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

December 22, 2011

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 efficient and practical. Its efficiency is comparable to (several times worse than) that of the most efficient (almost optimally efficient) ABS scheme the security for which is proven in the generic group model.

*An extended abstract was presented at Public Key Cryptography – PKC 2011, LNCS 6571, pages 35-52. This is the full paper.

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) [27]. 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 $x = v$.

Group signatures [10] are also in this class of signatures with another type of predicate $\boldsymbol{v}$, where $\boldsymbol{v}(\boldsymbol{x})$ holds iff predicate parameter v is the group identity (or pk_v is a public key identifying group v) and attribute x is a member identity of group v (or sk_x is a secret key of member x of group v). Due to the privacy requirement, signatures generated using sk_x release no information regarding member identity x except that x is a member of group v (Note that the concept of group signatures traditionally requires the *privacy-revocation* property as well as the above-mentioned privacy).

Recently, this class of signatures with more sophisticated predicates, *attribute-based signatures* (ABS), has been studied [12, 14, 15, 18, 19, 20, 21, 26, 30], 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 [20, 21], where predicate $\boldsymbol{v}$ is specified by a monotone span program (MSP), (M, ρ), along with a tuple of attributes $(v_1, \ldots, v_j)$, and $\boldsymbol{v}(\boldsymbol{x})$ holds iff MSP (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 (For example, $\mathsf{T}(x = v) := 1$ if $x = v$, and $\mathsf{T}(x = v) := 0$ if $x \neq v$). In general, such a predicate can be expressed using AND, OR, and Threshold gates.

An example of such monotone 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 [20, 21] 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, Prabhakaran, and Rosulek [20, 21] 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 [20] is an almost optimally efficient ABS scheme, but the security was only proven in the generic group model. The scheme shown in [21] is the only existing ABS scheme for which (full) security was proven in the standard model. It is, however, much less efficient and more complicated than the scheme in [20] since it employs the Groth-Sahai NIZK protocols [11] as building blocks.

Li, Au, Susilo, Xie and Ren [18], Li and Kim [19], and Shahandashti and Safavi-Naini [26] 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 [19] are limited to conjunction or (n, n)-threshold predicates, and those of [18, 26] are limited to (k, n)-threshold predicates.

Guo and Zeng [12] and Yang, Cao and Dong [30] presented ABS schemes for threshold predicates, but their security definitions do not include the *privacy* condition of ABS.

Khader [14, 15] 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 scheme in the *standard* model [21] is much less efficient than the (almost optimally efficient) ABS scheme in the generic group model [20].

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 $(\mathsf{T}(x_{i_1} = v_1), \ldots, \mathsf{T}(x_{i_j} = v_j))$.

  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., $(\overrightarrow{x}_1, \ldots, \overrightarrow{x}_i) \in \mathbb{F}_q^{n_1+\cdots+n_i}$), 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., $(\overrightarrow{v}_1, \ldots, \overrightarrow{v}_j) \in \mathbb{F}_q^{n_1+\cdots+n_j}$), where the component-wise inner-product relations for attribute vectors (e.g., $\{\overrightarrow{x}_{i_\iota} \cdot \overrightarrow{v}_\iota = 0$ 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 $(\mathsf{T}(\overrightarrow{x}_{i_1} \cdot \overrightarrow{v}_1 = 0), \ldots, \mathsf{T}(\overrightarrow{x}_{i_j} \cdot \overrightarrow{v}_j = 0))$ 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 scheme in [21] that employs 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 [22, 23, 16, 24], 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.

- To prove the security (especially the unforgeability), this paper employs the techniques for fully secure functional encryption (FE) [16, 24], which elaborately combine the dual system encryption methodology proposed by Waters [29] and DPVS.

  Note that although the techniques for the FE schemes in [16, 24] can be employed for ABS, it is still a challenging task to construct a fully secure ABS scheme, since the security requirements of ABS and FE differ in some important points, for example, the privacy condition is required in ABS but there is no counterpart notion in FE. This paper develops several novel techniques for our ABS scheme. See Section 4.1 for more details.

- The efficiency of the proposed ABS scheme is comparable to that of the most efficient ABS scheme in the generic group model [20], and better than that of the only existing fully secure ABS scheme in the standard model [21]. See Section 4.4 for a comparison.

- This paper also presents an extension, multi-authority (MA) setting, of the proposed ABS scheme in Section 5. 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 (see Appendix F for the proof). Our MA-ABS scheme is almost as efficient as the original SA-ABS scheme.

1.3 Related Works

- **Ring and mesh signatures:** Ring and mesh signatures [25, 5] 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 and Section 5.

- **Anonymous credentials (ACs):** Another related concept is ACs [2, 3, 6, 7, 8, 9]. 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 [21], 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.

1.4 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 $(y_1, \ldots, y_N)_{\mathbb{B}^*} := \sum_{i=1}^N y_i \boldsymbol{b}_i^*$.

2 Preliminaries

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

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.

In this paper, we concentrate on the symmetric version of dual pairing vector spaces [22, 23, 16, 24] constructed by using symmetric bilinear pairing groups given in Definition 1.

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 also has linear transformations $\phi_{i,j}$ on $\mathbb{V}$ s.t. $\phi_{i,j}(\boldsymbol{a}_j) = \boldsymbol{a}_i$ and $\phi_{i,j}(\boldsymbol{a}_k) = \boldsymbol{0}$ if $k \neq j$, which can be easily achieved by $\phi_{i,j}(\boldsymbol{x}) := (\overbrace{0, \ldots, 0}^{i-1}, G_j, \overbrace{0, \ldots, 0}^{N-i})$ where $\boldsymbol{x} := (G_1, \ldots, G_N)$. We call $\phi_{i,j}$ "canonical maps".

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 $\mathcal{G}_{\mathsf{bpg}}$.

The asymmetric version of DPVS, $(q, \mathbb{V}, \mathbb{V}^*, \mathbb{G}_T, \mathbb{A}, \mathbb{A}^*, e)$, is given in Appendix A.2. The above symmetric version is obtained by identifying $\mathbb{V} = \mathbb{V}^*$ and $\mathbb{A} = \mathbb{A}^*$ in the asymmetric version. (For another construction of DPVS using higher genus Jacobians, see [22].)

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*

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

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 $\mathsf{Adv}_{\mathcal{E}}^{\mathsf{H,CR}}(\lambda) := \Pr[(\varrho_1, \varrho_2) \in \mathsf{D}^2 \wedge \varrho_1 \neq \varrho_2 \wedge \mathsf{H}_{\mathsf{hk}}^{\lambda,\mathsf{D}}(\varrho_1) = \mathsf{H}_{\mathsf{hk}}^{\lambda,\mathsf{D}}(\varrho_2)]$, 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 $\mathcal{E}$, $\mathsf{Adv}_{\mathcal{E}}^{\mathsf{H,CR}}(\lambda)$ is negligible in λ.

3 ABS for Non-monotone Predicates

3.1 Span Programs and Non-monotone Access Structures

Definition 4 (Span Programs [1]) *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 $\overrightarrow{1} \in \mathsf{span}\langle M_\delta \rangle$, i.e., some linear combination of the rows of M_δ gives the all one vector, $\overrightarrow{1}$. (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 $\overrightarrow{0}$. 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., $(t, \overrightarrow{v})$, where $t \in \{1, \ldots, d\}$ and $\overrightarrow{v} \in \mathbb{F}_q^{n_t} \setminus \{\overrightarrow{0}\}$.*

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

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

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

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

Remark 1 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 $\overrightarrow{x} := (1, x)$ and $\overrightarrow{v} := (v, -1)$. Hence, $(t, \overrightarrow{x}_t) := (t, (1, x_t))$ and $(t, \overrightarrow{v}_i) := (t, (v_i, -1))$, but we often denote them shortly by (t, x_t) and (t, v_i). Then, $\mathbb{S} := (M, \rho)$ such that $\rho : \{1, \ldots, \ell\} \rightarrow \{(t, v), (t', v'), \ldots \neg(t, v), \neg(t', v'), \ldots\}$ $(v, v', \ldots \in \mathbb{F}_q)$, 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 $\gamma : \{1, \ldots, \ell\} \rightarrow \{0, 1\}$ 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 2 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 Appendix 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* $\overrightarrow{f}^{\mathrm{T}} := (f_1, \ldots, f_r)^{\mathrm{T}} \xleftarrow{\mathsf{U}} \mathbb{F}_q^r$. *Then,* $s_0 := \overrightarrow{1} \cdot \overrightarrow{f}^{\mathrm{T}} = \sum_{k=1}^{r} f_k$ *is the secret to be shared, and* $\overrightarrow{s}^{\mathrm{T}} := (s_1, \ldots, s_\ell)^{\mathrm{T}} := M \cdot \overrightarrow{f}^{\mathrm{T}}$ *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.,* $\overrightarrow{1} \in \mathsf{span}\langle (M_i)_{\gamma(i)=1} \rangle$ *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* $\overrightarrow{n} := (d; n_1, \ldots, n_d)$ *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,* $\Gamma := \{(t, \overrightarrow{x}_t) | \overrightarrow{x}_t \in \mathbb{F}_q^{n_t} \setminus \{\overrightarrow{0}\}, 1 \le t \le d\}$, 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 $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{\mathsf{R}} \mathsf{Setup}(1^\lambda, \overrightarrow{n})$, 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, and all signatures $\sigma \xleftarrow{\mathsf{R}} \mathsf{Sig}(\mathsf{pk}, \mathsf{sk}_\Gamma, m, \mathbb{S})$, 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* $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{\mathsf{R}} \mathsf{Setup}(1^\lambda, \overrightarrow{n})$*, all messages* m*, all attribute sets* Γ_1 *and* Γ_2*, all signing keys* $\mathsf{sk}_{\Gamma_1} \xleftarrow{\mathsf{R}} \mathsf{KeyGen}(\mathsf{pk}, \mathsf{sk}, \Gamma_1)$ *and* $\mathsf{sk}_{\Gamma_2} \xleftarrow{\mathsf{R}} \mathsf{KeyGen}(\mathsf{pk}, \mathsf{sk}, \Gamma_2)$*, 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.*

For an ABS scheme with prefect privacy, we define algorithm $\mathsf{AltSig}(\mathsf{pk}, \mathsf{sk}, m, \mathbb{S})$ with $\mathbb{S}$ and master key sk instead of Γ and sk_Γ: First, generate $\mathsf{sk}_\Gamma \xleftarrow{\mathsf{R}} \mathsf{KeyGen}(\mathsf{pk},\ \mathsf{sk},\ \Gamma)$ for arbitrary Γ which satisfies $\mathbb{S}$, then $\sigma \xleftarrow{\mathsf{R}} \mathsf{Sig}(\mathsf{pk},\ \mathsf{sk}_\Gamma,\ m,\ \mathbb{S})$. return σ.

Since the correct distribution on signatures can be perfectly simulated without taking any private information as input, 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* $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{\mathsf{R}} \mathsf{Setup}(1^\lambda, \vec{n})$ *and give* pk *to the adversary.*

2. *The adversary is given access to oracles* $\mathsf{KeyGen}(\mathsf{pk}, \mathsf{sk}, \cdot)$ *and* $\mathsf{AltSig}(\mathsf{pk}, \mathsf{sk}, \cdot, \cdot)$*.*

3. *At the end, the adversary outputs* $(m', \mathbb{S}', \sigma')$*.*

We say the adversary succeeds if $(m', \mathbb{S}')$ *was never queried to the* AltSig *oracle,* $\mathbb{S}'$ *does not accept any* Γ *queried to the* KeyGen *oracle, and* $\mathsf{Ver}(\mathsf{pk}, m', \mathbb{S}', \sigma') = 1$*.*

4 Proposed ABS Scheme

4.1 Construction Ideas

Here, we will show some basic ideas to construct the proposed ABS scheme. Our ABS scheme is constructed on a ciphertext policy (CP) functional encryption (FE) scheme [24], which is adaptively payload-hiding against chosen-plaintext attacks. The description of the CP-FE scheme is given in the full version of [24].

Roughly speaking, 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. 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 also specific in ABS. Signature $\vec{\boldsymbol{s}}^*$ in ABS may be interpreted to be a decryption key specialized to decrypt a ciphertext with access structure $\mathbb{S}$, that is delegated from secret key sk_Γ.

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, $\mathsf{H}_{\mathsf{hk}}^{\lambda,\mathsf{D}}(m \,||\, \mathbb{S})$, 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 $\mathsf{H}_{\mathsf{hk}}^{\lambda,\mathsf{D}}(m \,||\, \mathbb{S})$.

Security proofs Roughly speaking, the *adaptive*-predicate unforgeability of the ABS under the KeyGen oracle attacks can be guaranteed by the *adaptive* payload-hiding property of the CP-FE, since a forged signature implies a decryption key specified for the challenge ciphertext to break the payload-hiding. 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 Appendices D and E for more details.

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 Appendix C.

We define function $\widetilde{\rho} : \{1, \ldots, \ell\} \to \{1, \ldots, d\}$ by $\widetilde{\rho}(i) := t$ if $\rho(i) = (t, v)$ or $\rho(i) = \neg(t, v)$, where ρ is given in access structure $\mathbb{S} := (M, \rho)$. In the proposed scheme, we assume that $\widetilde{\rho}$ is injective for $\mathbb{S} := (M, \rho)$. We can relax the restriction by using the method given in Appendix F in the full version of [24].

$$
\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 := 7 \text{ for } t = 1, \ldots, d+1,\\
&\quad \text{for } t = 0, \ldots, 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}}),\\
&\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}),\\
&\quad g_T := e(G, G)^\psi, \ \ \mathsf{param}_{\vec{n}} := (\{\mathsf{param}_{\mathbb{V}_t}\}_{t=0,\ldots,d+1},\ g_T),\\
&\quad \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,7}) \ \text{ for } t = 1, \ldots, d+1,\\
&\quad \widehat{\mathbb{B}}^*_t := (\boldsymbol{b}^*_{t,1}, \boldsymbol{b}^*_{t,2}, \boldsymbol{b}^*_{t,5}, \boldsymbol{b}^*_{t,6}) \ \text{ for } t = 1, \ldots, d+1,\\
&\quad \mathsf{sk} := \boldsymbol{b}^*_{0,1}, \quad \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}).\\
&\quad \text{return } \ \mathsf{sk},\ \mathsf{pk}.\\
&\mathsf{KeyGen}(\mathsf{pk},\ \mathsf{sk},\ \ \Gamma := \{(t, x_t) \mid 1 \le t \le d\}) :\\
&\quad \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 \ \text{ for } t = 1, \ldots, d;\ \iota = 1, 2;\\
&\quad \boldsymbol{k}^*_0 := (\ \delta,\ 0,\ \varphi_0,\ 0\ )_{\mathbb{B}^*_0},\\
&\quad \boldsymbol{k}^*_t := (\delta(\ 1,\ x_t\ ),\ \ 0,\ 0,\ \ \varphi_{t,1},\ \varphi_{t,2},\ \ 0)_{\mathbb{B}^*_t} \ \text{ for } (t, 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 \le t \le d,\ (t, 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, x_t)\},\\
&\quad \text{then compute } I \text{ and } \{\alpha_i\}_{i \in I} \text{ such that } \textstyle\sum_{i \in I} \alpha_i M_i = \vec{1},\\
&\quad \text{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\qquad\qquad\qquad \vee \ \ [\rho(i) = \neg(t, v_i) \ \wedge\ (t, x_t) \in \Gamma \ \wedge\ v_i \ne x_t]\ \},\\
&\quad \xi \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times, \ \ (\beta_i) \xleftarrow{\mathsf{U}} \{(\beta_1, \ldots, \beta_\ell) \ \mid\ \textstyle\sum_{i=1}^\ell \beta_i M_i = \vec{0}\},\\
&\qquad \underline{\text{Remark}} : \text{If } \det M \ne 0, \text{ the set contains only } 0^\ell, \text{ i.e., all } \beta_i = 0 \text{ for } i = 1, \ldots, \ell.\\
&\quad \boldsymbol{s}^*_0 := \xi \boldsymbol{k}^*_0 + \boldsymbol{r}^*_0, \text{ where } \boldsymbol{r}^*_0 \xleftarrow{\mathsf{U}} \mathsf{span}\langle \boldsymbol{b}^*_{0,3} \rangle,\\
&\quad \boldsymbol{s}^*_i := \gamma_i \cdot \xi \boldsymbol{k}^*_t + \textstyle\sum_{\iota=1}^2 y_{i,\iota} \cdot \boldsymbol{b}^*_{t,\iota} + \boldsymbol{r}^*_i \quad \text{ for } 1 \le i \le \ell,\\
&\qquad \text{where } \boldsymbol{r}^*_i \xleftarrow{\mathsf{U}} \mathsf{span}\langle \boldsymbol{b}^*_{t,5}, \boldsymbol{b}^*_{t,6} \rangle, \text{ and } \gamma_i, \vec{y}_i := (y_{i,1}, y_{i,2}) \text{ are defined as}\\
&\qquad \text{if } i \in I \ \wedge\ \rho(i) = (t, v_i), \quad\ \ \gamma_i := \alpha_i, \ \ \vec{y}_i := \beta_i(1, v_i),\\
&\qquad \text{if } i \in I \ \wedge\ \rho(i) = \neg(t, v_i), \quad\ \gamma_i := \frac{\alpha_i}{v_i - x_t}, \ \ \vec{y}_i := \frac{\beta_i}{v_i - y_i}(1, y_i),
\end{aligned}
$$

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

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

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

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

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

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

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

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

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

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

for $1 \leq i \leq \ell$,

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

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

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

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

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

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 $\mathcal{E}_1, \mathcal{E}_2^+, \mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4$, whose running times are essentially the same as that of $\mathcal{A}$, such that for any security parameter λ,

$$\begin{aligned}\mathsf{Adv}^{\mathsf{ABS,UF}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}_1}(\lambda) &+ \sum_{h=0}^{\nu_1 - 1} \left(\mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}^+_{2,h}}(\lambda) + \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}_{2,h+1}}(\lambda) \right) \\ &+ \sum_{h=1}^{\nu_2} \left(\mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}_{3,h}}(\lambda) + \mathsf{Adv}^{\mathsf{H,CR}}_{\mathcal{E}_{4,h}}(\lambda) \right) + \epsilon,\end{aligned}$$

where $\mathcal{E}^+_{2,h}(\cdot) := \mathcal{E}^+_2(h, \cdot), \mathcal{E}_{2,h+1}(\cdot) := \mathcal{E}_2(h, \cdot)$ $(h = 0, \ldots, \nu_1 - 1)$, $\mathcal{E}_{3,h}(\cdot) := \mathcal{E}_3(h, \cdot), \mathcal{E}_{4,h}(\cdot) := \mathcal{E}_4(h, \cdot)$ $(h = 1, \ldots, \nu_2)$, ν_1 is the maximum number of $\mathcal{A}$'s KeyGen *queries, ν_2 is the maximum number of $\mathcal{A}$'s* AltSig *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

Table 1: Comparison with the Existing ABS Schemes

	MPR08 [20]	MPR10 [21] (Boneh-Boyen based)	MPR10 [21] (Waters based)	Proposed
Signature size (# of group elts)	$\ell + r + 2$	$51\ell + 2r + 18\lambda\ell$	$36\ell + 2r$ $+9\lambda + 12$	$7\ell + 11$
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	81
Sig. size example 2 ($\ell = 100, r = 50$, $\lambda = 128$)	152	282400	4864	711

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) [21] as well as the ABS scheme in the generic group model [20] (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 [17]). λ is the security parameter (e.g., 128).

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.

$\mathsf{UserReg}$ *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 $(1 \le t \le d)$ *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* $(\mathsf{ask}_t, \mathsf{apk}_t) \xleftarrow{\mathsf{R}} \mathsf{ASetup}(\mathsf{tpk})$ $(1 \le t \le d)$, *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.*

For a MA-ABS scheme with perfect privacy, we define algorithm $\mathsf{AltSig}(\mathsf{tpk}, \mathsf{tsk}, \{\mathsf{apk}_t, \mathsf{ask}_t\}, m, \mathbb{S})$ with $\mathbb{S}$, trustee secret key tsk and attribute-authority secret keys ask_t instead of $\Gamma, \mathsf{token}_{\mathsf{uid}}$ and $\{\mathsf{usk}_t\}_{(t,x_t) \in \Gamma}$: First, generate $\mathsf{token}_{\mathsf{uid}} \xleftarrow{\mathsf{R}} \mathsf{UserReg}(\mathsf{tpk}, \mathsf{tsk}, \mathsf{uid})$ for arbitrary uid and $\mathsf{usk}_t \xleftarrow{\mathsf{R}} \mathsf{AttrGen}(\mathsf{tpk}, t, \mathsf{ask}_t, \mathsf{token}_{\mathsf{uid}}, x_t)\}_{(t,x_t) \in \Gamma}$ for arbitrary $\Gamma := \{(t, x_t)\}$ which satisfies $\mathbb{S}$, then $\sigma \xleftarrow{\mathsf{R}} \mathsf{Sig}(\mathsf{tpk}, \mathsf{token}_{\mathsf{uid}}, \{\mathsf{apk}_t, \mathsf{usk}_t \mid (t, x_t) \in \Gamma\}, m, \mathbb{S})$. Return σ.

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* $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{MA\text{-}ABS,UF}}(\lambda)$ *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. *Run* $(\mathsf{tsk}, \mathsf{tpk}) \xleftarrow{\mathsf{R}} \mathsf{TSetup}(1^\lambda)$ *and give* tpk *to the adversary* $\mathcal{A}$. *For authorities* $t \in T$, *run* $(\mathsf{ask}_t, \mathsf{apk}_t) \xleftarrow{\mathsf{R}} \mathsf{ASetup}(\mathsf{tpk})$ *and give* $\{\mathsf{apk}_t\}_{t \in T}$ *to* $\mathcal{A}$. *Adversary* $\mathcal{A}$ *specifies a set* $\widetilde{T} \subset T$ *of corrupt attribute authorities, and gets* $\{\mathsf{ask}_t\}_{t \in \widetilde{T}}$.

2. *The adversary* $\mathcal{A}$ *is given access to oracles* UserReg, AttrGen *and* AltSig *over* $S := T \setminus \widetilde{T}$.

3. *At the end, the adversary outputs* $(m', \mathbb{S}', \sigma')$.

Let $\Gamma_{\mathsf{uid}_i} := \{(t \in S, x_t)\}$ $(i \in \{1, \ldots, \nu_1\})$ *queried to the* AttrGen *oracle with* uid_i. *We say the adversary succeeds, if* $(m', \mathbb{S}')$ *was never queried to the* AltSig *oracle,* $\mathbb{S}'$ *does not accept* Γ_{uid_i} *with any* uid_i $(i \in \{1, \ldots, \nu_1\})$ *queried to the* AttrGen *oracle,* $\mathbb{S}'$ *is specified over* S, *and* $\mathsf{Ver}(\mathsf{pk}, m', \mathbb{S}', \sigma') = 1$.

Remark 3 The model regarding *corrupted authorities* in this definition is weaker than that in [21]. Roughly, the security on this model implies that no adversary $\mathcal{A}$ can forge a signature with a predicate $\mathbb{S}'_S$ unless $\mathcal{A}$ issues key queries for Γ_S such that $\mathbb{S}'_S$ accepts Γ_S, where $\mathbb{S}'_S$ and Γ_S are a predicate and attributes over uncorrupted parties S. On the other hand, the security on the model in [21] implies that no adversary $\mathcal{A}$ can forge a signature with a predicate $\mathbb{S}'_{S\cup\widetilde{T}}$ unless $\mathcal{A}$ issues key queries for Γ_S such that, for some $\Gamma_{\widetilde{T}}$, $\mathbb{S}'_{S\cup\widetilde{T}}$ accepts $(\Gamma_S \cup \Gamma_{\widetilde{T}})$.

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 $X := (\chi_{i,j})_{i,j=1,\dots,N} \in \mathbb{F}_q^{N\times N}$ and element $\boldsymbol{v}$ in N-dimensional $\mathbb{V}$, $X(\boldsymbol{v})$ denotes $\sum_{i=1,j=1}^{N,N} \chi_{i,j}\phi_{i,j}(\boldsymbol{v})$ using canonical maps $\{\phi_{i,j}\}$ (Definition 2). Similarly, for matrix $(\vartheta_{i,j}) := (X^{-1})^{\mathrm{T}}$, $(X^{-1})^{\mathrm{T}}(\boldsymbol{v}) := \sum_{i=1,j=1}^{N,N} \vartheta_{i,j}\phi_{i,j}(\boldsymbol{v})$. It holds that $e(X(\boldsymbol{x}),\ (X^{-1})^{\mathrm{T}}(\boldsymbol{y})) = e(\boldsymbol{x},\boldsymbol{y})$ for any $\boldsymbol{x},\boldsymbol{y} \in \mathbb{V}$.

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

$$
\begin{aligned}
&\mathsf{TSetup}(1^\lambda): \quad \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, \quad (\mathsf{verk}, \mathsf{sigk}) \xleftarrow{\mathsf{R}} \mathsf{G_{SIG}}(1^\lambda) \quad N_0 := 4,\ N_{d+1} := 7, \quad \kappa, \xi \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times,\\
&\quad \text{for } t = 0, 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} := (X_t^{-1})^{\mathrm{T}},\\
&\quad\quad \boldsymbol{b}_{t,i} := \kappa(\chi_{t,i,1}, \dots, \chi_{t,i,N_t})_{\mathbb{A}_t},\ \mathbb{B}_t := (\boldsymbol{b}_{t,1}, \dots, \boldsymbol{b}_{t,N_t}),\\
&\quad\quad \boldsymbol{b}^*_{t,i} := \xi(\vartheta_{t,i,1}, \dots, \vartheta_{t,i,N_t})_{\mathbb{A}_t},\ \mathbb{B}^*_t := (\boldsymbol{b}^*_{t,1}, \dots, \boldsymbol{b}^*_{t,N_t}),\\
&\quad G_0 := \kappa G, \quad G_1 := \xi G, \quad g_T := e(G,G)^{\kappa\xi},\\
&\quad \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}),\\
&\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 \mathsf{tsk} := (\boldsymbol{b}^*_{0,1}, \mathsf{sigk}),\\
&\quad \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}),\\
&\quad \text{return} \quad (\mathsf{tsk},\ \mathsf{tpk}).\\
&\mathsf{UserReg}(\mathsf{tpk}, \mathsf{tsk}, \mathsf{uid}): \ \delta \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times, \quad \varphi_0, \varphi_{d+1,1,\iota}, \varphi_{d+1,2,\iota} \xleftarrow{\mathsf{U}} \mathbb{F}_q, \quad G_{\mathsf{uid}} := \delta G_1,\\
&\quad \boldsymbol{k}^*_0 := (\ \delta,\ 0,\ \varphi_0,\ 0\ )_{\mathbb{B}^*_0},\\
&\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 \mathsf{usk}_0 := (\boldsymbol{k}^*_0, \boldsymbol{k}^*_{d+1,1}, \boldsymbol{k}^*_{d+1,2}), \quad \sigma_{\mathsf{uid}} := \mathsf{S}(\mathsf{sigk}, (\mathsf{uid}, G_{\mathsf{uid}})),\\
&\quad \text{return} \quad \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^{7-i}) \text{ for } j=0,1; i=1,..,7,\ X_t \xleftarrow{\mathsf{U}} GL(7, \mathbb{F}_q),\\
&\quad \mathbb{B}_t := (\boldsymbol{b}_{t,i})_{i=1,\dots,7} := (X_t(\boldsymbol{u}_{0,1}), \dots, X_t(\boldsymbol{u}_{0,7})),\\
&\quad \mathbb{B}^*_t := (\boldsymbol{b}^*_{t,i})_{i=1,\dots,7} := ((X_t^{-1})^{\mathrm{T}}(\boldsymbol{u}_{1,1}), \dots, (X_t^{-1})^{\mathrm{T}}(\boldsymbol{u}_{1,7})),\\
&\quad \widehat{\mathbb{B}}_t := (\boldsymbol{b}_{t,1}, \boldsymbol{b}_{t,2}, \boldsymbol{b}_{t,7}), \quad \widehat{\mathbb{B}}^*_t := (\boldsymbol{b}^*_{t,1}, \boldsymbol{b}^*_{t,2}, \boldsymbol{b}^*_{t,5}, \boldsymbol{b}^*_{t,6}),\\
&\quad \text{return} \quad (\mathsf{ask}_t := X_t,\ \mathsf{apk}_t := (\widehat{\mathbb{B}}_t, \widehat{\mathbb{B}}^*_t)).
\end{aligned}
$$

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

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

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

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

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 $\mathsf{Ver}(\mathsf{tpk},\ \{\mathsf{apk}_t\}_{t=1,\ldots,d},\ m,\ \mathbb{S} := (M, \rho), \overrightarrow{\boldsymbol{s}}^*)$ 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 $\mathcal{E}_1, \mathcal{E}_2^+, \mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4$, whose running times are essentially the same as that of $\mathcal{A}$, such that for any security parameter λ,

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{MA\text{-}ABS,UF}}(\lambda) \le \mathsf{Adv}_{\mathcal{E}_1}^{\mathsf{DLIN}}(\lambda) + \textstyle\sum_{h=0}^{\nu_1-1}\left(\mathsf{Adv}_{\mathcal{E}_{2,h}^+}^{\mathsf{DLIN}}(\lambda) + \mathsf{Adv}_{\mathcal{E}_{2,h+1}}^{\mathsf{DLIN}}(\lambda)\right)$$
$$+ \textstyle\sum_{h=1}^{\nu_2}\left(\mathsf{Adv}_{\mathcal{E}_{3,h}}^{\mathsf{DLIN}}(\lambda) + \mathsf{Adv}_{\mathcal{E}_{4,h}}^{\mathsf{H,CR}}(\lambda)\right) + \epsilon,$$

where $\mathcal{E}_{2,h}^+(\cdot) := \mathcal{E}_2^+(h, \cdot), \mathcal{E}_{2,h+1}(\cdot) := \mathcal{E}_2(h, \cdot)$ $(h = 0, \ldots, \nu_1 - 1)$, $\mathcal{E}_{3,h}(\cdot) := \mathcal{E}_3(h, \cdot), \mathcal{E}_{4,h}(\cdot) := \mathcal{E}_4(h, \cdot)$ $(h = 1, \ldots, \nu_2)$, ν_1 is the maximum number of $\mathcal{A}$'s UserReg *queries, ν_2 is the maximum number of $\mathcal{A}$'s* AltSig *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] Beimel, A., Secure schemes for secret sharing and key distribution. PhD Thesis, Israel Institute of Technology, Technion, Haifa, Israel (1996)

[2] Belenkiy, M., Camenisch, J., Chase, M., Kohlweiss, M., Lysyanskaya, A., Shacham, H.: Randomizable proofs and delegatable anonymous credentials. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 108–125. Springer Heidelberg (2009)

[3] Belenkiy, M., Chase, M., Kohlweiss, M., Lysyanskaya, A.: P-signatures and noninteractive anonymous credentials. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 356–374. Springer Heidelberg (2008)

[4] Boneh, D., Hamburg, M.: Generalized identity based and broadcast encryption scheme. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 455–470. Springer, Heidelberg (2008)

[5] Boyen, X.: Mesh signatures. In: Naor, M. (ed.) EUROCRYPT 2007, LNCS, vol. 4515, pp. 210–227. Springer Heidelberg (2007)

[6] Camenisch, J., Groß, T.: Efficient attributes for anonymous credentials. In: CCS 2008. pp.345–356. ACM (2008)

[7] Camenisch, J., Lysyanskaya, A.: An efficient system for non-transferable anonymous credentials with optinal anonymity revocation. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 93–118. Springer Heidelberg (2001)

[8] Camenisch, J., Lysyanskaya, A.: Signature schemes and anonymous credentials from bilinear maps. In: Franklin, M.K. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 56–72. Springer Heidelberg (2004)

[9] Chaum, D.: Security without identification: Transaction systems to make big brother obsolete. In: CACM, vol. 28 (10), pp. 1030–1044. ACM (1985)

[10] Chaum, D., van Heyst, E.: Group signatures. In: Davies, D.W. (ed.) EUROCRYPT '91. LNCS, vol. 547, pp. 257–265. Springer Heidelberg (1991)

[11] Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer Heidelberg (2008)

[12] Guo, S., Zeng, Y.: Attribute-based signature scheme, In: ISA 08, pp. 509–511. IEEE (2008)

[13] Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 146–162. Springer Heidelberg (2008)

[14] Khader, D.: Attribute based group signatures, ePrint, IACR, `http://eprint.iacr.org/2007/159`.

[15] Khader, D.: Attribute based group signature with revocation. ePrint, IACR, `http://eprint.iacr.org/2007/241`

[16] Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption, In Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 62–91. Springer Heidelberg (2010)

[17] Lewko, A.B., Waters, B.: Decentralizing attribute-based encryption. In: EUROCRYPT 2011. LNCS, vol. 6632, pp. 568–588. Springer Heidelberg (2011)

[18] Li, J., Au, M.H., Susilo, W., Xie, D., Ren. K.: Attribute-based signature and its application, In: ASIACCS 2010, pp. 60–69. ACM (2010)

[19] Li, J., Kim, K.: Attribute-based ring signatures. ePrint, IACR, `http://eprint.iacr.org/2008/394`

[20] Maji, H., Prabhakaran, M., Rosulek, M.: Attribute-based signatures: Achieving attribute-privacy and collusion-resistance. ePrint, IACR, `http://eprint.iacr.org/2008/328`

[21] Maji, H., Prabhakaran, M., Rosulek, M.: Attribute-based signatures. In: CT-RSA 2011, LNCS 6558, pp. 376–392 (2011). Full version is available at `http://eprint.iacr.org/2010/595`

[22] Okamoto, T., Takashima, K.: Homomorphic encryption and signatures from vector decomposition. In: Galbraith, S.D., Paterson, K.G. (eds.) Pairing 2008. LNCS, vol. 5209, pp. 57–74. Springer Heidelberg (2008)

[23] Okamoto, T., Takashima, K.: Hierarchical predicate encryption for inner-products. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 214–231. Springer Heidelberg (2009)

[24] Okamoto, T., Takashima, K.: Fully secure functional encryption with general relations from the decisional linear assumption. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 191–208. Springer Heidelberg (2010). Full version is available at `http://eprint.iacr.org/2010/563`

[25] Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 552–565. Springer Heidelberg (2001)

[26] Shahandashti, S.F., Safavi-Naini, R.: Threshold attribute-based signatures and their application to anonymous credential systems. In: Preneel B. (ed.) AFRICACRYPT 2009. LNCS, vol. 5580, pp. 198–216. Springer Heidelberg (2009)

[27] Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO '84. LNCS, vol. 196, pp. 47–53. Springer Heidelberg (1984)

[28] Shi, E., Waters, B.: Delegating capability in predicate encryption systems. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldørsson, M.M., Ingølfsdøttir, A., Walukiewicz, I. (eds.) ICALP (2) 2008. LNCS, vol. 5126, pp. 560–578. Springer Heidelberg (2008)

[29] Waters, B.: Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer Heidelberg (2009)

[30] Yang, P., Cao, Z., Dong, X.: Fuzzy identity based signature. ePrint, IACR, `http://eprint.iacr.org/2008/002`.

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

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, $X := (\chi_{i,j}) \xleftarrow{\mathsf{U}} GL(N, \mathbb{F}_q)$, 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 $(x_1, \ldots, x_N)_{\mathbb{B}} = (\overrightarrow{x})_{\mathbb{B}} := \boldsymbol{x}$ and $(y_1, \ldots, y_N)_{\mathbb{B}^*} = (\overrightarrow{y})_{\mathbb{B}^*} := \boldsymbol{y}$, and $e(\boldsymbol{x}, \boldsymbol{y}) = e(G, G)^{\sum_{i=1}^{N} x_i y_i} = e(G, G)^{\overrightarrow{x} \cdot \overrightarrow{y}} \in \mathbb{G}_T$.

Intractable problem: One of the most natural decisional problems in this approach is the decisional subspace problem [22]. 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 $(v_1, \ldots, v_{N_1}) \xleftarrow{\mathsf{U}} \mathbb{F}_q{}^{N_1}$ 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., [4, 11]). For example, in a typical vector treatment, two vector forms of $P := (x_1 G, \ldots, x_N G)$ and $Q := (y_1 G, \ldots, y_N G)$ are set and pairing for P and Q is operated as $e(P, Q) := \prod_{i=1}^{N} e(x_i G, y_i G)$. 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 $x_i \boldsymbol{a}_i = (0, \ldots, 0, x_i G, 0, \ldots, 0)$ from $P := x_1 \boldsymbol{a}_1 + \cdots x_N \boldsymbol{a}_N = (x_1 G, \ldots, x_N G)$.

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., [13, 28]).

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 $\boldsymbol{a}_i^* := (\overbrace{0, \ldots, 0}^{i-1}, G_2, \overbrace{0, \ldots, 0}^{N-i})$ with the following operations:*

1. *[Non-degenerate bilinear pairing] The pairing on $\mathbb{V}$ and $\mathbb{V}^*$ is defined by $e(\boldsymbol{x}, \boldsymbol{y}) := \prod_{i=1}^{N} e(D_i, H_i) \in \mathbb{G}_T$ where $(D_1, \ldots, D_N) := \boldsymbol{x} \in \mathbb{V}$ and $(H_1, \ldots, H_N) := \boldsymbol{y} \in \mathbb{V}^*$. This is non-degenerate 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 $e(G_1, G_2) \neq 1 \in \mathbb{G}_T$.*

2. *[Distortion maps] Linear transformation $\phi_{i,j}$ on $\mathbb{V}$ s.t. $\phi_{i,j}(\boldsymbol{a}_j) = \boldsymbol{a}_i$ and $\phi_{i,j}(\boldsymbol{a}_k) = \boldsymbol{0}$ if $k \neq j$ can be easily achieved by $\phi_{i,j}(\boldsymbol{x}) := (\overbrace{0, \ldots, 0}^{i-1}, D_j, \overbrace{0, \ldots, 0}^{N-i})$ where $(D_1, \ldots, D_N) := \boldsymbol{x}$. Moreover, linear transformation $\phi_{i,j}^*$ on $\mathbb{V}^*$ s.t. $\phi_{i,j}^*(\boldsymbol{a}_j^*) = \boldsymbol{a}_i^*$ and $\phi_{i,j}^*(\boldsymbol{a}_k^*) = \boldsymbol{0}$ if $k \neq j$ can be easily achieved by $\phi_{i,j}^*(\boldsymbol{y}) := (\overbrace{0, \ldots, 0}^{i-1}, H_j, \overbrace{0, \ldots, 0}^{N-i})$ where $(H_1, \ldots, H_N) := \boldsymbol{y}$. We call $\phi_{i,j}$ and $\phi_{i,j}^*$ "distortion maps".*

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 $(\mathbb{V}, \mathbb{V}^*)$.

B Anonymous Credentials

The notion of anonymous credentials (ACs) [2, 3, 6, 7, 8, 9] 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 [3] 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 [6] 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.

We define function $\widetilde{\rho} : \{1, \ldots, \ell\} \to \{1, \ldots, d\}$ by $\widetilde{\rho}(i) := t$ if $\rho(i) = (t, \overrightarrow{v})$ or $\rho(i) = \neg(t, \overrightarrow{v})$, where ρ is given in access structure $\mathbb{S} := (M, \rho)$. In the proposed scheme, we assume that $\widetilde{\rho}$ is injective for $\mathbb{S} := (M, \rho)$. We can relax the restriction by using the method given in Appendix F in the full version of [24].

In the description of the scheme, we assume that an input vector, $\overrightarrow{x}_t := (x_{t,1}, \ldots, x_{t,n_t})$, is normalized such that $x_{t,1} := 1$. (If $\overrightarrow{x}_t$ is not normalized, change it to a normalized one by $(1/x_{t,1}) \cdot \overrightarrow{x}_t$, assuming that $x_{t,1}$ is non-zero). In addition, we assume that input vector $\overrightarrow{v}_i := (v_{i,1}, \ldots, v_{i,n_t})$ satisfies that $v_{i,n_t} \neq 0$. We refer to Section 1.4 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.

$$
\begin{aligned}
&\mathcal{G}_{\mathsf{ob}}(1^\lambda, \overrightarrow{n} := (d; n_1, \ldots, n_d)) : \mathsf{param}_{\mathbb{G}} := (q, \mathbb{G}, \mathbb{G}_T, G, e) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{bpg}}(1^\lambda), \quad \psi \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times, \\
&\quad n_0 := 1, \; n_{d+1} := 2, \quad N_t := 3n_t + 1 \;\; \text{for } t = 0, \ldots, d+1, \\
&\quad \text{for } t = 0, \ldots, d+1, \quad \mathsf{param}_{\mathbb{V}_t} := (q, \mathbb{V}_t, \mathbb{G}_T, \mathbb{A}_t, e) := \mathcal{G}_{\mathsf{dpvs}}(1^\lambda, N_t, \mathsf{param}_{\mathbb{G}}), \\
&\qquad X_t := (\chi_{t,i,j})_{i,j} \xleftarrow{\mathsf{U}} GL(N_t, \mathbb{F}_q), \; (\vartheta_{t,i,j})_{i,j} := \psi \cdot (X_t^{\mathrm{T}})^{-1}, \\
&\qquad \boldsymbol{b}_{t,i} := (\chi_{t,i,1}, \ldots, \chi_{t,i,N_t})_{\mathbb{A}_t} = \textstyle\sum_{j=1}^{N_t} \chi_{t,i,j} \boldsymbol{a}_{t,j}, \; \mathbb{B}_t := (\boldsymbol{b}_{t,1}, \ldots, \boldsymbol{b}_{t,N_t}), \\
&\qquad \boldsymbol{b}^*_{t,i} := (\vartheta_{t,i,1}, \ldots, \vartheta_{t,i,N_t})_{\mathbb{A}_t} = \textstyle\sum_{j=1}^{N_t} \vartheta_{t,i,j} \boldsymbol{a}_{t,j}, \; \mathbb{B}^*_t := (\boldsymbol{b}^*_{t,1}, \ldots, \boldsymbol{b}^*_{t,N_t}), \\
&\quad g_T := e(G, G)^\psi, \quad \mathsf{param}_{\overrightarrow{n}} := (\{\mathsf{param}_{\mathbb{V}_t}\}_{t=0,\ldots,d+1}, \; g_T) \\
&\quad \mathsf{return} \quad (\mathsf{param}_{\overrightarrow{n}}, \{\mathbb{B}_t, \mathbb{B}^*_t\}_{t=0,\ldots,d+1}).
\end{aligned}
$$

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

$\mathsf{Setup}(1^\lambda,\ \vec{n} := (d; n_1, \ldots, n_d))$:

$\mathsf{hk} \xleftarrow{\mathsf{R}} \mathsf{KH}_\lambda,\ n_0 := 1,\ n_{d+1} := 2,\ \ (\mathsf{param}_{\vec{n}}, \{\mathbb{B}_t, \mathbb{B}_t^*\}_{t=0,\ldots,d+1}) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}(1^\lambda, \vec{n}),$

$\widehat{\mathbb{B}}_t := (\boldsymbol{b}_{t,1}, \ldots, \boldsymbol{b}_{t,n_t}, \boldsymbol{b}_{t,3n_t+1})$ for $t = 0, \ldots, d+1$,

$\widehat{\mathbb{B}}_t^* := (\boldsymbol{b}_{t,1}^*, \ldots, \boldsymbol{b}_{t,n_t}^*, \boldsymbol{b}_{t,2n_t+1}^*, \ldots, \boldsymbol{b}_{t,3n_t}^*)$ for $t = 1, \ldots, d+1$,

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

$\mathsf{KeyGen}(\mathsf{pk},\ \mathsf{sk},\ \ \Gamma := \{(t, \vec{x}_t := (x_{t,1}, \ldots, x_{t,n_t}) \in \mathbb{F}_q{}^{n_t}) \mid 1 \le t \le d\})$:

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

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

$$\boldsymbol{k}_t^* := (\ \overbrace{\delta(x_{t,1}, \ldots, x_{t,n_t}),}^{n_t}\ \ \overbrace{0^{n_t},}^{n_t}\ \ \overbrace{\varphi_{t,1}, \ldots, \varphi_{t,n_t},}^{n_t}\ \ \overbrace{0}^{1}\ )_{\mathbb{B}_t^*}\ \text{ for } (t, \vec{x}_t) \in \Gamma,$$

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

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

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

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 $\Gamma := \{(t, \vec{x}_t)\}$,

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

and $I \subseteq\ \{i \in \{1, \ldots, \ell\} \mid\ [\rho(i) = (t, \vec{v}_i)\ \wedge\ (t, \vec{x}_t) \in \Gamma\ \wedge\ \vec{v}_i \cdot \vec{x}_t = 0]$

$\vee\ \ [\rho(i) = \neg(t, \vec{v}_i)\ \wedge\ (t, \vec{x}_t) \in \Gamma\ \wedge\ \vec{v}_i \cdot \vec{x}_t \neq 0]\ \}$,

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

$\boldsymbol{s}_0^* := \xi \boldsymbol{k}_0^* + \boldsymbol{r}_0^*$, 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 $1 \le i \le \ell$,

where $\boldsymbol{r}_i^* \xleftarrow{\mathsf{U}} \mathsf{span}\langle \boldsymbol{b}_{t,2n_t+1}^*, \ldots, \boldsymbol{b}_{t,3n_t}^* \rangle$, 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 $1 \le i \le \ell$,

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 $\theta_i, \eta_i \xleftarrow{\mathsf{U}} \mathbb{F}_q$,

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

if $\rho(i) = \neg(t, \vec{v}_i)$,

return 0 if $\boldsymbol{s}_i^* \notin \mathbb{V}_t$, else $\eta_i \xleftarrow{\mathsf{U}} \mathbb{F}_q$,

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

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

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]

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

D Proof of Theorem 1

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

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

$\mathsf{AltSig}(\mathsf{pk},\ \mathsf{sk},\ m,\ \mathbb{S})$

$\widetilde{\delta} \xleftarrow{\mathsf{U}} \mathbb{F}_q^{\times}$, $\varphi_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q$,

$(\zeta_i) \xleftarrow{\mathsf{U}} \{(\zeta_1, \ldots, \zeta_\ell) \mid \sum_{i=1}^{\ell} \zeta_i M_i = \vec{1}\}$, $\boldsymbol{s}_0^* := (\widetilde{\delta}, 0, \varphi_0, 0)_{\mathbb{B}_0^*}$,

for $i = 1, \ldots, \ell$,

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

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

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

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

Remark: Theorem 1 implies that AltSig defined above is equivalent to AltSig defined just after Definition 8, and this justifies the notations.

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

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

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

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

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

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

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

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

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

$$\overrightarrow{z}_0 := (\xi\delta), \quad \overrightarrow{z}_{\ell+1} := \xi\delta(1, \mathsf{H}^{\lambda,\mathsf{D}}_{\mathsf{hk}}(m \,||\, \mathbb{S})),$$
$$\text{for } 1 \le i \le \ell,$$
$$\overrightarrow{z}_i \xleftarrow{\mathsf{U}} \{\overrightarrow{z}_i \mid \overrightarrow{z}_i \cdot \overrightarrow{v}_i = 0 \ \wedge \ z_{i,1} = \xi\delta\alpha'_i + \beta_i\} \quad \text{if } \rho(i) = (t, \overrightarrow{v}_i),$$
$$\overrightarrow{z}_i \xleftarrow{\mathsf{U}} \{\overrightarrow{z}_i \mid \overrightarrow{z}_i \cdot \overrightarrow{v}_i = \xi\delta\alpha'_i + \beta_i\} \quad \text{if } \rho(i) = \neg(t, \overrightarrow{v}_i),$$

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

$$\boldsymbol{s}^*_i := (z_{i,1}, \ldots, z_{i,n_t}, 0^{n_t}, \sigma_{i,1}, \ldots, \sigma_{i,n_t}, 0)_{\mathbb{B}^*_t} \quad (i = 0, \ldots, \ell+1), \quad \text{where}$$
$$\overrightarrow{z}_0 := (\widetilde{\delta}), \quad \overrightarrow{z}_{\ell+1} := \widetilde{\delta}(1, \mathsf{H}^{\lambda,\mathsf{D}}_{\mathsf{hk}}(m \,||\, \mathbb{S})),$$
$$\text{for } 1 \le i \le \ell,$$
$$\overrightarrow{z}_i \xleftarrow{\mathsf{U}} \{\overrightarrow{z}_i \mid \overrightarrow{z}_i \cdot \overrightarrow{v}_i = 0 \ \wedge \ z_{i,1} = \widetilde{\delta}\zeta_i\} \quad \text{if } \rho(i) = (t, \overrightarrow{v}_i),$$
$$\overrightarrow{z}_i \xleftarrow{\mathsf{U}} \{\overrightarrow{z}_i \mid \overrightarrow{z}_i \cdot \overrightarrow{v}_i = \widetilde{\delta}\zeta_i\} \quad \text{if } \rho(i) = \neg(t, \overrightarrow{v}_i),$$

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

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

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 *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 $\mathcal{E}_1, \mathcal{E}_2^+, \mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4$, whose running times are essentially the same as that of $\mathcal{A}$, such that for any security parameter λ,

$$\mathsf{Adv}^{\mathsf{ABS,UF}}_{\mathcal{A}}(\lambda) \le \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}_1}(\lambda) + \textstyle\sum_{h=0}^{\nu_1-1} \left(\mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}^+_{2,h}}(\lambda) + \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}_{2,h+1}}(\lambda)\right)$$
$$+ \textstyle\sum_{h=1}^{\nu_2} \left(\mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}_{3,h}}(\lambda) + \mathsf{Adv}^{\mathsf{H,CR}}_{\mathcal{E}_{4,h}}(\lambda)\right) + \epsilon,$$

where $\mathcal{E}^+_{2,h}(\cdot) := \mathcal{E}^+_2(h, \cdot), \mathcal{E}_{2,h+1}(\cdot) := \mathcal{E}_2(h, \cdot)$ $(h = 0, \ldots, \nu_1 - 1)$, $\mathcal{E}_{3,h}(\cdot) := \mathcal{E}_3(h, \cdot), \mathcal{E}_{4,h}(\cdot) := \mathcal{E}_4(h, \cdot)$ $(h = 1, \ldots, \nu_2)$, ν_1 is the maximum number of $\mathcal{A}$'s KeyGen *queries, ν_2 is the maximum number of $\mathcal{A}$'s* AltSig *queries, and $\epsilon := ((2d + 16)\nu_1 + 8\nu_2 + 2d + 11)/q$.*

E.1 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 [29], at the top level of strategy of the unforgeability proof.

In the methodology, verification texts (ciphertexts), secret keys and signatures have two forms, *normal* and *semi-functional*. In our proof, we also introduce another form, *pre-semi-functional* for verification texts and secret keys. The real system uses only normal verification texts, normal secret keys and normal signatures, and semi-functional/pre-semi-functional 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 4. In Game 1, the verification text is changed to semi-functional. When at most ν_1 secret key (KeyGen) queries are issued by an adversary, there are $2\nu_1$ game changes from Game 1 (Game 2-0), Game 2-0^+, Game 2-1 through Game 2-$(\nu_1-1)^+$, Game 2-ν_1. When at most ν_2 signing (AltSig) queries are issued by an adversary, there are ν_2 game changes from Game 2-ν_1 (Game 3-0), Game 3-1 through Game 3-ν_2. The final game, Game 4, is changed from Game 3-ν_2. Since $\boldsymbol{c}_0$ in the verification text is uniformly randomized in Game 4, the probability that any signature output by an adversary is correctly verified by using the randomized verification text is negligible in Game 4. As usual, we prove that the advantage gaps between neighboring games are negligible.

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, and is expressed by Eqs. (2)–(3). Similarly, a *normal* verification text $\overrightarrow{\boldsymbol{c}}_{\mathbb{S}}^{\mathsf{norm}} := (\boldsymbol{c}_0, \ldots, \boldsymbol{c}_{\ell+1})$ (with access structure $\mathbb{S}$) is Eqs. (7)–(9), and a *normal* signature $\overrightarrow{\boldsymbol{s}}^{*\ \mathsf{norm}}$, is Eqs. (4)–(6).

A *semi-functional* secret key, $\mathsf{sk}_\Gamma^{*\ \mathsf{semi}}$, is Eqs. (15),(3), and a *semi-functional* verification text, $\overrightarrow{\boldsymbol{c}}_{\mathbb{S}}^{\mathsf{semi}}$, is Eqs. (10)-(12). A *pre-semi-functional* secret key, $\mathsf{sk}_\Gamma^{*\ \mathsf{pre\text{-}semi}}$, and *pre-semi-functional* verification text, $\overrightarrow{\boldsymbol{c}}_{\mathbb{S}}^{\mathsf{pre\text{-}semi}}$, are Eqs. (13),(3) and Eqs. (10),(14),(12). A *semi-functional* signature, $\overrightarrow{\boldsymbol{s}}^{*\ \mathsf{semi}}$, is Eqs. (16), (5).

In Game 2-h, the first h keys are semi-functional while the remaining keys are normal, the verification text is semi-functional, and the signatures are normal. In Game 2-h^+, the first h keys are semi-functional and the $(h+1)$-th key is *pre-semi-functional* while the remaining keys are normal, the verification text is *pre-semi-functional*, and the signatures are normal. In Game 3-h, the first h signatures are semi-functional while the remaining signatures are normal, and all keys and the verification text are semi-functional.

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 $\beta \xleftarrow{\mathsf{U}} \{0,1\}$ 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 5). 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 2-h and 2-h^+ is similarly shown to be bounded by the advantage of Problem 2 (i.e., of the DLIN assumption) with ignoring a negligible factor (Lemmas 6 and 2). Here, we introduce *special form of pre-semi-functional* keys and verification texts, $\mathsf{sk}_\Gamma^{*\ \mathsf{spec.pre\text{-}semi}}$, and $\overrightarrow{\boldsymbol{c}}_{\mathbb{S}}^{\mathsf{spec.pre\text{-}semi}}$, such that they are equivalent to pre-semi-functional keys and verification texts except that $w_0 r_0 = a_0 := \sum_{k=1}^{r} g_k$ and $r_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q$ (note that $r_0, w_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q$ for pre-semi-functional keys and verification texts). The special form of pre-semi-functional keys and verification texts can be simulated by using Problem 2 with $\beta = 1$. From the definition, $\mathsf{sk}_\Gamma^{*\ \mathsf{spec.pre\text{-}semi}}$ can decrypt $\overrightarrow{\boldsymbol{c}}_{\mathbb{S}}^{\mathsf{spec.pre\text{-}semi}}$ for any Γ with $\mathbb{S}$ accepts Γ (i.e., it is hard for simulator $\mathcal{B}_2^+$ to tell $(\mathsf{sk}_\Gamma^{*\ \mathsf{spec.pre\text{-}semi}}, \overrightarrow{\boldsymbol{c}}_{\mathbb{S}}^{\mathsf{spec.pre\text{-}semi}})$ for Game 2-h^+ from $(\mathsf{sk}_\Gamma^{*\ \mathsf{norm}}, \overrightarrow{\boldsymbol{c}}_{\mathbb{S}}^{\mathsf{semi}})$ for Game 2-h under the assumption of Problem 2). In addition, a_0 is independently distributed from the other variables when $\mathbb{S}$ does not accept Γ (shown in Proof of Claim 1 by using Lemma 4). That is, the joint distribution of $\mathsf{sk}_\Gamma^{*\ \mathsf{pre\text{-}semi}}$ and $\overrightarrow{\boldsymbol{c}}_{\mathbb{S}}^{\mathsf{pre\text{-}semi}}$ is equivalent to that of $\mathsf{sk}_\Gamma^{*\ \mathsf{spec.pre\text{-}semi}}$ and $\overrightarrow{\boldsymbol{c}}_{\mathbb{S}}^{\mathsf{spec.pre\text{-}semi}}$, when $\mathbb{S}$ does not accept Γ (i.e., $\mathcal{B}_2^+$'s simulation using Problem 2 with $\beta = 1$ is the same distribution as that of Game 2-h^+ for the adversary's view).

The advantage gap between Games 2-h^+ and 2-$(h+1)$ is similarly shown to be bounded by the advantage of Problem 2 (i.e., of the DLIN assumption) with ignoring a negligible factor (Lemmas 7 and 2).

The advantage gap between Games 3-$(h-1)$ and 3-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 8 and 3).

Finally we show that Game 3-ν_2 can be conceptually changed to Game 4 with a negligible error probability (Lemma 9).

E.2 Main Part of the Proof

To prove Theorem 2, we consider the following $(2\nu_1 + \nu_2 + 3)$ 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. That is, the reply to a KeyGen query for $\Gamma := \{(t, \overrightarrow{x}_t)\}$ are:

$$\left.\begin{array}{l} \boldsymbol{k}_0^* := (\delta, \boxed{0}, \varphi_0, 0)_{\mathbb{B}_0^*}, \\ \boldsymbol{k}_t^* := (\delta(x_{t,1}, \ldots, x_{t,n_t}), \boxed{0^{n_t}}, \varphi_{t,1}, \ldots, \varphi_{t,n_t}, 0)_{\mathbb{B}_t^*} \ \ \text{for } (t, \overrightarrow{x}_t) \in \Gamma, \end{array}\right\} \tag{2}$$

$$\left.\begin{array}{l} \boldsymbol{k}_{d+1,1}^* := (\delta(1,0), 0, 0, \varphi_{d+1,1,1}, \varphi_{d+1,1,2}, 0)_{\mathbb{B}_{d+1}^*}, \\ \boldsymbol{k}_{d+1,2}^* := (\delta(0,1), 0, 0, \varphi_{d+1,2,1}, \varphi_{d+1,2,2}, 0)_{\mathbb{B}_{d+1}^*}, \end{array}\right\} \tag{3}$$

where $\delta \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times$, $\varphi_0, \varphi_{t,i}, \varphi_{d+1,1,i}, \varphi_{d+1,2,i} \xleftarrow{\mathsf{U}} \mathbb{F}_q$ for $t \in T$ and $i = 1, \ldots, n_t$. The reply to an AltSig query for $(m, \mathbb{S})$ with $\mathbb{S} := (M, \rho)$ are:

$$\boldsymbol{s}_0^* := (\widetilde{\delta}, \boxed{0}, \sigma_0, 0)_{\mathbb{B}_0^*}, \tag{4}$$

$$\boldsymbol{s}_i^* := (z_{i,1}, \ldots, z_{i,n_t}, 0^{n_t}, \sigma_{i,1}, \ldots, \sigma_{i,n_t}, 0)_{\mathbb{B}_t^*} \ \ \text{for } i = 1, \ldots, \ell+1, \tag{5}$$

$$\boldsymbol{s}_{\ell+1}^* := (\widetilde{\delta}(1, \mathsf{H}_{\mathsf{hk}}^{\lambda,\mathsf{D}}(m \,||\, \mathbb{S})), \boxed{0, 0}, \sigma_{\ell+1,1}, \sigma_{\ell+1,2}, 0)_{\mathbb{B}_{d+1}^*}, \tag{6}$$

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

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:

$$\boldsymbol{c}_0 := (-s_0 - s_{\ell+1}, \boxed{0}, 0, \eta_0)_{\mathbb{B}_0}, \tag{7}$$

$$\left.\begin{array}{l} \text{for } 1 \le i \le \ell, \\ \quad \text{if } \rho(i) = (t, \overrightarrow{v}_i), \quad \boldsymbol{c}_i := (s_i + \theta_i v_{i,1}, \theta_i v_{i,2}, \ldots, \theta_i v_{i,n_t}, \boxed{0^{n_t}}, 0^{n_t}, \eta_i)_{\mathbb{B}_t}, \\ \quad \text{if } \rho(i) = \neg(t, \overrightarrow{v}_i), \quad \boldsymbol{c}_i := (s_i(v_{i,1}, \ldots, v_{i,n_t}), \boxed{0^{n_t}}, 0^{n_t}, \eta_i)_{\mathbb{B}_t}, \end{array}\right\} \tag{8}$$

$$\boldsymbol{c}_{\ell+1} := (s_{\ell+1} - \theta_{\ell+1} \cdot \mathsf{H}^{\lambda,\mathsf{D}}_{\mathsf{hk}}(m' \,||\, \mathbb{S}'),\ \theta_{\ell+1}, \boxed{0, 0}, 0, 0, \eta_{\ell+1})_{\mathbb{B}_{d+1}}, \tag{9}$$

where $\vec{f} \xleftarrow{\mathsf{R}} \mathbb{F}_q{}^r$, $\vec{s}^{\mathrm{T}} := (s_1, \ldots, s_\ell)^{\mathrm{T}} := M \cdot \vec{f}^{\mathrm{T}}$, $s_0 := \vec{1} \cdot \vec{f}^{\mathrm{T}}$, $\eta_0, \eta_i, \theta_i, s_{\ell+1} \xleftarrow{\mathsf{U}} \mathbb{F}_q$ $(i = 1, \ldots, \ell+1)$.

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:

$$\boldsymbol{c}_0 := (-s_0 - s_{\ell+1}, \boxed{w_0}, 0, \eta_0)_{\mathbb{B}_0}, \tag{10}$$

$$\left.\begin{array}{l} \text{for } 1 \le i \le \ell, \\ \ \text{if } \rho(i) = (t, \vec{v}_i),\ \boldsymbol{c}_i := (s_i + \theta_i v_{i,1}, \theta_i v_{i,2}, .., \theta_i v_{i,n_t}, \boxed{w_{i,1}, .., w_{i,n_t}}, 0^{n_t}, \eta_i)_{\mathbb{B}_t}, \\ \ \text{if } \rho(i) = \neg(t, \vec{v}_i),\ \boldsymbol{c}_i := (s_i(v_{i,1}, .., v_{i,n_t}), \boxed{\overline{w}_{i,1}, .., \overline{w}_{i,n_t}}, 0^{n_t}, \eta_i)_{\mathbb{B}_t}, \end{array}\right\} \tag{11}$$

$$\boldsymbol{c}_{\ell+1} := (s_{\ell+1} - \theta_{\ell+1} \cdot \mathsf{H}^{\lambda,\mathsf{D}}_{\mathsf{hk}}(m' \,||\, \mathbb{S}'),\ \theta_{\ell+1}, \boxed{w_{\ell+1,1}, w_{\ell+1,2}}, 0, 0, \eta_{\ell+1})_{\mathbb{B}_{d+1}}, \tag{12}$$

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

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

$$\left.\begin{array}{l} \boldsymbol{k}^*_0 := (\delta, \boxed{r_0}, \varphi_0, 0)_{\mathbb{B}^*_0}, \\ \boldsymbol{k}^*_t := (\delta(x_{t,1}, \ldots, x_{t,n_t}), \boxed{r_{t,1}, \ldots, r_{t,n_t}}, \varphi_{t,1}, \ldots, \varphi_{t,n_t}, 0)_{\mathbb{B}_t} \ \ \text{for } (t, \vec{x}_t) \in \Gamma, \end{array}\right\} \tag{13}$$

$$\left.\begin{array}{l} \text{for } 1 \le i \le \ell, \\ \ \text{if } \rho(i) = (t, \vec{v}_i),\ \boldsymbol{c}_i := (s_i + \theta_i v_{i,1}, \theta_i v_{i,2}, .., \theta_i v_{i,n_t}, \boxed{w_{i,1}, .., w_{i,n_t}}, 0^{n_t}, \eta_i)_{\mathbb{B}_t}, \\ \ \text{if } \rho(i) = \neg(t, \vec{v}_i),\ \boldsymbol{c}_i := (s_i(v_{i,1}, .., v_{i,n_t}), \boxed{\overline{w}_{i,1}, .., \overline{w}_{i,n_t}}, 0^{n_t}, \eta_i)_{\mathbb{B}_t}, \end{array}\right\} \tag{14}$$

where $r_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $\vec{g} \xleftarrow{\mathsf{U}} \mathbb{F}_q{}^r$, $\vec{a}^{\mathrm{T}} := (a_1, \ldots, a_\ell)^{\mathrm{T}} := M \cdot \vec{g}^{\mathrm{T}}$, $\tau_i \xleftarrow{\mathsf{U}} \mathbb{F}_q$ $(i = 1, \ldots, \ell)$, $Z_t \xleftarrow{\mathsf{U}} GL(n_t, \mathbb{F}_q)$, $U_t := (Z_t^{-1})^{\mathrm{T}}$ for $t = 1, \ldots, d$,

$$(w_{i,1}, \ldots, w_{i,n_t}) := (a_i + \tau_i v_{i,1}, \tau_i v_{i,2}, \ldots, \tau_i v_{i,n_t}) \cdot Z_t,$$
$$(\overline{w}_{i,1}, \ldots, \overline{w}_{i,n_t}) := a_i(v_{i,1}, \ldots, v_{i,n_t}) \cdot Z_t,$$
$$(r_{t,1}, \ldots, r_{t,n_t}) := (x_{t,1}, \ldots, x_{t,n_t}) \cdot U_t,$$

and all the other variables are generated as in Game 2-h.

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

$$\left.\begin{array}{l} \boldsymbol{k}^*_0 := (\delta, r_0, \varphi_0, 0)_{\mathbb{B}^*_0}, \\ \boldsymbol{k}^*_t := (\delta(x_{t,1}, \ldots, x_{t,n_t}), \boxed{0^{n_t}}, \varphi_{t,1}, \ldots, \varphi_{t,n_t}, 0)_{\mathbb{B}_t} \ \ \text{for } (t, \vec{x}_t) \in \Gamma, \end{array}\right\} \tag{15}$$

$$\begin{array}{l} \text{for } 1 \le i \le \ell, \\ \ \text{if } \rho(i) = (t, \vec{v}_i), \quad \boldsymbol{c}_i := (s_i + \theta_i v_{i,1}, \theta_i v_{i,2}, .., \theta_i v_{i,n_t}, \boxed{w_{i,1}, .., w_{i,n_t}}, 0^{n_t}, \eta_i)_{\mathbb{B}_t}, \\ \ \text{if } \rho(i) = \neg(t, \vec{v}_i), \quad \boldsymbol{c}_i := (s_i(v_{i,1}, .., v_{i,n_t}), \boxed{\overline{w}_{i,1}, .., \overline{w}_{i,n_t}}, 0^{n_t}, \eta_i)_{\mathbb{B}_t}, \end{array}$$

where $(w_{i,1},\ldots,w_{i,n_t}),(\overline{w}_{i,1},\ldots,\overline{w}_{i,n_t}) \xleftarrow{\mathsf{U}} \mathbb{F}_q{}^{n_t}$ for $i=1,\ldots,\ell$, and all the other variables are generated as in Game 2-h^+.

Game 3-h ($h=1,\ldots,\nu_2$) : Game 3-0 is Game 2-ν_1. Game 3-h is the same as Game 3-$(h-1)$ except that $\boldsymbol{s}_0^*, \boldsymbol{s}_{\ell+1}^*$ of the reply to the h-th AltSig query for $(m,\mathbb{S})$ are:

$$\left.\begin{array}{l} \boldsymbol{s}_0^* := (\widetilde{\delta}, \boxed{\widetilde{r}_0}, \sigma_0, 0)_{\mathbb{B}_0^*}, \\ \boldsymbol{s}_{\ell+1}^* := (\widetilde{\delta}(1, \mathsf{H}_{\mathsf{hk}}^{\lambda,\mathsf{D}}(m \,||\, \mathbb{S})), \boxed{\widetilde{r}_{\ell+1,1}, \widetilde{r}_{\ell+1,2}}, \sigma_{\ell+1,1}, \sigma_{\ell+1,2}, 0)_{\mathbb{B}_{d+1}^*}, \end{array}\right\} \quad (16)$$

where $\widetilde{r}_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $(\widetilde{r}_{\ell+1,1}, \widetilde{r}_{\ell+1,2}) \xleftarrow{\mathsf{U}} \mathbb{F}_q{}^2$, and all the other variables are generated as in Game 3-$(h-1)$.

Game 4 : Same as Game 3-ν_2 except that $\boldsymbol{c}_0$ generated in Ver for verifying the output of the adversary is:

$$\boldsymbol{c}_0 := (\boxed{\widetilde{s}_0}, w_0, 0, \eta_0)_{\mathbb{B}_0}, \quad (17)$$

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

Let $\mathsf{Adv}_{\mathcal{A}}^{(0)}(\lambda)$ be $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{ABS,UF}}(\lambda)$ in Game 0, and $\mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda), \mathsf{Adv}_{\mathcal{A}}^{(2\text{-}h^+)}(\lambda), \mathsf{Adv}_{\mathcal{A}}^{(2\text{-}h)}(\lambda), \mathsf{Adv}_{\mathcal{A}}^{(3\text{-}h)}(\lambda)$, $\mathsf{Adv}_{\mathcal{A}}^{(4)}(\lambda)$ be the advantage of $\mathcal{A}$ in Game $1, 2\text{-}h, 2\text{-}h^+, 3\text{-}h, 4$, respectively. It is obtained that $\mathsf{Adv}_{\mathcal{A}}^{(4)}(\lambda) = 1/q$ by Lemma 10.

We will show five lemmas (Lemmas 5–9) that evaluate the gaps between pairs of $\mathsf{Adv}_{\mathcal{A}}^{(0)}(\lambda)$, $\mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda), \mathsf{Adv}_{\mathcal{A}}^{(2\text{-}h)}(\lambda), \mathsf{Adv}_{\mathcal{A}}^{(2\text{-}h^+)}(\lambda), \mathsf{Adv}_{\mathcal{A}}^{(2\text{-}(h+1))}(\lambda)$ for $h = 0,\ldots,\nu_1-1$, $\mathsf{Adv}_{\mathcal{A}}^{(3\text{-}h)}(\lambda)$ for $h = 1,\ldots,\nu_2$, $\mathsf{Adv}_{\mathcal{A}}^{(4)}(\lambda)$. From these lemmas and Lemmas 1–3, we obtain $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{ABS,UF}}(\lambda) = \mathsf{Adv}_{\mathcal{A}}^{(0)}(\lambda)$ $\leq \left|\mathsf{Adv}_{\mathcal{A}}^{(0)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda)\right| + \sum_{h=0}^{\nu_1-1} \left|\mathsf{Adv}_{\mathcal{A}}^{(2\text{-}h)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(2\text{-}h^+)}(\lambda)\right| + \sum_{h=0}^{\nu_1-1} \left|\mathsf{Adv}_{\mathcal{A}}^{(2\text{-}h^+)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(2\text{-}(h+1))}(\lambda)\right| + \sum_{h=1}^{\nu_2} \left|\mathsf{Adv}_{\mathcal{A}}^{(3\text{-}(h-1))}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(3\text{-}h)}(\lambda)\right| + \left|\mathsf{Adv}_{\mathcal{A}}^{(3\text{-}\nu_2)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(4)}(\lambda)\right| + \mathsf{Adv}_{\mathcal{A}}^{(4)}(\lambda) \leq \mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{P1}}(\lambda) + \sum_{h=0}^{\nu_1-1} \mathsf{Adv}_{\mathcal{B}_{2,h}^+}^{\mathsf{P2}}(\lambda) + \sum_{h=0}^{\nu_1-1} \mathsf{Adv}_{\mathcal{B}_{2,h+1}}^{\mathsf{P2}}(\lambda) + \sum_{h=1}^{\nu_2} \left(\mathsf{Adv}_{\mathcal{B}_{3,h}}^{\mathsf{P3}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_{4,h}}^{\mathsf{H,CR}}(\lambda)\right) + (2(d+3)\nu_1 + 3\nu_2 + d + 4)/q \leq \mathsf{Adv}_{\mathcal{E}_1}^{\mathsf{DLIN}}(\lambda) + \sum_{h=0}^{\nu_1-1} \left(\mathsf{Adv}_{\mathcal{E}_{2,h}^+}^{\mathsf{DLIN}}(\lambda) + \mathsf{Adv}_{\mathcal{E}_{2,h+1}}^{\mathsf{DLIN}}(\lambda)\right) + \sum_{h=1}^{\nu_2} \left(\mathsf{Adv}_{\mathcal{E}_{3,h}}^{\mathsf{DLIN}}(\lambda) + \mathsf{Adv}_{\mathcal{E}_{4,h}}^{\mathsf{H,CR}}(\lambda)\right) + ((2d+16)\nu_1 + 8\nu_2 + 2d + 11)/q$. This completes the proof of Theorem 2. $\square$

E.3 Lemmas for Theorem 2

We will show lemmas for the proof of Theorem 2. The proofs of the Lemmas 5–10 are given in Appendix E.4.

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

$$\begin{aligned} &\mathcal{G}_\beta^{\mathsf{P1}}(1^\lambda, \vec{n}): \quad n_0 := 1, \ n_{d+1} := 2, \quad (\mathsf{param}_{\vec{n}}, \{\mathbb{B}_t, \mathbb{B}_t^*\}_{t=0,\ldots,d+1}) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}(1^\lambda, \vec{n}), \\ &\quad \widehat{\mathbb{B}}_t^* := (\boldsymbol{b}_{t,1}^*, \ldots, \boldsymbol{b}_{t,n_t}^*, \boldsymbol{b}_{t,2n_t+1}^*, \ldots, \boldsymbol{b}_{t,3n_t+1}^*) \quad \text{for } t = 0, \ldots, d+1, \\ &\quad \omega, \gamma_0, \gamma_t, w_0, w_{t,1}, \ldots, w_{t,n_t} \xleftarrow{\mathsf{U}} \mathbb{F}_q \quad \text{for } t = 1, \ldots, d+1, \\ &\quad \boldsymbol{e}_{0,0} := (\omega, 0, 0, \gamma_0)_{\mathbb{B}_0}, \quad \boldsymbol{e}_{1,0} := (\omega, w_0, 0, \gamma_0)_{\mathbb{B}_0}, \end{aligned}$$

$$
\begin{array}{llllll}
\text{for } t=1,\ldots,d+1; & & & & & \\
\boldsymbol{e}_{0,t,1} := (& \overbrace{\omega, 0^{n_t-1},}^{n_t} & \overbrace{0^{n_t},}^{n_t} & \overbrace{0^{n_t},}^{n_t} & \overbrace{\gamma_t}^{1} &)_{\mathbb{B}_t}, \\
\boldsymbol{e}_{1,t,1} := (& \omega, 0^{n_t-1}, & w_{t,1},..,w_{t,n_t}, & 0^{n_t}, & \gamma_t &)_{\mathbb{B}_t}, \\
\end{array}
$$

$\boldsymbol{e}_{t,i} := \omega \boldsymbol{b}_{t,i}$ for $i=2,\ldots,n_t$,

return $(\mathsf{param}_{\vec{n}}, \{\mathbb{B}_t, \widehat{\mathbb{B}}^*_t\}_{t=0,\ldots,d+1}, \boldsymbol{e}_{\beta,0}, \{\boldsymbol{e}_{\beta,t,1}, \boldsymbol{e}_{t,i}\}_{t=1,\ldots,d+1; i=2,\ldots,n_t})$.

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

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

Lemma 1 *For any adversary* $\mathcal{B}$*, there 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*,* $\mathsf{Adv}^{\mathsf{P1}}_{\mathcal{B}}(\lambda) \le \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}}(\lambda) + (d+7)/q$*.*

Lemma 1 is proven similarly to Lemma 1 in [24]. □

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*

$\mathcal{G}^{\mathsf{P2}}_\beta(1^\lambda, \vec{n})$: $n_0 := 1$, $n_{d+1} := 2$, $(\mathsf{param}_{\vec{n}}, \{\mathbb{B}_t, \mathbb{B}^*_t\}_{t=0,\ldots,d+1}) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}(1^\lambda, \vec{n})$,

$\widehat{\mathbb{B}}_t := (\boldsymbol{b}_{t,1}, \ldots, \boldsymbol{b}_{t,n_t}, \boldsymbol{b}_{t,2n_t+1}, \ldots, \boldsymbol{b}_{t,3n_t+1})$ for $t=0,\ldots,d$,

$u_0, \tau \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times$, $\omega, \delta, \delta_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q$,

$(z_{t,i,j})_{i,j=1,\ldots,n_t} := Z_t \xleftarrow{\mathsf{U}} GL(n_t, \mathbb{F}_q)$, $(u_{t,i,j})_{i,j=1,\ldots,n_t} := U_t := (Z_t^{-1})^{\mathrm{T}}$ for $t=1,\ldots,d$,

$\boldsymbol{h}^*_{0,0} := (\delta, 0, \delta_0, 0)_{\mathbb{B}^*_0}$, $\boldsymbol{h}^*_{1,0} := (\delta, u_0, \delta_0, 0)_{\mathbb{B}^*_0}$, $\boldsymbol{e}_0 := (\omega, \tau u_0^{-1}, 0, 0)_{\mathbb{B}_0}$,

for $t=1,\ldots,d$; $i=1,\ldots,n_t$;

$(\ w_{t,i,j}\ )_{i,j=1,\ldots,n_t} := \tau \cdot Z_t$, $\delta_{t,i,j} \xleftarrow{\mathsf{U}} \mathbb{F}_q$ for $j=1,\ldots,n_t$,

$$
\begin{array}{llllll}
\boldsymbol{h}^*_{0,t,i} := (& \overbrace{0^{i-1}, \delta, 0^{n_t-i},}^{n_t} & \overbrace{0^{n_t},}^{n_t} & \overbrace{\delta_{t,i,1},..,\delta_{t,i,n_t},}^{n_t} & \overbrace{0}^{1} &)_{\mathbb{B}^*_t}, \\
\boldsymbol{h}^*_{1,t,i} := (& 0^{i-1}, \delta, 0^{n_t-i}, & u_{t,i,1},..,u_{t,i,n_t}, & \delta_{t,i,1},..,\delta_{t,i,n_t}, & 0 &)_{\mathbb{B}^*_t}, \\
\boldsymbol{e}_{t,i} := (& 0^{i-1}, \omega, 0^{n_t-i}, & w_{t,i,1},..,w_{t,i,n_t}, & 0^{n_t}, & 0 &)_{\mathbb{B}_t}
\end{array}
$$

$\boldsymbol{h}^*_{d+1,i} := \delta \boldsymbol{b}^*_{d+1,i}$ for $i=1,2$,

return $(\mathsf{param}_{\vec{n}}, \{\widehat{\mathbb{B}}_t, \mathbb{B}^*_t\}_{t=0,..,d}, \mathbb{B}_{d+1}, \mathbb{B}^*_{d+1}, \boldsymbol{h}^*_{\beta,0}, \boldsymbol{e}_0, \{\boldsymbol{h}^*_{\beta,t,i}, \boldsymbol{e}_{t,i}\}_{t=1,..,d; i=1,..,n_t}, \{\boldsymbol{h}^*_{d+1,i}\}_{i=1,2})$.

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*,* $\mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}}(\lambda) \le \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}}(\lambda) + 5/q$*.*

Lemma 2 is proven similarly to Lemma 2 in [24]. □

Definition 17 (Problem 3) *Problem 3 is to guess* $\beta \in \{0,1\}$*, given* $(\mathsf{param}_{\vec{n}}, \{\widehat{\mathbb{B}}_t, \mathbb{B}^*_t\}_{t=0,d+1},$

$\{\mathbb{B}_t, \mathbb{B}_t^*\}_{t=1,..,d}, \boldsymbol{h}_{\beta,0}^*, \boldsymbol{e}_0, \{\boldsymbol{h}_{t,i}^*\}_{t=1,..,d; i=1,..,n_t}, \{\boldsymbol{h}_{\beta,d+1,i}^*, \boldsymbol{e}_{d+1,i}\}_{i=1,2}) \xleftarrow{\mathsf{R}} \mathcal{G}_\beta^{\mathsf{P3}}(1^\lambda, \vec{n})$, *where*

$$
\begin{aligned}
&\mathcal{G}_\beta^{\mathsf{P3}}(1^\lambda, \vec{n}): \; n_0 := 1, \; n_{d+1} := 2, \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}}_t := (\boldsymbol{b}_{t,1}, \ldots, \boldsymbol{b}_{t,n_t}, \boldsymbol{b}_{t,2n_t+1}, \ldots, \boldsymbol{b}_{t,3n_t+1}) \;\; \text{for } t = 0, d+1,\\
&\quad \tau, u_0 \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^*}, \quad \boldsymbol{h}_{1,0}^* := (\delta, u_0, \delta_0, 0)_{\mathbb{B}_0^*}, \quad \boldsymbol{e}_0 := (\omega, \tau u_0^{-1}, 0, 0)_{\mathbb{B}_0},\\
&\quad \boldsymbol{h}_{t,i}^* := \delta \boldsymbol{b}_{t,i}^* \;\; \text{for } t = 1, \ldots, d; \; i = 1, \ldots, n_t,\\
&\quad (u_{d+1,i,j}) := U_{d+1} \xleftarrow{\mathsf{U}} GL(2, \mathbb{F}_q), \quad (z_{d+1,i,j}) := Z_{d+1} := (U_{d+1}^{-1})^{\mathrm{T}} \;\; \text{for } i, j = 1, 2,\\
&\quad \text{for } i = 1, 2,\\
&\qquad \delta_{d+1,i,j} \xleftarrow{\mathsf{U}} \mathbb{F}_q \;\; \text{for } j = 1, 2,\\
&\qquad \begin{array}{lllllll}
\boldsymbol{h}_{0,d+1,i}^* := & (& 0^{i-1}, \delta, 0^{2-i}, & 0^2 & \delta_{d+1,i,1}, \delta_{d+1,i,2}, & 0 &)_{\mathbb{B}_{d+1}^*},\\
\boldsymbol{h}_{1,d+1,i}^* := & (& 0^{i-1}, \delta, 0^{2-i}, & u_{d+1,i,1}, u_{d+1,i,2}, & \delta_{d+1,i,1}, \delta_{d+1,i,2}, & 0 &)_{\mathbb{B}_{d+1}^*},\\
\boldsymbol{e}_{d+1,i} := & (& 0^{i-1}, \omega, 0^{2-i}, & \tau(z_{d+1,i,1}, z_{d+1,i,2}), & 0^2, & 0 &)_{\mathbb{B}_{d+1}},
\end{array}\\
&\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}),
\end{aligned}
$$

for $\beta \xleftarrow{\mathsf{U}} \{0,1\}$. *For a probabilistic machine* $\mathcal{B}$*, the advantage of* $\mathcal{B}$ *for Problem 3,* $\mathsf{Adv}_{\mathcal{B}}^{\mathsf{P3}}(\lambda)$*, 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*,* $\mathsf{Adv}_{\mathcal{B}}^{\mathsf{P3}}(\lambda) \leq \mathsf{Adv}_{\mathcal{E}}^{\mathsf{DLIN}}(\lambda) + 5/q$*.*

Lemma 3 is proven similarly to Lemma 2 in [24]. □

Lemma 4 (Lemma 3 in [24]) *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 $U := (Z^{-1})^{\mathrm{T}}$*.*

Lemma 5 *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*,* $|\mathsf{Adv}_{\mathcal{A}}^{(0)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda)| \leq \mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{P1}}(\lambda) + (d+2)/q$*.*

Lemma 6 *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*,* $|\mathsf{Adv}_{\mathcal{A}}^{(2\text{-}h)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(2\text{-}h^+)}(\lambda)| \leq \mathsf{Adv}_{\mathcal{B}_{2,h}^+}^{\mathsf{P2}}(\lambda) + (d+3)/q$*, where* $\mathcal{B}_{2,h}^+(\cdot) := \mathcal{B}_2^+(h, \cdot)$*.*

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*,* $|\mathsf{Adv}_{\mathcal{A}}^{(2\text{-}h^+)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(2\text{-}(h+1))}(\lambda)| \leq \mathsf{Adv}_{\mathcal{B}_{2,h+1}}^{\mathsf{P2}}(\lambda) + (d+3)/q$*, where* $\mathcal{B}_{2,h+1}(\cdot) := \mathcal{B}_2(h, \cdot)$*.*

Lemma 8 *For any adversary $\mathcal{A}$, there exist probabilistic machines $\mathcal{B}_3$ and $\mathcal{E}_4$, whose running time are essentially the same as that of $\mathcal{A}$, such that for any security parameter λ, $|\mathsf{Adv}^{(3\text{-}(h-1))}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(3\text{-}h)}_{\mathcal{A}}(\lambda)| \le \mathsf{Adv}^{\mathsf{P3}}_{\mathcal{B}_{3,h}}(\lambda) + \mathsf{Adv}^{\mathsf{H,CR}}_{\mathcal{E}_{4,h}}(\lambda) + 3/q$, where $\mathcal{B}_{3,h}(\cdot) := \mathcal{B}_3(h,\cdot)$ and $\mathcal{E}_{4,h}(\cdot) := \mathcal{E}_4(h,\cdot)$.*

Lemma 9 *For any adversary $\mathcal{A}$, $\mathsf{Adv}^{(3\text{-}\nu_2)}_{\mathcal{A}}(\lambda) \le \mathsf{Adv}^{(4)}_{\mathcal{A}}(\lambda) + 1/q$.*

Lemma 10 *For any adversary $\mathcal{A}$, $\mathsf{Adv}^{(4)}_{\mathcal{A}}(\lambda) = 1/q$.*

E.4 Proofs of Lemmas 5–10

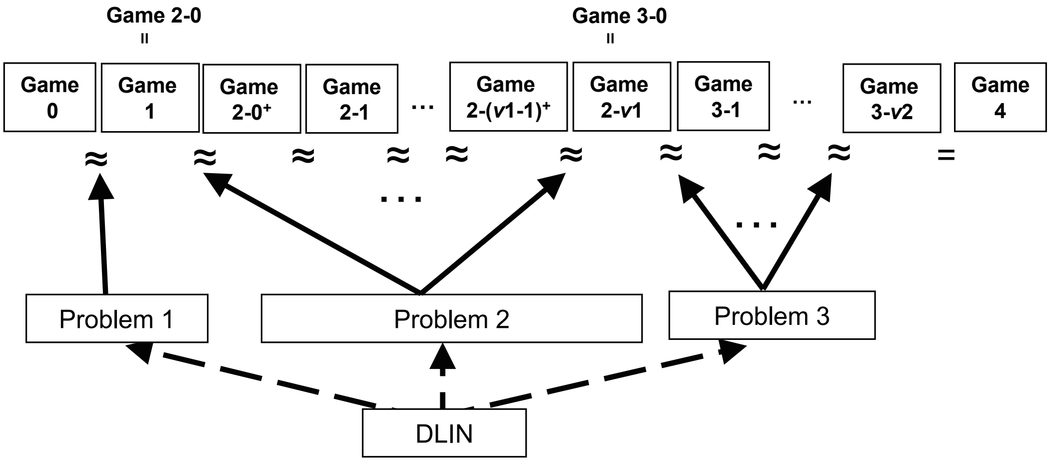

Figure 1: Structure of Reductions

Outline: 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 and 2 is reduced to that of the DLIN Problem through several intermediate steps, or intermediate problems, in [24]. They are indicated in Figure 1 by dotted arrows. The intractability of Problems 3 is also reduced to that of the DLIN Problem in a similar manner and is indicated in Figure 1 by a dotted arrow.

Problem 1 is used for evaluating the gap between advantages of adversary in Game 0 and 1 (Lemma 5). Problem 2 is used for evaluating the gaps between advantages of adversary in Game 2-h^+ and 2-h (Lemma 6) and between those in Game 2-h and 2-$(h+1)^+$ (Lemma 7). Problem 3 is used for evaluating the gap of those in Game 3-h and 3-$(h+1)$ (Lemma 8). They are indicated in Figure 1 by arrows. The gap between Games 3-ν_2 and Game 4 are evaluated without computational assumptions (Lemma 9).

Proof of Lemma 5

Lemma 5 *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, $|\mathsf{Adv}^{(0)}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(1)}_{\mathcal{A}}(\lambda)| \le \mathsf{Adv}^{\mathsf{P1}}_{\mathcal{B}_1}(\lambda) + (d+2)/q$.*

Proof. In order to prove Lemma 5, 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 $(\mathsf{param}_{\vec{n}}, \{\mathbb{B}_t, \widehat{\mathbb{B}}^*_t\}_{t=0,\ldots,d+1}, \boldsymbol{e}_{\beta,0}, \{\boldsymbol{e}_{\beta,t,1}, \boldsymbol{e}_{t,j}\}_{t=1,\ldots,d+1;j=2,\ldots,n_t})$.
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

$$\begin{aligned}
\mathbb{D}_t &:= (\boldsymbol{d}_{t,j})_{j=1,\ldots,3n_t+1} := (\boldsymbol{b}_{t,2},\ldots,\boldsymbol{b}_{t,n_t},\boldsymbol{b}_{t,1},\boldsymbol{b}_{t,n_t+1},\ldots,\boldsymbol{b}_{t,3n_t+1}) \ \text{ for } t=0,\ldots,d+1,\\
\mathbb{D}^*_t &:= (\boldsymbol{d}^*_{t,j})_{j=1,\ldots,3n_t+1} := (\boldsymbol{b}^*_{t,2},\ldots,\boldsymbol{b}^*_{t,n_t},\boldsymbol{b}^*_{t,1},\boldsymbol{b}^*_{t,n_t+1},\ldots,\boldsymbol{b}^*_{t,3n_t+1}) \ \text{ for } t=0,\ldots,d+1,\\
\widehat{\mathbb{D}}_t &:= (\boldsymbol{d}_{t,1},\ldots,\boldsymbol{d}_{t,n_t},\boldsymbol{d}_{t,3n_t+1}) \ \text{ for } t=0,\ldots,d+1,\\
\widehat{\mathbb{D}}^*_t &:= (\boldsymbol{d}^*_{t,1},\ldots,\boldsymbol{d}^*_{t,n_t},\boldsymbol{d}^*_{t,2n_t+1},\ldots,\boldsymbol{d}^*_{t,3n_t}) \ \text{ for } t=1,\ldots,d+1.
\end{aligned}$$

   $\mathcal{B}_1$ obtains $\widehat{\mathbb{D}}_t$ and $\widehat{\mathbb{D}}^*_t$ from $\mathbb{B}_t$ and $\widehat{\mathbb{B}}^*_t$ in the Problem 1 instance, and returns $\mathsf{pk} := (1^\lambda, \mathsf{hk}, \mathsf{param}_{\vec{n}}, \{\widehat{\mathbb{D}}_t\}_{t=0,..,d+1}, \{\widehat{\mathbb{D}}^*_t\}_{t=1,..,d+1}, \boldsymbol{b}^*_{0,3})$ to $\mathcal{A}$, where $\mathsf{hk} \xleftarrow{\mathsf{R}} \mathsf{KH}_\lambda$.
4. When a KeyGen (resp. AltSig) query is issued, $\mathcal{B}_1$ answers a correct secret key (resp. signature) computed by using $\{\widehat{\mathbb{B}}^*_t\}_{t=0,..,d+1}$, i.e., normal key (resp. 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:

$$\boldsymbol{c}_0 := (-s_0 - s_{\ell+1})\boldsymbol{e}_{\beta,0} + \zeta \boldsymbol{b}_{0,3}, \quad \boldsymbol{c}_i := \textstyle\sum_{j=1}^{n_t-1} c_{i,j}\boldsymbol{e}_{t,j+1} + c_{i,n_t}\boldsymbol{e}_{\beta,t,1} \ \text{ for } i=1,\ldots,\ell+1,$$

   where $\vec{f} \xleftarrow{\mathsf{R}} \mathbb{F}_q{}^r$, $\vec{s}^{\mathrm{T}} := (s_1,\ldots,s_\ell)^{\mathrm{T}} := M\cdot \vec{f}^{\mathrm{T}}$, $s_0 := \vec{1}\cdot\vec{f}^{\mathrm{T}}$, $\theta_i, s_{\ell+1} \xleftarrow{\mathsf{U}} \mathbb{F}_q$ $(i = 1,\ldots,\ell+1)$, if $\rho(i) = (t, \vec{v}_i)$, then $\vec{c}_i := (s_i + \theta_i v_{i,1}, \theta_i v_{i,2},\ldots,\theta_i v_{i,n_t})$, if $\rho(i) = \neg(t,\vec{v}_i)$, then $\vec{c}_i := s_i(v_{i,1},\ldots,v_{i,n_t})$ for $1 \le i \le \ell$, $\vec{c}_{\ell+1} := (s_{\ell+1} - \theta_{\ell+1}\cdot \mathsf{H}^{\lambda,\mathsf{D}}_{\mathsf{hk}}(m' \,||\, \mathbb{S}'),\ \theta_{\ell+1})$, and $\boldsymbol{e}_{\beta,t,1}, \boldsymbol{e}_{t,j}$ $(j=2,\ldots,n_t)$ 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 $s_0 + s_{\ell+1} = 0$ or there exists an $i \in \{1,..,\ell+1\}$ such that $c_{i,n_t} = 0$, i.e., except with probability $(\ell+2)/q \le (d+2)/q$ since $\ell \le d$. □

Proof of Lemma 6

Lemma 6 *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, $|\mathsf{Adv}^{(2\text{-}h)}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(2\text{-}h^+)}_{\mathcal{A}}(\lambda)| \le \mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}^+_{2,h}}(\lambda) + (d+3)/q$, where $\mathcal{B}^+_{2,h}(\cdot) := \mathcal{B}^+_2(h,\cdot)$.*

Proof. In order to prove Lemma 6, we construct a probabilistic machine $\mathcal{B}^+_2$ against Problem 2 by using an adversary $\mathcal{A}$ in a security game (Game 2-h or 2-h^+) as a black box as follows:

1. $\mathcal{B}^+_2$ is given an integer h and a Problem 2 instance,

   $(\mathsf{param}_{\vec{n}}, \{\widehat{\mathbb{B}}_t, \mathbb{B}^*_t\}_{t=0,..,d}, \mathbb{B}_{d+1}, \mathbb{B}^*_{d+1}, \boldsymbol{h}^*_{\beta,0}, \boldsymbol{e}_0, \{\boldsymbol{h}^*_{\beta,t,j}, \boldsymbol{e}_{t,j}\}_{t=1,..,d;j=1,..,n_t}, \{\boldsymbol{h}^*_{d+1,j}\}_{j=1,2})$.

2. $\mathcal{B}_2^+$ plays a role of the challenger in the security game against adversary $\mathcal{A}$.

3. At the first step of the game, $\mathcal{B}_2^+$ 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 2-h (and 2-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 $\widehat{\mathbb{B}}^*_t := (\boldsymbol{b}^*_{t,1}, \ldots, \boldsymbol{b}^*_{t,n_t}, \boldsymbol{b}^*_{t,2n_t+1}, \ldots, \boldsymbol{b}^*_{t,3n_t})$ from the Problem 2 instance.

4. When the ι-th key query is issued for attribute $\Gamma := \{(t, \vec{x}_t)\}$, $\mathcal{B}_2^+$ answers as follows:

   (a) When $1 \le \iota \le h$, $\mathcal{B}_2^+$ answers semi-functional key $\{\boldsymbol{k}^*_t\}_{t \in T}$ where $T := \{0, (d+1,1), (d+1,2)\} \cup \{t \mid 1 \le t \le d, (t, \vec{x}_t) \in \Gamma\}$ with Eqs. (3) and (15), that is computed by using $\{\mathbb{B}^*_t\}_{t=0,\ldots,d+1}$ of the Problem 2 instance.

   (b) When $\iota = h+1$, $\mathcal{B}_2^+$ calculates $\{\boldsymbol{k}^*_t\}_{t \in T}$ by using $\boldsymbol{h}^*_{\beta,0}, \{\boldsymbol{h}^*_{\beta,t,j}\}_{t=1,..,d; j=1,..,n_t}, \{\boldsymbol{h}^*_{d+1,j}\}_{j=1,2}$ of the Problem 2 instance as follows:

$$\boldsymbol{k}^*_0 := \boldsymbol{h}^*_{\beta,0}, \quad \boldsymbol{k}^*_t := \textstyle\sum_{j=1}^{n_t} x_{t,j} \boldsymbol{h}^*_{\beta,t,j} \quad \text{for } (t, \vec{x}_t) \in \Gamma,$$

$$\boldsymbol{k}^*_{d+1,j} := \boldsymbol{h}^*_{d+1,j} + \boldsymbol{r}^*_{d+1,j} \quad \text{where } \boldsymbol{r}^*_{d+1,j} \xleftarrow{\mathsf{U}} \mathsf{span}\langle \boldsymbol{b}^*_{d+1,5}, \boldsymbol{b}^*_{d+1,6} \rangle \quad \text{for } j = 1, 2.$$

   (c) When $\iota \ge h+2$, $\mathcal{B}_2^+$ answers normal key $\{\boldsymbol{k}^*_t\}_{t \in T}$ with Eqs. (2) and (3), that is computed by using $\{\mathbb{B}^*_t\}_{t=0,\ldots,d+1}$ of the Problem 2 instance.

5. When a AltSig query is issued, $\mathcal{B}_2^+$ answers a correct signature computed by using $\{\widehat{\mathbb{B}}^*_t\}_{t=0,..,d+1}$, i.e., normal signature.

6. When $\mathcal{B}_2^+$ receives an output $(m', \mathbb{S}', \vec{\boldsymbol{s}}'^*)$ from $\mathcal{A}$ (where $\mathbb{S}' := (M, \rho)$), $\mathcal{B}_2^+$ computes semi-functional verification text $(\boldsymbol{c}_0, \ldots, \boldsymbol{c}_{\ell+1})$ as follows: $\boldsymbol{c}_{\ell+1}$ is calculated as Eq. (12) with $\mathbb{B}_{d+1}$ from the Problem 2 instance, and using $s_{\ell+1}$ in $\boldsymbol{c}_{\ell+1}$,

$$\begin{aligned}
&\alpha_{t,l}, \widetilde{\alpha}_{k,l} \xleftarrow{\mathsf{U}} \mathbb{F}_q \quad \text{for } t = 1, \ldots, d;\ k = 1, \ldots, r;\ l = 1, 2, \\
&\widetilde{\boldsymbol{f}}_0 := \textstyle\sum_{k=1}^{r} (\widetilde{\alpha}_{k,1} \boldsymbol{e}_0 + \widetilde{\alpha}_{k,2} \boldsymbol{b}_{0,1}), \\
&\text{for } t = 1, \ldots, d;\ k = 1, \ldots, r;\ j = 1, \ldots, n_t; \\
&\quad \boldsymbol{f}_{t,j} := \alpha_{t,1} \boldsymbol{e}_{t,j} + \alpha_{t,2} \boldsymbol{b}_{t,j}, \quad \widetilde{\boldsymbol{f}}_{t,k,j} := \widetilde{\alpha}_{k,1} \boldsymbol{e}_{t,j} + \widetilde{\alpha}_{k,2} \boldsymbol{b}_{t,j}, \\
&\boldsymbol{c}_0 := -\widetilde{\boldsymbol{f}}_0 - s_{\ell+1} \boldsymbol{b}_{0,1} + \boldsymbol{q}_0, \\
&\text{For } 1 \le i \le \ell, \\
&\quad \text{if } \rho(i) = (t, \vec{v}_i), \quad \boldsymbol{c}_i := \textstyle\sum_{j=1}^{n_t} v_{i,j} \boldsymbol{f}_{t,j} + \sum_{k=1}^{r} M_{i,k} \widetilde{\boldsymbol{f}}_{t,k,n_t} + \boldsymbol{q}_i, \\
&\quad \text{if } \rho(i) = \neg(t, \vec{v}_i), \quad \boldsymbol{c}_i := \textstyle\sum_{j=1}^{n_t} v_{i,j} (\sum_{k=1}^{r} M_{i,k} \widetilde{\boldsymbol{f}}_{t,k,j}) + \boldsymbol{q}_i,
\end{aligned}$$

   where $(M_{i,k})_{i=1,\ldots,\ell; k=1,\ldots,r} := M$, $\boldsymbol{q}_0 \xleftarrow{\mathsf{U}} \mathsf{span}\langle \boldsymbol{b}_{0,4} \rangle$, and $\boldsymbol{q}_i \xleftarrow{\mathsf{U}} \mathsf{span}\langle \boldsymbol{b}_{t,3n_t+1} \rangle$. $\mathcal{B}_2^+$ verifies the signature $(m', \mathbb{S}', \vec{\boldsymbol{s}}'^*)$ using Ver with the above $\{\boldsymbol{c}_i\}_{i=0,..,\ell+1}$, and outputs $\beta' := 1$ if the verification succeeds, $\beta' := 0$ otherwise.

Remark 4 $\boldsymbol{f}_0, \boldsymbol{f}_{t,j}, \widetilde{\boldsymbol{f}}_{t,k,j}$ for $t = 1, \ldots, d; k = 1, \ldots, r; j = 1, \ldots, n_t$ calculated in the step 6 in the above simulation are expressed as:

$$\begin{aligned}
&\theta_t := \alpha_{t,1} \omega + \alpha_{t,2}, \quad \widetilde{\tau}_t := \alpha_{t,1} \tau, \\
&f_k := \widetilde{\alpha}_{k,1} \omega + \widetilde{\alpha}_{k,2}, \quad s_0 := \textstyle\sum_{k=1}^{r} f_k, \quad g_k := \widetilde{\alpha}_{k,1} \tau, \quad a_0 := \sum_{k=1}^{r} g_k, \\
&w_0 := a_0 / u_0, \quad (\varepsilon_{t,j,l})_{j,l=1,..,n_t} := \widetilde{\tau}_t \cdot Z_t, \quad (\widetilde{\varepsilon}_{t,k,j,l})_{j,l=1,..,n_t} := g_k \cdot Z_t, \\
&\boldsymbol{f}_0 = (s_0, w_0, 0, 0)_{\mathbb{B}_0},
\end{aligned}$$

$$\begin{array}{lllllll}
\boldsymbol{f}_{t,j} := & (& \overbrace{0^{j-1}, \theta_t, 0^{n_t-j},}^{n_t} & \overbrace{\varepsilon_{t,j,1}, .., \varepsilon_{t,j,n_t},}^{n_t} & \overbrace{0^{n_t},}^{n_t} & \overbrace{0}^{1} &)_{\mathbb{B}_t}, \\
\widetilde{\boldsymbol{f}}_{t,k,j} := & (& 0^{j-1}, f_k, 0^{n_t-j}, & \widetilde{\varepsilon}_{t,k,j,1}, .., \widetilde{\varepsilon}_{t,k,j,n_t}, & 0^{n_t}, & 0 &)_{\mathbb{B}_t},
\end{array}$$

where $u_0, \omega, \tau, \{Z_t\}_{t=1,\ldots,d}$ are defined in Problem 2. Note that variables $\{\theta_t, \widetilde{\tau}_t\}_{t=1,\ldots,d}, \{f_k, g_k\}_{k=1,\ldots,r}$ are independently and uniformly distributed. Therefore, $\{\boldsymbol{c}_i\}_{i=0,\ldots,\ell}$ are distributed as Eqs. (10) and (14) except $w_0 := a_0/r_0$, i.e., $w_0 r_0 = a_0$, using a_0 and $r_0 := u_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q$ in $\boldsymbol{k}_0^*$ (Eq. (13)).

Claim 1 *The distribution of the view of adversary $\mathcal{A}$ in the above-mentioned game simulated by $\mathcal{B}_2^+$ given a Problem 2 instance with $\beta \in \{0,1\}$ is the same as that in Game 2-h (resp. Game 2-h^+) if $\beta = 0$ (resp. $\beta = 1$) except with probability $(d+2)/q$ (resp. $1/q$).*

Proof. It is clear that $\mathcal{B}_2^+$'s simulation of the public-key generation (step 3) and the ι-th key query's answer for $\iota \neq h+1$ (cases (a) and (c) of step 4) is perfect, i.e., exactly the same as the Setup and the KeyGen oracle in Game 2-h and Game 2-h^+.

Therefore, to prove this lemma we will show that the joint distribution of the $(h+1)$-th key query's answer and verification text $\{\boldsymbol{c}_i\}_{i=0,..,\ell+1}$ by $\mathcal{B}_2^+$'s simulation given a Problem 2 instance with β is equivalent to that in Game 2-h (resp. Game 2-h^+), when $\beta = 0$ (resp. $\beta = 1$).

When $\beta = 0$, it is straightforward to show that they are equivalent except for that δ defined in Problem 2 is zero or there exists $i \in \{0,\ldots,\ell\}$ such that $\overrightarrow{w}_i = \overrightarrow{0}$ with $i = 0$ or $\rho(i) = (t, \overrightarrow{v}_i)$, or $\overrightarrow{\overline{w}}_i = \overrightarrow{0}$ with $\rho(i) = \neg(t, \overrightarrow{v}_i)$, where $\overrightarrow{w}_i$ and $\overrightarrow{\overline{w}}_i$ are defined in Eqs. (10) and (11) i.e., except with probability $(\ell+2)/q \leq (d+2)/q$ since $\ell \leq d$.

When $\beta = 1$, the distribution by $\mathcal{B}_2^+$'s simulation is Eqs. (3) and (13) for the key and Eqs. (10), (12), and (14) for the elements in $\mathbb{V}$, $\{\boldsymbol{c}_i\}_{i=0,\ldots,\ell+1}$, used for verifying the output of $\mathcal{A}$, where the distribution is the same as that defined in these equations except $w_0 := a_0/r_0$, i.e., $w_0 r_0 = a_0$, using $a_0 := \overrightarrow{1} \cdot \overrightarrow{g}^{\mathrm{T}}$ and $r_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q$ in $\boldsymbol{k}_0^*$ (Eq. (13)) from Remark 4. The corresponding distribution in Game 2-h^+ is Eqs. (3) and (13) and Eqs. (10), (12), and (14) where $r_0, w_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q$ as defined in the equations.

Therefore, we will show that a_0 is uniformly and independently distributed from the other variables in the joint distribution of $\mathcal{B}_2^+$'s simulation. Since $a_0 := \overrightarrow{1} \cdot \overrightarrow{g}^{\mathrm{T}}$ is only related to $(a_1,\ldots,a_\ell)^{\mathrm{T}} := M \cdot \overrightarrow{g}^{\mathrm{T}}$ and $U_t = (Z_t^{-1})^{\mathrm{T}}$ holds, a_0 is only related to $\{\overrightarrow{w}_i\}_{i=1,\ldots,\ell}, \{\overrightarrow{\overline{w}}_i\}_{i=1,\ldots,\ell}$ and $\{\overrightarrow{r}_t\}_{t=1,\ldots,d}$, where $\overrightarrow{r}_t := (r_{t,1},\ldots,r_{t,n_t}) := (x_{t,1},\ldots,x_{t,n_t}) \cdot U_t$ in Eq. (13) for $t = 1,\ldots,d$, and $\overrightarrow{w}_i := (w_{i,1},\ldots,w_{i,n_t}) := (a_i + \tau_i v_{i,1}, \tau_i v_{i,2},\ldots,\tau_i v_{i,n_t}) \cdot Z_t$ and $\overrightarrow{\overline{w}}_i := (\overline{w}_{i,1},\ldots,\overline{w}_{i,n_t}) := a_i(v_{i,1},\ldots,v_{i,n_t}) \cdot Z_t$ in Eq. (14) for $i = 1,\ldots,\ell$ with $t := \widetilde{\rho}(i)$. ($\widetilde{\rho}$ is defined at the start of Section 4.) With respect to the joint distribution of these variables, there are five cases for each $i \in \{1,\ldots,\ell\}$. Note that for any $i \in \{1,\ldots,\ell\}$, (Z_t, U_t) with $t := \widetilde{\rho}(i)$ is independent from the other variables, since $\widetilde{\rho}$ is injective:

1. $\gamma(i) = 1$ and $[\rho(i) = (t, \overrightarrow{v}_i) \ \wedge \ (t, \overrightarrow{x}_t) \in \Gamma \ \wedge \ \overrightarrow{v}_i \cdot \overrightarrow{x}_t = 0]$.

   Then, from Lemma 4, the joint distribution of $(\overrightarrow{w}_i, \overrightarrow{r}_t)$ is uniformly and independently distributed on $C_{a_i} := \{(\overrightarrow{w}, \overrightarrow{r}) | \overrightarrow{w} \cdot \overrightarrow{r} = a_i\}$ (over $Z_t \xleftarrow{\mathsf{U}} GL(n_t, \mathbb{F}_q)$).

2. $\gamma(i) = 1$ and $[\rho(i) = \neg(t, \overrightarrow{v}_i) \ \wedge \ (t, \overrightarrow{x}_t) \in \Gamma \ \wedge \ \overrightarrow{v}_i \cdot \overrightarrow{x}_t \neq 0]$.

   Then, from Lemma 4, the joint distribution of $(\overrightarrow{\overline{w}}_i, \overrightarrow{r}_t)$ is uniformly and independently distributed on $C_{(\overrightarrow{v}_i \cdot \overrightarrow{x}_t) \cdot a_i}$ (over $Z_t \xleftarrow{\mathsf{U}} GL(n_t, \mathbb{F}_q)$).

3. $\gamma(i) = 0$ and $[\rho(i) = (t, \overrightarrow{v}_i) \ \wedge \ (t, \overrightarrow{x}_t) \in \Gamma]$ (i.e., $\overrightarrow{v}_i \cdot \overrightarrow{x}_t \neq 0$).

   Then, from Lemma 4, the joint distribution of $(\overrightarrow{w}_i, \overrightarrow{r}_t)$ is uniformly and independently distributed on $C_{(\overrightarrow{v}_i \cdot \overrightarrow{x}_t) \cdot \widetilde{\tau}_t + a_i}$ (over $Z_t \xleftarrow{\mathsf{U}} GL(n_t, \mathbb{F}_q)$) where $\widetilde{\tau}_t$ is defined in Remark 4. Since $\widetilde{\tau}_t$ is uniformly and independently distributed on $\mathbb{F}_q$, the joint distribution of $(\overrightarrow{w}_i, \overrightarrow{r}_t)$ is uniformly and independently distributed over ${\mathbb{F}_q}^{2n_t}$.

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, from Lemma 4, the joint distribution of $(\vec{\vec{w}}_i, \vec{r}_t)$ is uniformly and independently distributed on C_0 (over $Z_t \xleftarrow{\mathsf{U}} GL(n_t, \mathbb{F}_q)$).

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, the distribution of $\vec{w}_i$ is uniformly and independently distributed on ${\mathbb{F}_q}^{n_t}$ (over $Z_t \xleftarrow{\mathsf{U}} GL(n_t, \mathbb{F}_q)$).

We then observe the joint distribution (or relation) of a_0, $\{\vec{w}_i\}_{i=1,..,\ell}$, $\{\vec{\vec{w}}_i\}_{i=1,..,\ell}$ and $\{\vec{r}_t\}_{t=1,..,d}$. Those in cases 3-5 are obviously independent from a_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, $a_0 := \vec{1} \cdot \vec{g}^{\mathrm{T}}$ is independent from the joint distribution of $\{a_i := M_i \cdot \vec{g}^{\mathrm{T}} \mid \gamma(i) = 1\}$ (over the random selection of $\vec{g}$), which can be given by $(\vec{w}_i, \vec{r}_t)$ in case 1 and $(\vec{\vec{w}}_i, \vec{r}_t)$ in case 2. Thus, a_0 is uniformly and independently distributed from the other variables in the joint distribution of $\mathcal{B}_2^+$'s simulation.

Therefore, the view of adversary $\mathcal{A}$ in the game simulated by $\mathcal{B}_2^+$ given a Problem 2 instance with $\beta = 1$ is the same as that in Game 2-h^+ except that δ defined in Problem 2 is zero i.e., except with probability $1/q$. □

Proof of Lemma 7

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, $|\mathsf{Adv}_{\mathcal{A}}^{(2\text{-}h^+)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(2\text{-}(h+1))}(\lambda)| \leq \mathsf{Adv}_{\mathcal{B}_{2,h+1}}^{\mathsf{P2}}(\lambda) + (d+3)/q$, where $\mathcal{B}_{2,h+1}(\cdot) := \mathcal{B}_2(h, \cdot)$.*

Proof. In order to prove Lemma 7, we construct a probabilistic machine $\mathcal{B}_2$ against Problem 2 by using an adversary $\mathcal{A}$ in a security game (Game 2-h^+ or 2-$(h+1)$) as a black box. $\mathcal{B}_2$ acts in the same way as $\mathcal{B}_2^+$ in the proof of Lemma 6 except the following two points:

1. In case (b) of step 4; $\boldsymbol{k}_0^*$ is calculated as

$$\boldsymbol{k}_0^* := \boldsymbol{h}_{\beta,0}^* + r_0' \boldsymbol{b}_{0,2}^*,$$

   where $r_0' \xleftarrow{\mathsf{U}} \mathbb{F}_q$, and $\boldsymbol{h}_{\beta,0}^*, \boldsymbol{b}_{0,2}^*$ are in the Problem 2 instance.

2. In the last step; if the verification succeeds, $\mathcal{B}_2$ outputs $\beta' := 0$. Otherwise, $\mathcal{B}_2$ outputs $\beta' := 1$.

When $\beta = 0$, it is straightforward that the distribution by $\mathcal{B}_2$'s simulation is equivalent to that in Game 2-$(h+1)$ except that δ defined in Problem 2 is zero, i.e., except with probability $1/q$. When $\beta = 1$, the distribution by $\mathcal{B}_2$'s simulation is equivalent to that in Game 2-h^+ except that δ defined in Problem 2 is zero or there exists $i \in \{0, \ldots, \ell\}$ such that $\vec{w}_i = 0$ with $i = 0$ or $\rho(i) = (t, \vec{v}_i)$, or $\vec{\vec{w}}_i = 0$ with $\rho(i) = \neg(t, \vec{v}_i)$ where $\vec{w}_i$ and $\vec{\vec{w}}_i$ are defined in Eqs. (10) and (11), i.e., except with probability $(\ell+2)/q \leq (d+2)/q$. □

Proof of Lemma 8

Lemma 8 *For any adversary $\mathcal{A}$, there exist probabilistic machines $\mathcal{B}_3$ and $\mathcal{E}_4$, whose running time are essentially the same as that of $\mathcal{A}$, such that for any security parameter λ, $|\mathsf{Adv}_{\mathcal{A}}^{(3\text{-}(h-1))}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(3\text{-}h)}(\lambda)| \leq \mathsf{Adv}_{\mathcal{B}_{3,h}}^{\mathsf{P3}}(\lambda) + \mathsf{Adv}_{\mathcal{E}_{4,h}}^{\mathsf{H,CR}}(\lambda) + 3/q$, where $\mathcal{B}_{3,h}(\cdot) := \mathcal{B}_3(h, \cdot)$ and $\mathcal{E}_{4,h}(\cdot) := \mathcal{E}_4(h, \cdot)$.*

Proof. In order to prove Lemma 8, we construct a probabilistic machine $\mathcal{B}_3$ against Problem 3 by using any adversary $\mathcal{A}$ in a security game (Game 3-$(h-1)$ or 3-h) as a black box as follows:

1. $\mathcal{B}_3$ 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. $\mathcal{B}_3$ plays a role of the challenger in the security game against adversary $\mathcal{A}$.

3. At the first step of the game, $\mathcal{B}_3$ provides $\mathcal{A}$ a public key $\mathsf{pk} := (1^\lambda, \mathsf{hk}, \mathsf{param}_{\vec{n}}, \{\widehat{\mathbb{B}}'_t\}_{t=0,\dots,d+1}, \{\widehat{\mathbb{B}}_t^*\}_{t=1,\dots,d+1}, \boldsymbol{b}_{0,3}^*)$ of Game 3-$(h-1)$ (and 3-h), where $\mathsf{hk} \xleftarrow{\mathsf{R}} \mathsf{KH}_\lambda$, $\widehat{\mathbb{B}}'_t := (\boldsymbol{b}_{t,1}, \dots, \boldsymbol{b}_{t,n_t}, \boldsymbol{b}_{t,3n_t+1})$, and $\widehat{\mathbb{B}}_t^* := (\boldsymbol{b}_{t,1}^*, \dots, \boldsymbol{b}_{t,n_t}^*, \boldsymbol{b}_{t,2n_t+1}^*, \dots, \boldsymbol{b}_{t,3n_t}^*)$, that are obtained from the Problem 3 instance.

4. When KeyGen query is issued for attribute $\Gamma := \{(t, \vec{x}_t)\}$, $\mathcal{B}_{3,h}$ answers semi-functional key $\{\boldsymbol{k}_t^*\}_{t\in T}$ where $T := \{0, (d+1,1), (d+1,2)\} \cup \{t \mid 1 \le t \le d,\ (t, \vec{x}_t) \in \Gamma\}$, with Eqs. (3) and (15), that is computed by using $\{\mathbb{B}_t^*\}_{t=0,\dots,d+1}$ of the Problem 3 instance.

5. When the ι-th AltSig query is issued for attribute $\mathbb{S} := (M, \rho)$, $\mathcal{B}_3$ answers as follows:
   (a) When $1 \le \iota \le h-1$, $\mathcal{B}_3$ answers semi-functional signature $\vec{\boldsymbol{s}}^*$ with Eqs. (5) and (16), that is computed by using $\{\mathbb{B}_t^*\}_{t=0,\dots,\ell+1}$ of the Problem 3 instance.
   (b) When $\iota = h$, $\mathcal{B}_3$ 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:
   $$\boldsymbol{s}_0^* := \boldsymbol{h}_{\beta,0}^*, \quad \boldsymbol{s}_i^* := \textstyle\sum_{j=1}^n z_j \boldsymbol{h}_{t,j}^* + \boldsymbol{r}_i^* \ \text{ for } i = 1, \dots, \ell,$$
   $$\boldsymbol{s}_{\ell+1}^* := \boldsymbol{h}_{\beta,d+1,1}^* + \mathsf{H}_{\mathsf{hk}}^{\lambda,\mathsf{D}}(m \,||\, \mathbb{S}) \cdot \boldsymbol{h}_{\beta,d+1,2}^*,$$
   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 $\boldsymbol{r}_i^* \xleftarrow{\mathsf{U}} \mathsf{span}\langle \boldsymbol{b}_{t,2n_t+1}^*, \dots, \boldsymbol{b}_{t,3n_t}^* \rangle$ with $t := \widetilde{\rho}(i)$ for $i = 1, \dots, \ell$.
   (c) When $\iota \ge h+1$, $\mathcal{B}_3$ answers normal signature $\vec{\boldsymbol{s}}^*$ with Eqs. (4), (5), and (6), that is computed by using $\{\mathbb{B}_t^*\}_{t=0,\dots,\ell+1}$ of the Problem 3 instance.

6. When $\mathcal{B}_3$ receives an output $(m', \mathbb{S}', \vec{\boldsymbol{s}}'^*)$ from $\mathcal{A}$, $\mathcal{B}_3$ calculates semi-functional verification text $\vec{\boldsymbol{c}} := (\boldsymbol{c}_0, \dots, \boldsymbol{c}_{\ell+1})$ with Eqs. (10), (11), and (12) as follows: $\boldsymbol{c}_i$ for $i = 1, \dots, \ell$ are calculated as Eq. (11) by using bases $\{\mathbb{B}_t\}_{t=1,\dots,d}$, and using the coefficient $s_0 := \sum_{k=1}^r f_k$,
   $$\alpha_l, \widetilde{\alpha}_l \xleftarrow{\mathsf{U}} \mathbb{F}_q \ \text{ for } l = 1, 2, \quad \widetilde{\boldsymbol{f}}_0 := \widetilde{\alpha}_1 \boldsymbol{e}_0 + \widetilde{\alpha}_2 \boldsymbol{b}_{0,1},$$
   $$\boldsymbol{f}_{d+1,j} := \alpha_1 \boldsymbol{e}_{d+1,j} + \alpha_2 \boldsymbol{b}_{d+1,j}, \quad \widetilde{\boldsymbol{f}}_{d+1,j} := \widetilde{\alpha}_1 \boldsymbol{e}_{d+1,j} + \widetilde{\alpha}_2 \boldsymbol{b}_{d+1,j} \quad \text{for } j = 1, 2;$$
   $$\boldsymbol{c}_0 := -s_0 \boldsymbol{b}_{0,1} - \widetilde{\boldsymbol{f}}_0 + \boldsymbol{q}_0, \quad \boldsymbol{c}_{\ell+1} := \widetilde{\boldsymbol{f}}_{d+1,1} - \mathsf{H}_{\mathsf{hk}}^{\lambda,\mathsf{D}}(m' \,||\, \mathbb{S}') \cdot \boldsymbol{f}_{d+1,1} + \boldsymbol{f}_{d+1,2} + \boldsymbol{q}_{\ell+1},$$
   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. $\mathcal{B}_3$ verifies the signature $(m', \mathbb{S}', \vec{\boldsymbol{s}}'^*)$ using Ver with the above $(\boldsymbol{c}_0, \dots, \boldsymbol{c}_{\ell+1})$, 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 3-$(h-1)$ (resp. Game 3-h) when $\beta = 0$ (resp. $\beta = 1$) except with probability $1/q$ (resp. $\mathsf{Adv}_{\mathcal{E}_{4,h}}^{\mathsf{H,CR}}(\lambda) + 2/q$ for a probabilistic machine $\mathcal{E}_4$ with essentially same running time as that of $\mathcal{A}$, where $\mathcal{E}_{4,h}(\cdot) := \mathcal{E}_4(h, \cdot)$).*

Proof. We consider the joint distribution of $\vec{c}$ and $\vec{s}^*$. 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 3-$(h-1)$ and Game 3-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 3-$(h-1)$ except that δ defined in Problem 3 is zero, i.e., except with probability $1/q$.

When $\beta = 1$, as in Remark 4, we need to check that w_0 in $\boldsymbol{c}_0$ (given in Eq. (10)), $\vec{w}_{\ell+1}$ in $\boldsymbol{c}_{\ell+1}$ (given in Eq. (12)), $\widetilde{r}_0$ in $\boldsymbol{s}_0^*$ and $\vec{\widetilde{r}}_{\ell+1}$ in $\boldsymbol{s}_{\ell+1}^*$ (given in Eq. (16)) are distributed as in those in Game 3-h, i.e., these are uniformly and independently distributed (with negligible probability). These are given as

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

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 α $\left(:= \mathsf{H}_{\mathsf{hk}}^{\lambda,\mathsf{D}}(m \,||\, \mathbb{S}) - \mathsf{H}_{\mathsf{hk}}^{\lambda,\mathsf{D}}(m' \,||\, \mathbb{S}')\right)$ except with probability $\mathsf{Adv}_{\mathcal{E}_{4,h}}^{\mathsf{H},\mathsf{CR}}(\lambda)$ for a probabilistic machine $\mathcal{E}_{4,h}$ 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 3-h except that δ defined in Problem 2 is zero or $\vec{w}_{\ell+1} \cdot \vec{\widetilde{r}}_{\ell+1} = 0$ i.e., except with probability $\mathsf{Adv}_{\mathcal{E}_{4,h}}^{\mathsf{H},\mathsf{CR}}(\lambda) + 2/q$. This completes the proof of Claim 2.

Therefore, $|\mathsf{Adv}_{\mathcal{A}}^{(3\text{-}(h-1))}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(3\text{-}h)}(\lambda)| \leq \mathsf{Adv}_{\mathcal{B}_{3,h}}^{\mathsf{P3}}(\lambda) + \mathsf{Adv}_{\mathcal{E}_{4,h}}^{\mathsf{H},\mathsf{CR}}(\lambda) + 1/q + 2/q = \mathsf{Adv}_{\mathcal{B}_{3,h}}^{\mathsf{P3}}(\lambda) + \mathsf{Adv}_{\mathcal{E}_{4,h}}^{\mathsf{H},\mathsf{CR}}(\lambda) + 3/q$ from Shoup's difference lemma. This completes the proof of Lemma 8. $\square$

Proof of Lemma 9

Lemma 9 *For any adversary* $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}^{(3\text{-}\nu_2)}(\lambda) \leq \mathsf{Adv}_{\mathcal{A}}^{(4)}(\lambda) + 1/q$.

Proof. To prove Lemma 9, we will show distribution $(\mathsf{param}_{\vec{n}}, \{\widehat{\mathbb{B}}_t\}_{t=0,..,d+1}, \{\widehat{\mathbb{B}}_t^*\}_{t=1,..,d+1}, \boldsymbol{b}_{0,3}^*, \{\mathsf{sk}_{\Gamma}^{(j)*}\}_{j=1,..,\nu_1}, \{\vec{s}^{(j)*}\}_{j=1,..,\nu_2}, \boldsymbol{c})$ in Game 3-ν_2 and that in Game 4 are equivalent, where $\mathsf{sk}_{\Gamma}^{(j)*}$ is the answer to the j-th key query, $\vec{s}^{(j)*}$ is that to the j-th signature query, and $\vec{c}$ 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 3-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, $\{\mathsf{sk}_\Gamma^{(j)*}\}_{j=1,\ldots,\nu_1}$, $\{\overrightarrow{\boldsymbol{s}}^{(j)*}\}_{j=1,\ldots,\nu_2}$, and $\overrightarrow{\boldsymbol{c}}$ can be expressed as keys, signatures, and verification text in two ways, in Game 3-ν_2 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-$\nu_2$ can be conceptually changed to Game 4 if $w_0 \neq 0$, i.e., except with probability $1/q$. □

Proof of Lemma 10

Lemma 10 *For any adversary $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}^{(4)}(\lambda) = 1/q$.*

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. (17)) is uniform and independent from all the other variables, and coefficient of $\boldsymbol{b}_{0,1}^*$ in $\boldsymbol{s}_0'^*$ is nonzero. Hence, $\mathsf{Adv}_{\mathcal{A}}^{(4)}(\lambda) = 1/q$. □

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. □

Theorem 4 *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 $\mathcal{E}_1, \mathcal{E}_2^+, \mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4$, whose running times are essentially the same as that of $\mathcal{A}$, such that for any security parameter λ,

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{MA\text{-}ABS,UF}}(\lambda) \leq \mathsf{Adv}_{\mathcal{E}_1}^{\mathsf{DLIN}}(\lambda) + \textstyle\sum_{h=0}^{\nu_1-1}\left(\mathsf{Adv}_{\mathcal{E}_{2,h}^+}^{\mathsf{DLIN}}(\lambda) + \mathsf{Adv}_{\mathcal{E}_{2,h+1}}^{\mathsf{DLIN}}(\lambda)\right) \\ + \textstyle\sum_{h=1}^{\nu_2}\left(\mathsf{Adv}_{\mathcal{E}_{3,h}}^{\mathsf{DLIN}}(\lambda) + \mathsf{Adv}_{\mathcal{E}_{4,h}}^{\mathsf{H,CR}}(\lambda)\right) + \epsilon,\end{aligned}$$

where $\mathcal{E}_{2,h}^+(\cdot) := \mathcal{E}_2^+(h,\cdot), \mathcal{E}_{2,h+1}(\cdot) := \mathcal{E}_2(h,\cdot)$ $(h = 0,\ldots,\nu_1 - 1)$, $\mathcal{E}_{3,h}(\cdot) := \mathcal{E}_3(h,\cdot), \mathcal{E}_{4,h}(\cdot) := \mathcal{E}_4(h,\cdot)$ $(h = 1,\ldots,\nu_2)$, ν_1 is the maximum number of $\mathcal{A}$'s UserReg *queries, ν_2 is the maximum number of $\mathcal{A}$'s* AltSig *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 proofs of Lemmas 5, 6, 7 and 8 are slightly changed; Lemmas 5 and 8 in this proof employ Problems 4 and 5 (to be shown below) in place of Problems 1 and 3, respectively, and Lemmas 6 and 7 employ Problem 5 in place of Problem 2.

Problems 1, 2 and 3 that do not include parameters G_0, G_1 and δG_1 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 4 and 5, 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. $\square$

Problems 4 and 5 and the related lemmas

We show Problems 4 and 5 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 4 and 5.

$$
\begin{aligned}
&\mathcal{G}_{\mathsf{ob}}{}'(1^\lambda, \vec{n}) : \mathsf{param}_{\mathbb{G}} := (q, \mathbb{G}, \mathbb{G}_T, G, e) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{bpg}}(1^\lambda),\\
&\quad n_0 := 1,\ n_{d+1} := 2, \quad \kappa, \xi \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times,\\
&\quad \text{for } t = 0, \ldots, d+1,\\
&\qquad N_t := 3n_t + 1 \text{ for } t = 0, \ldots, d+1, \quad \mathsf{param}_{\mathbb{V}_t} := (q, \mathbb{V}_t, \mathbb{G}_T, \mathbb{A}_t, e) := \mathcal{G}_{\mathsf{dpvs}}(1^\lambda, N_t, \mathsf{param}_{\mathbb{G}}),\\
&\qquad X_t := (\chi_{t,i,j})_{i,j} \xleftarrow{\mathsf{U}} GL(N_t, \mathbb{F}_q),\ (\vartheta_{t,i,j})_{i,j} := (X_t^{\mathrm{T}})^{-1},\\
&\qquad \boldsymbol{b}_{t,i} := \kappa(\chi_{t,i,1}, \ldots, \chi_{t,i,N_t})_{\mathbb{A}_t} = \kappa \textstyle\sum_{j=1}^{N_t} \chi_{t,i,j} \boldsymbol{a}_{t,j},\ \mathbb{B}_t := (\boldsymbol{b}_{t,1}, \ldots, \boldsymbol{b}_{t,N_t}),\\
&\qquad \boldsymbol{b}^*_{t,i} := \xi(\vartheta_{t,i,1}, \ldots, \vartheta_{t,i,N_t})_{\mathbb{A}_t} = \xi \textstyle\sum_{j=1}^{N_t} \vartheta_{t,i,j} \boldsymbol{a}_{t,j},\ \mathbb{B}^*_t := (\boldsymbol{b}^*_{t,1}, \ldots, \boldsymbol{b}^*_{t,N_t}),\\
&\quad G_0 := \kappa G, \quad G_1 := \xi G, \quad g_T := e(G, G)^{\kappa\xi},\\
&\quad \mathsf{param}_{\vec{n}} := (\{\mathsf{param}_{\mathbb{V}_t}\}_{t=0,\ldots,d+1}, g_T),\\
&\quad \mathsf{return} \quad (\mathsf{param}_{\vec{n}}, \{\mathbb{B}_t, \mathbb{B}^*_t\}_{t=0,\ldots,d+1}, G_0, G_1).
\end{aligned}
$$

Definition 18 (Problem 4) *Problem 4 is to guess* $\beta \in \{0,1\}$, *given* $(\mathsf{param}_{\vec{n}}, \{\mathbb{B}_t, \widehat{\mathbb{B}}^*_t\}_{t=0,\ldots,d+1}, \boldsymbol{e}_{\beta,0}, \{\boldsymbol{e}_{\beta,t,1}, \boldsymbol{e}_{t,i}\}_{t=1,\ldots,d+1;i=2,\ldots,n_t}, G_0, G_1) \xleftarrow{\mathsf{R}} \mathcal{G}^{\mathsf{P4}}_\beta(1^\lambda, \vec{n})$, *where*

$$
\begin{aligned}
&\mathcal{G}^{\mathsf{P4}}_\beta(1^\lambda, \vec{n}) : \quad n_0 := 1,\ n_{d+1} := 2, \quad (\mathsf{param}_{\vec{n}}, \{\mathbb{B}_t, \mathbb{B}^*_t\}_{t=0,\ldots,d+1}, G_0, G_1) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}{}'(1^\lambda, \vec{n}),\\
&\quad \widehat{\mathbb{B}}^*_t := (\boldsymbol{b}^*_{t,1}, \ldots, \boldsymbol{b}^*_{t,n_t}, \boldsymbol{b}^*_{t,2n_t+1}, \ldots, \boldsymbol{b}^*_{t,3n_t+1}) \quad \text{for } t = 0, \ldots, d+1,\\
&\quad \omega, \gamma_0, \gamma_t, w_0, w_{t,1}, \ldots, w_{t,n_t} \xleftarrow{\mathsf{U}} \mathbb{F}_q \quad \text{for } t = 1, \ldots, d+1,\\
&\quad \boldsymbol{e}_{0,0} := (\omega, 0, 0, \gamma_0)_{\mathbb{B}_0}, \quad \boldsymbol{e}_{1,0} := (\omega, w_0, 0, \gamma_0)_{\mathbb{B}_0},\\
&\quad \text{for } t = 1, \ldots, d+1;\\
&\qquad \boldsymbol{e}_{0,t,1} := (\ \overbrace{\omega, 0^{n_t-1},}^{n_t}\ \overbrace{0^{n_t},}^{n_t}\ \overbrace{0^{n_t},}^{n_t}\ \overbrace{\gamma_t}^{1}\ )_{\mathbb{B}_t},\\
&\qquad \boldsymbol{e}_{1,t,1} := (\ \omega, 0^{n_t-1},\ w_{t,1}, .., w_{t,n_t},\ 0^{n_t},\ \gamma_t\ )_{\mathbb{B}_t},\\
&\qquad \boldsymbol{e}_{t,i} := \omega \boldsymbol{b}_{t,i} \quad \text{for } i = 2, \ldots, n_t,\\
&\quad \text{return} \quad (\mathsf{param}_{\vec{n}}, \{\mathbb{B}_t, \widehat{\mathbb{B}}^*_t\}_{t=0,\ldots,d+1}, \boldsymbol{e}_{\beta,0}, \{\boldsymbol{e}_{\beta,t,1}, \boldsymbol{e}_{t,i}\}_{t=1,\ldots,d+1;i=2,\ldots,n_t}, G_0, G_1).
\end{aligned}
$$

for $\beta \xleftarrow{\mathsf{U}} \{0,1\}$. *For a probabilistic machine* $\mathcal{B}$, *the advantage of* $\mathcal{B}$ *for Problem 4*, $\mathsf{Adv}^{\mathsf{P4}}_{\mathcal{B}}(\lambda)$, *is similarly defined as in Definition 15.*

Lemma 11 *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, $\mathsf{Adv}^{\mathsf{P4}}_{\mathcal{B}}(\lambda) \leq \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}}(\lambda) + (d+7)/q$.

Lemma 11 is proven similarly to Lemma 1 in [24]. $\square$

Definition 19 (Problem 5) *Problem 5 is to guess* $\beta \in \{0,1\}$, *given* $(\mathsf{param}_{\vec{n}}, \{\widehat{\mathbb{B}}_t, \mathbb{B}_t^*\}_{t=0,..,d+1}, \boldsymbol{h}^*_{\beta,0}, \boldsymbol{e}_0, \{\boldsymbol{h}^*_{\beta,t,i}, \boldsymbol{e}_{t,i}\}_{t=1,..,d+1;i=1,..,n_t}, G_0, G_1, \delta G_1) \xleftarrow{\mathsf{R}} \mathcal{G}_\beta^{\mathsf{P5}}(1^\lambda, \vec{n})$, *where*

$$\begin{aligned}
&\mathcal{G}_\beta^{\mathsf{P5}}(1^\lambda, \vec{n}): \quad n_0 := 1,\ n_{d+1} := 2, \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 := (\boldsymbol{b}_{t,1}, \ldots, \boldsymbol{b}_{t,n_t}, \boldsymbol{b}_{t,2n_t+1}, \ldots, \boldsymbol{b}_{t,3n_t+1}) \quad \text{for } t = 0, \ldots, d,\\
&\quad u_0, \tau \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times,\ \omega, \delta, \delta_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q,\\
&\quad (z_{t,i,j})_{i,j=1,\ldots,n_t} := Z_t \xleftarrow{\mathsf{U}} GL(n_t, \mathbb{F}_q), \quad (u_{t,i,j})_{i,j=1,\ldots,n_t} := U_t := (Z_t^{-1})^{\mathrm{T}} \quad \text{for } t = 1, \ldots, d,\\
&\quad \boldsymbol{h}^*_{0,0} := (\delta, 0, \delta_0, 0)_{\mathbb{B}_0^*}, \quad \boldsymbol{h}^*_{1,0} := (\delta, u_0, \delta_0, 0)_{\mathbb{B}_0^*}, \quad \boldsymbol{e}_0 := (\omega, \tau u_0^{-1}, 0, 0)_{\mathbb{B}_0},\\
&\quad \text{for } t = 1, \ldots, d+1;\ i = 1, \ldots, n_t;\\
&\qquad (\ w_{t,i,j}\ )_{i,j=1,\ldots,n_t} := \tau \cdot Z_t, \quad \delta_{t,i,j} \xleftarrow{\mathsf{U}} \mathbb{F}_q \quad \text{for } j = 1, \ldots, n_t,\\
&\qquad \boldsymbol{h}^*_{0,t,i} := (\ \overbrace{0^{i-1}, \delta, 0^{n_t-i},}^{n_t}\ \overbrace{0^{n_t},}^{n_t}\ \overbrace{\delta_{t,i,1}, .., \delta_{t,i,n_t},}^{n_t}\ \overbrace{0}^{1}\ )_{\mathbb{B}_t^*},\\
&\qquad \boldsymbol{h}^*_{1,t,i} := (\ 0^{i-1}, \delta, 0^{n_t-i},\ u_{t,i,1}, .., u_{t,i,n_t},\ \delta_{t,i,1}, .., \delta_{t,i,n_t},\ 0\ )_{\mathbb{B}_t^*},\\
&\qquad \boldsymbol{e}_{t,i} := (\ 0^{i-1}, \omega, 0^{n_t-i},\ w_{t,i,1}, .., w_{t,i,n_t},\ 0^{n_t},\ 0\ )_{\mathbb{B}_t}\\
&\quad \text{return } (\mathsf{param}_{\vec{n}}, \{\widehat{\mathbb{B}}_t, \mathbb{B}_t^*\}_{t=0,..,d+1}, \boldsymbol{h}^*_{\beta,0}, \boldsymbol{e}_0, \{\boldsymbol{h}^*_{\beta,t,i}, \boldsymbol{e}_{t,i}\}_{t=1,..,d+1;i=1,..,n_t}, 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 5,* $\mathsf{Adv}_{\mathcal{B}}^{\mathsf{P5}}(\lambda)$, *is similarly defined as in Definition 15.*

Lemma 12 *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, $\mathsf{Adv}_{\mathcal{B}}^{\mathsf{P5}}(\lambda) \leq \mathsf{Adv}_{\mathcal{E}}^{\mathsf{DLIN}}(\lambda) + 5/q$.

Lemma 12 is proven similarly to Lemma 2 in [24]. $\square$