of various types of keys or verification texts. Therefore, in the semi-functional subspace, parameters can be embedded only for honest authority $t \in \mathcal{T}_{\mathsf{good}}$, in particular, secret sharing system over just those $t \in \mathcal{T}_{\mathsf{good}}$ is used for any simulation in the security games. (The original idea is given in [19].)

2. Problems 1, 2 and 3 that do not include parameters G_0, G_1 (and δG_1 for Problem 2) cannot be used to simulate the security games of the MA-ABS scheme, because G_0, G_1 and δG_1 are employed in the security games. Therefore, modified problems, Problems 1', 2' and 3', where G_0, G_1 and δG_1 are included, are introduced and employed in the simulation of the security games of the MA-ABS scheme.

To prove Theorem 4, we consider the following games. In the games, a part framed by a box indicates coefficients which were changed in a game from the previous game, and a shadowed part indicates what was changed from the corresponding games for the (single-authority) ABS.

Game 0 : Original game (Definition 12).

Game 0' : Game 0' is the same as Game 0' for (single-authority) ABS.

Game 1 : Same as Game 0' except that the verification text $(\boldsymbol{c}_0, \ldots, \boldsymbol{c}_{\ell+1})$ for $(m', \mathbb{S}')$ with $\mathbb{S}' := (M, \rho)$ generated in Ver for verifying the output of the adversary are:

$$\boldsymbol{c}_0 := (\ -s_0 - s_{\ell+1},\ \boxed{w_0},\ 0,\ \eta_0\ )_{\mathbb{B}_0}, \tag{23}$$

$$\left.\begin{array}{l} \text{for } 1 \le i \le \ell, \\ \quad \text{if } \rho(i) = (t, \vec{v}_i) \ \wedge t \in \mathcal{T}_{\mathsf{good}},\ \ \boldsymbol{c}_i := (\ \ s_i \vec{e}_{t,1} + \theta_i \vec{v}_i,\ \ \boxed{\vec{w}_i},\ 0,\ \ 0^{n_t},\ \ \vec{\eta}_i\ )_{\mathbb{B}_t}, \\ \quad \text{if } \rho(i) = \neg(t, \vec{v}_i) \ \wedge t \in \mathcal{T}_{\mathsf{good}},\ \ \boldsymbol{c}_i := (\ \ \ s_i \vec{v}_i,\ \ \ \boxed{\vec{w}_i},\ 0,\ \ 0^{n_t},\ \ \ \vec{\eta}_i\ )_{\mathbb{B}_t}, \end{array}\right\} \tag{24}$$

$$\boldsymbol{c}_{\ell+1} := (\ s_{\ell+1}\vec{e}_1 + \theta_{\ell+1}(-\mathsf{H}^{\lambda,\mathsf{D}}_{\mathsf{hk}}(m' \,||\, \mathbb{S}'),\ 1),\ \boxed{\vec{w}_{\ell+1}},\ 0^2,\ \eta_{\ell+1}\ )_{\mathbb{B}_{d+1}}, \tag{25}$$

where $w_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $\vec{w}_i \xleftarrow{\mathsf{U}} \mathbb{F}_q^{n_t}$ $(i = 1, \ldots, \ell)$, $\vec{w}_{\ell+1} \xleftarrow{\mathsf{U}} \mathbb{F}_q^2$, and all the other variables are generated as in Game 0'.

Game 2 : Same as Game 1 except that the verification text $(\boldsymbol{c}_0, \ldots, \boldsymbol{c}_{\ell+1})$ for $(m', \mathbb{S}')$ with $\mathbb{S}' := (M, \rho)$ generated in Ver for verifying the output of the adversary are:

$$\left.\begin{array}{l} \boldsymbol{c}_0 := (\ -s_0 - s_{\ell+1},\ \boxed{-s'_0},\ 0,\ \eta_0\ )_{\mathbb{B}_0}, \\ \text{for } 1 \le i \le \ell, \\ \quad \text{if } \rho(i) = (t, \vec{v}_i) \ \wedge t \in \mathcal{T}_{\mathsf{good}}, \\ \qquad \boldsymbol{c}_i := (\ \ s_i \vec{e}_{t,1} + \theta_i \vec{v}_i,\ \ \boxed{s'_i \vec{e}_{t,1} + \theta'_i \vec{v}_i},\ 0,\ \ 0^{n_t},\ \ \vec{\eta}_i\ )_{\mathbb{B}_t}, \\ \quad \text{if } \rho(i) = \neg(t, \vec{v}_i) \ \wedge t \in \mathcal{T}_{\mathsf{good}},\ \ \boldsymbol{c}_i := (\ \ s_i \vec{v}_i,\ \ \boxed{s'_i \vec{v}_i},\ 0,\ \ 0^{n_t},\ \ \vec{\eta}_i\ )_{\mathbb{B}_t}, \end{array}\right\} \tag{26}$$

where $\vec{f}' \xleftarrow{\mathsf{U}} \{\vec{f}' \in {\mathbb{F}_q}^r \mid M_i \cdot \vec{f}' = 0 \text{ for } \forall i \text{ s.t. } \widetilde{\rho}(i) \in \mathcal{T}_{\mathsf{bad}}\}$, $(\vec{s}')^{\mathrm{T}} := (s'_1, \ldots, s'_\ell)^{\mathrm{T}} := M \cdot (\vec{f}')^{\mathrm{T}}$, $s'_0 := \vec{1} \cdot (\vec{f}')^{\mathrm{T}}$, $\theta'_i \xleftarrow{\mathsf{U}} \mathbb{F}_q$ $(i = 1, \ldots, \ell)$, and all the other variables are generated as in Game 1.

Game 3-h-1 $(h = 1, \ldots, \nu_1)$: Game 3-0-1 is Game 2. Game 3-h-1 is the same as Game 3-$(h-1)$-7 except that $\boldsymbol{k}^*_t$ for $t = 0$ and $(t, \vec{x}_t) \in \Gamma \ \wedge t \in \mathcal{T}_{\mathsf{good}}$ of the reply to $\mathsf{UserReg}$ and $\mathsf{AttrGen}$ reveal queries for the h-th user identity uid is:

$$\boldsymbol{k}^*_0 := (\ \delta,\ \boxed{\delta'},\ \varphi_0,\ 0\ )_{\mathbb{B}^*_0}, \tag{27}$$

$$\boldsymbol{k}^*_t := (\ \ \delta\vec{x}_t,\ \ \boxed{\delta'\vec{x}_t},\ 0,\ \ \vec{\varphi}_t,\ \ 0^2\ )_{\mathbb{B}^*_t} \ \text{ for } (t, \vec{x}_t) \in \Gamma \ \wedge t \in \mathcal{T}_{\mathsf{good}}, \tag{28}$$

where $\delta' \xleftarrow{\mathsf{U}} \mathbb{F}_q$, and all the other variables are generated as in Game 3-$(h-1)$-7.

Game 3-h-2 $(h = 1, \ldots, \nu_1)$: Game 3-h-2 is the same as Game 3-h-1 except that $\boldsymbol{k}^*_t$ for $(t, \vec{x}_t) \in \Gamma \ \wedge t \in \mathcal{T}_{\mathsf{good}}$ of the reply to $\mathsf{AttrGen}$ reveal queries for the h-th uid, and $(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_\ell)$ of the verification text for $(m', \mathbb{S}')$ with $\mathbb{S}' := (M, \rho)$ generated in Ver for

verifying the output of the adversary are:

$$\boldsymbol{k}_t^* := (\quad \delta\vec{x}_t, \quad \boxed{0^{n_t},\ \delta'}, , \quad \vec{\varphi}_t, \quad 0^2\ )_{\mathbb{B}_t^*} \ \text{ for } (t,\vec{x}_t) \in \Gamma \ \boxed{\wedge\ t \in \mathcal{T}_{\mathsf{good}}}, \tag{29}$$

$$\left.\begin{array}{l} \text{for } 1 \le i \le \ell, \\ \quad \text{if } \rho(i) = (t,\vec{v}_i) \ \wedge\ (t,\vec{x}_t) \in \Gamma \ \boxed{\wedge\ t \in \mathcal{T}_{\mathsf{good}}}, \\ \qquad \boldsymbol{c}_i := (\ s_i\vec{e}_{t,1} + \theta_i\vec{v}_i,\ \boxed{(s_i'\vec{e}_{t,1} + \theta_i'\vec{v}_i,\ 0)\cdot Z_t},\ 0^{n_t},\ \vec{\eta}_i\ )_{\mathbb{B}_t}, \\ \quad \text{if } \rho(i) = \neg(t,\vec{v}_i) \ \wedge\ (t,\vec{x}_t) \in \Gamma \ \boxed{\wedge\ t \in \mathcal{T}_{\mathsf{good}}}, \\ \qquad \boldsymbol{c}_i := (\ s_i\vec{v}_i,\ \boxed{(s_i'\vec{v}_i,\ 0)\cdot Z_t},\ 0^{n_t},\ \vec{\eta}_i\ )_{\mathbb{B}_t}, \end{array}\right\} \tag{30}$$

where $\vec{e}_{t,1} := (1,0,\ldots,0) \in \mathbb{F}_q^{n_t}$, $Z_t \xleftarrow{\cup} \{Z_t \in GL(n_t+1,\mathbb{F}_q) \,|\, (0^{n_t},\ 1) = (\vec{x}_t,\ 0)\cdot(Z_t^{-1})^{\mathrm{T}}\}$, and all the other variables are generated as in Game 3-h-1. We note that the last ((n_t+1)-th) coordinate of $(s_i'\vec{e}_{t,1}+\theta_i'\vec{v}_i,\ 0)\cdot Z_t \in \mathbb{F}_q^{n_t+1}$ (resp. $(s_i'\vec{v}_i,\ 0)\cdot Z_t \in \mathbb{F}_q^{n_t+1}$) is inner-product value $(s_i'\vec{e}_{t,1} + \theta_i'\vec{v}_i)\cdot\vec{x}_t = s_i' + \theta_i'\vec{v}_i\cdot\vec{x}_t$ (resp. $s_i'\vec{v}_i\cdot\vec{x}_t$).

Game 3-h-3 ($h = 1,\ldots,\nu_1$) : Game 3-h-3 is the same as Game 3-h-2 except that $(\boldsymbol{c}_1,\ldots,\boldsymbol{c}_\ell)$ of the verification text for $(m',\mathbb{S}')$ with $\mathbb{S}' := (M,\rho)$ generated in Ver for verifying the output of the adversary are:

$$\left.\begin{array}{l} \text{for } 1 \le i \le \ell, \\ \quad \text{if } \rho(i) = (t,\vec{v}_i) \ \wedge\ (t,\vec{x}_t) \in \Gamma \ \boxed{\wedge\ t \in \mathcal{T}_{\mathsf{good}}}, \\ \qquad \boldsymbol{c}_i := (\ s_i\vec{e}_{t,1} + \theta_i\vec{v}_i,\ \boxed{\vec{w}_i},\ s_i' + \theta_i'\vec{v}_i\cdot\vec{x}_t,\ 0^{n_t},\ \vec{\eta}_i\ )_{\mathbb{B}_t}, \\ \quad \text{if } \rho(i) = \neg(t,\vec{v}_i) \ \wedge\ (t,\vec{x}_t) \in \Gamma \ \boxed{\wedge\ t \in \mathcal{T}_{\mathsf{good}}}, \\ \qquad \boldsymbol{c}_i := (\ s_i\vec{v}_i,\ \boxed{\vec{w}_i},\ s_i'\vec{v}_i\cdot\vec{x}_t,\ 0^{n_t},\ \vec{\eta}_i\ )_{\mathbb{B}_t}, \\ \quad \text{if } \rho(i) = (t,\vec{v}_i) \ \wedge\ (t,\vec{x}_t) \notin \Gamma \ \boxed{\wedge\ t \in \mathcal{T}_{\mathsf{good}}}, \\ \qquad \boldsymbol{c}_i := (\ s_i\vec{e}_{t,1} + \theta_i\vec{v}_i,\ \boxed{\vec{w}_i},\ 0,\ 0^{n_t},\ \vec{\eta}_i\ )_{\mathbb{B}_t}, \\ \quad \text{if } \rho(i) = \neg(t,\vec{v}_i) \ \wedge\ (t,\vec{x}_t) \notin \Gamma \ \boxed{\wedge\ t \in \mathcal{T}_{\mathsf{good}}}, \\ \qquad \boldsymbol{c}_i := (\ s_i\vec{v}_i,\ \boxed{\vec{w}_i},\ 0,\ 0^{n_t},\ \vec{\eta}_i\ )_{\mathbb{B}_t}, \end{array}\right\} \tag{31}$$

where, $\vec{w}_i \xleftarrow{\cup} \mathbb{F}_q^{n_t}$ for $i = 1,\ldots,\ell$, and all the other variables are generated as in Game 3-h-2.

Game 3-h-4 ($h = 1,\ldots,\nu_1$) : Game 3-h-4 is the same as Game 3-h-3 except that $\boldsymbol{k}_0^*$ of the reply to the h-th $\mathsf{UserReg}$ query is:

$$\boldsymbol{k}_0^* := (\ \delta,\ \boxed{r_0},\ \varphi_0,\ 0\ )_{\mathbb{B}_0^*}, \tag{32}$$

where $r_0 \xleftarrow{\cup} \mathbb{F}_q$, which is independent from $\delta' \xleftarrow{\cup} \mathbb{F}_q$ in Eq. (29), and all the other variables are generated as in Game 3-h-3.

Game 3-h-5 ($h = 1,\ldots,\nu_1$) : Game 3-h-5 is the same as Game 3-h-4 except that $(\boldsymbol{c}_1,\ldots,\boldsymbol{c}_\ell)$ of the verification text for $(m',\mathbb{S}')$ with $\mathbb{S}' := (M,\rho)$ generated in Ver for verifying the output of the adversary are given as in Eq. (30) (for $\boxed{t \in \mathcal{T}_{\mathsf{good}}}$), where $Z_t \xleftarrow{\cup} \{Z_t \in GL(n_t+1,\mathbb{F}_q) \,|\, (0^{n_t},\ 1) = (\vec{x}_t,\ 0)\cdot(Z_t^{-1})^{\mathrm{T}}\}$, and all the other variables are generated as in Game 3-h-4.

Game 3-h-6 ($h = 1, \ldots, \nu_1$) : Game 3-h-6 is the same as Game 3-h-5 except that $\boldsymbol{k}_t^*$ for $(t, \vec{x}_t) \in \Gamma \ \wedge t \in \mathcal{T}_{\mathsf{good}}$ of the reply to $\mathsf{AttrGen}$ reveal queries for the h-th uid are given as in Eq. (28) with $\delta' \xleftarrow{\mathsf{U}} \mathbb{F}_q$ (independent from $r_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q$ in Eq. (32)) and $(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_\ell)$ of the verification text for $(m', \mathbb{S}')$ with $\mathbb{S}' := (M, \rho)$ generated in Ver for verifying the output of the adversary are given as in Eq. (26).

Game 3-h-7 ($h = 1, \ldots, \nu_1$) : Game 3-h-7 is the same as Game 3-h-6 except that $\boldsymbol{k}_t^*$ for $(t, \vec{x}_t) \in \Gamma \ \wedge t \in \mathcal{T}_{\mathsf{good}}$ of the reply to $\mathsf{AttrGen}$ reveal queries for the h-th uid are given as a normal key component.

Game 4, Game 5-h ($h = 1, \ldots, \nu_2$), Game 6 are given in a similar manner to those for the (single-authority) ABS.

We obtain the inequality (22) in a similar manner to that of (single-authority) ABS using Problem 1', 2', and 3'. This completes the proof of Theorem 4. □

Problems 1', 2', 3' and the related lemmas

We show Problems 1', 2', 3' and the related lemmas below. We describe random dual orthonormal basis generator $\mathcal{G}_{\mathsf{ob}}{}'$ below, which is used as a subroutine in Problems 1' and 2'.

$$
\begin{aligned}
&\mathcal{G}_{\mathsf{ob}}{}'(1^\lambda, \vec{n}) : \mathsf{param}_{\mathbb{G}} := (q, \mathbb{G}, \mathbb{G}_T, G, e) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{bpg}}(1^\lambda),\\
&\quad N_0 := 4,\ N_t := 3n_t + 3 \text{ for } t = 1, \ldots, d,\ N_{d+1} := 7, \quad \kappa, \xi \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times,\\
&\quad \text{for } t = 0, \ldots, d+1, \quad \mathsf{param}_{\mathbb{V}_t} := (q, \mathbb{V}_t, \mathbb{G}_T, \mathbb{A}_t, e) := \mathcal{G}_{\mathsf{dpvs}}(1^\lambda, N_t, \mathsf{param}_{\mathbb{G}}),\\
&\qquad X_t := (\chi_{t,i,j})_{i,j} \xleftarrow{\mathsf{U}} GL(N_t, \mathbb{F}_q),\ (\vartheta_{t,i,j})_{i,j} := (X_t^{\mathrm{T}})^{-1},\\
&\qquad \boldsymbol{b}_{t,i} := \kappa(\chi_{t,i,1}, \ldots, \chi_{t,i,N_t})_{\mathbb{A}_t} = \kappa \textstyle\sum_{j=1}^{N_t} \chi_{t,i,j} \boldsymbol{a}_{t,j},\ \mathbb{B}_t := (\boldsymbol{b}_{t,1}, \ldots, \boldsymbol{b}_{t,N_t}),\\
&\qquad \boldsymbol{b}_{t,i}^* := \xi(\vartheta_{t,i,1}, \ldots, \vartheta_{t,i,N_t})_{\mathbb{A}_t} = \xi \textstyle\sum_{j=1}^{N_t} \vartheta_{t,i,j} \boldsymbol{a}_{t,j},\ \mathbb{B}_t^* := (\boldsymbol{b}_{t,1}^*, \ldots, \boldsymbol{b}_{t,N_t}^*),\\
&\quad G_0 := \kappa G, \quad G_1 := \xi G, \quad g_T := e(G, G)^{\kappa\xi},\\
&\quad \mathsf{param}_{\vec{n}} := (\{\mathsf{param}_{\mathbb{V}_t}\}_{t=0,\ldots,d+1}, g_T),\\
&\quad \mathsf{return} \ \ (\mathsf{param}_{\vec{n}}, \{\mathbb{B}_t, \mathbb{B}_t^*\}_{t=0,\ldots,d+1}, G_0, G_1).
\end{aligned}
$$

Definition 18 (Problem 1') *Problem 1' is to guess $\beta \in \{0,1\}$, given* $(\mathsf{param}_{\vec{n}}, \{\mathbb{B}_t, \widehat{\mathbb{B}}_t^*\}_{t=0,\ldots,d+1}, \boldsymbol{f}_{\beta,0}, \{\boldsymbol{e}_{\beta,t,i}\}_{t=1,\ldots,d;i=1,\ldots,n_t}, \boldsymbol{f}_{\beta,d+1}, \boldsymbol{f}_{d+1,2},\ G_0, G_1) \xleftarrow{\mathsf{R}} \mathcal{G}_\beta^{\mathsf{P1}'}(1^\lambda, \vec{n})$, *where*

$$
\begin{aligned}
&\mathcal{G}_\beta^{\mathsf{P1}'}(1^\lambda, \vec{n}) : \quad (\mathsf{param}_{\vec{n}}, \{\mathbb{B}_t, \mathbb{B}_t^*\}_{t=0,\ldots,d+1},\ G_0, G_1) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}{}'(1^\lambda, \vec{n}),\\
&\quad (\{\widehat{\mathbb{B}}_t^*\}_{t=0,\ldots,d+1}, \boldsymbol{f}_{\beta,0}, \{\boldsymbol{e}_{\beta,t,i}\}_{t=1,\ldots,d;i=1,\ldots,n_t}, \boldsymbol{f}_{\beta,d+1}, \boldsymbol{f}_{d+1,2}) \text{ are generated as in } \mathcal{G}_\beta^{\mathsf{P1}}.\\
&\qquad \text{using } \{\mathbb{B}_t, \mathbb{B}_t^*\}_{t=0,\ldots,d+1}, \text{and random } \omega, \tau, \gamma_0, \{z_{t,i},\ \vec{\gamma}_{t,i}\}_{t=1,\ldots,d;\ i=1,\ldots,n_t}, \gamma_{d+1}, \vec{z}_{d+1},\\
&\quad \text{return} \ \ (\mathsf{param}_{\vec{n}}, \{\mathbb{B}_t, \widehat{\mathbb{B}}_t^*\}_{t=0,\ldots,d+1}, \boldsymbol{f}_{\beta,0}, \{\boldsymbol{e}_{\beta,t,i}\}_{t=1,\ldots,d;i=1,\ldots,n_t}, \boldsymbol{f}_{\beta,d+1}, \boldsymbol{f}_{d+1,2}\ G_0, G_1).
\end{aligned}
$$

for $\beta \xleftarrow{\mathsf{U}} \{0,1\}$. For a probabilistic machine $\mathcal{B}$, the advantage of $\mathcal{B}$ for Problem 1', $\mathsf{Adv}_{\mathcal{B}}^{\mathsf{P1}'}(\lambda)$, is similarly defined as in Definition 15.

Lemma 19 *For any adversary $\mathcal{B}$, there are probabilistic machine $\mathcal{E}_i$, whose running time is essentially the same as that of $\mathcal{B}$, such that for any security parameter λ, $\mathsf{Adv}_{\mathcal{B}}^{\mathsf{P1}'}(\lambda) \leq \mathsf{Adv}_{\mathcal{E}}^{\mathsf{DLIN}}(\lambda) + 5/q$.*

Lemma 19 is proven similarly to Lemma 1. □

Definition 19 (Problem 2') *Problem 2' is to guess* $\beta \in \{0,1\}$*, given* $(\mathsf{param}_{\vec{n}}, \{\widehat{\mathbb{B}}_t, \mathbb{B}^*_t\}_{t=0,..,d}, \mathbb{B}_{d+1}, \mathbb{B}^*_{d+1}, \boldsymbol{h}^*_{\beta,0}, \boldsymbol{e}_0, \{\boldsymbol{h}^*_{\beta,t,i}, \boldsymbol{e}_{t,i}\}_{t=1,..,d;i=1,..,n_t}, \{\boldsymbol{h}^*_{d+1,i}\}_{i=1,2},\ G_0, G_1) \xleftarrow{\mathsf{R}} \mathcal{G}^{\mathsf{P2}'}_\beta(1^\lambda, \vec{n})$*, where*

$$\begin{aligned}
&\mathcal{G}^{\mathsf{P2}'}_\beta(1^\lambda, \vec{n}): \quad (\mathsf{param}_{\vec{n}}, \{\mathbb{B}_t, \mathbb{B}^*_t\}_{t=0,...,d+1},\ G_0, G_1) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}{}'(1^\lambda, \vec{n}),\\
&\quad (\{\widehat{\mathbb{B}}_t\}_{t=0,..,d}, \boldsymbol{h}^*_{\beta,0}, \boldsymbol{e}_0, \{\boldsymbol{h}^*_{\beta,t,i}, \boldsymbol{e}_{t,i}\}_{t=1,..,d;i=1,..,n_t}, \{\boldsymbol{h}^*_{d+1,i}\}_{i=1,2}) \text{ are generated as in } \mathcal{G}^{\mathsf{P2}}_\beta.\\
&\qquad \text{using } \{\mathbb{B}_t, \mathbb{B}^*_t\}_{t=0,...,d+1}, \text{and random } \sigma, \tau, \omega, \delta, \delta_0, \{\vec{\delta}_{t,i}\}_{t=1,...,d;\ i=1,...,n_t},\\
&\quad \text{return } (\mathsf{param}_{\vec{n}}, \{\widehat{\mathbb{B}}_t, \mathbb{B}^*_t\}_{t=0,..,d}, \mathbb{B}_{d+1}, \mathbb{B}^*_{d+1}, \boldsymbol{h}^*_{\beta,0}, \boldsymbol{e}_0, \{\boldsymbol{h}^*_{\beta,t,i}, \boldsymbol{e}_{t,i}\}_{t=1,..,d;i=1,..,n_t},\\
&\qquad\qquad \{\boldsymbol{h}^*_{d+1,i}\}_{i=1,2},\ G_0, G_1, \delta G_1).
\end{aligned}$$

for $\beta \xleftarrow{\mathsf{U}} \{0,1\}$*. For a probabilistic machine* $\mathcal{B}$*, the advantage of* $\mathcal{B}$ *for Problem 2',* $\mathsf{Adv}^{\mathsf{P2}'}_{\mathcal{B}}(\lambda)$*, is similarly defined as in Definition 15.*

Lemma 20 *For any adversary* $\mathcal{B}$*, there exists a probabilistic machine* $\mathcal{E}$*, whose running time is essentially the same as that of* $\mathcal{B}$*, such that for any security parameter* λ*,* $\mathsf{Adv}^{\mathsf{P2}'}_{\mathcal{B}}(\lambda) \leq \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}}(\lambda) + 5/q$*.*

Lemma 20 is proven similarly to Lemma 2. □

Definition 20 (Problem 3') *Problem 3' is to guess* $\beta \in \{0,1\}$*, given* $(\mathsf{param}_{\vec{n}}, \{\widehat{\mathbb{B}}_t, \mathbb{B}^*_t\}_{t=0,d+1}, \{\mathbb{B}_t, \mathbb{B}^*_t\}_{t=1,..,d}, \boldsymbol{h}^*_{\beta,0}, \boldsymbol{e}_0, \{\boldsymbol{h}^*_{t,i}\}_{t=1,..,d;i=1,..,n_t}, \{\boldsymbol{h}^*_{\beta,d+1,i}, \boldsymbol{e}_{d+1,i}\}_{i=1,2}, G_0, G_1) \xleftarrow{\mathsf{R}} \mathcal{G}^{\mathsf{P3}'}_\beta(1^\lambda, \vec{n})$*, where*

$$\begin{aligned}
&\mathcal{G}^{\mathsf{P3}'}_\beta(1^\lambda, \vec{n}): \quad (\mathsf{param}_{\vec{n}}, \{\mathbb{B}_t, \mathbb{B}^*_t\}_{t=0,...,d+1}) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}(1^\lambda, \vec{n}),\\
&\quad (\{\widehat{\mathbb{B}}_t\}_{t=0,d+1}, \boldsymbol{h}^*_{\beta,0}, \boldsymbol{e}_0, \{\boldsymbol{h}^*_{t,i}\}_{t=1,..,d;i=1,..,n_t}, \{\boldsymbol{h}^*_{\beta,d+1,i}, \boldsymbol{e}_{d+1,i}\}_{i=1,2}) \text{ are generated as in } \mathcal{G}^{\mathsf{P3}}_\beta.\\
&\qquad \text{using } \{\mathbb{B}_t, \mathbb{B}^*_t\}_{t=0,...,d+1}, \text{and random } \sigma, \tau, \omega, \delta, \delta_0, U_{d+1}, Z_{d+1}, \{\vec{\delta}_{d+1,i}\}_{i=1,2},\\
&\quad \text{return } (\mathsf{param}_{\vec{n}}, \{\widehat{\mathbb{B}}_t, \mathbb{B}^*_t\}_{t=0,d+1}, \{\mathbb{B}_t, \mathbb{B}^*_t\}_{t=1,..,d},\\
&\qquad\qquad \boldsymbol{h}^*_{\beta,0}, \boldsymbol{e}_0, \{\boldsymbol{h}^*_{t,i}\}_{t=1,..,d;i=1,..,n_t}, \{\boldsymbol{h}^*_{\beta,d+1,i}, \boldsymbol{e}_{d+1,i}\}_{i=1,2}, G_0, G_1),
\end{aligned}$$

for $\beta \xleftarrow{\mathsf{U}} \{0,1\}$*. For a probabilistic machine* $\mathcal{B}$*, the advantage of* $\mathcal{B}$ *for Problem 3',* $\mathsf{Adv}^{\mathsf{P3}'}_{\mathcal{B}}(\lambda)$*, is similarly defined as in Definition 15.*

Lemma 21 *For any adversary* $\mathcal{B}$*, there is a probabilistic machine* $\mathcal{E}$*, whose running time is essentially the same as that of* $\mathcal{B}$*, such that for any security parameter* λ*,* $\mathsf{Adv}^{\mathsf{P3}'}_{\mathcal{B}}(\lambda) \leq \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}}(\lambda) + 5/q$*.*

Lemma 21 is proven similarly to Lemma 3. □