SCIS 2014 The 31st Symposium on
Cryptography and Information Security
kagoshima, Japan, Jan. 21 - 24, 2014
The Institute of Electronics,
Information and Communication Engineers

Proxy Signatures for General Access Structures

Tadatoshi Tanimura* Masayuki Abe*† Tatsuaki Okamoto*†

Abstract— Proxy signature schemes allow a signer to delegate his/her right to sign messages to someone else or a group of people that conforms to a policy. So far, only simple forms of policies have been considered and proposed as multi-proxy signatures and threshold proxy signatures. In this paper, we propose a proxy signature scheme that supports considerably more general policies for delegation. In our construction, a policy is expressed by a general monotone formula (or a monotone span program), that naturally allows to describe policies for multi-proxy signatures and threshold proxy signatures as special cases. Our scheme is perfectly private in the sense that the policy and proxy signers are information theoretically hidden against verifiers from signatures. It is existentially unforgeable under the decisional linear (DLIN) assumption (over bilinear pairing groups) in the standard model. The proposed scheme is efficient and applicable in practice.

Keywords: Proxy Signatures, Digital Signatures, General Access Structures

1 Introduction

1.1 Background

Proxy signature schemes allow a signer to delegate his/her right to sign messages to someone else or a group of people that conforms to a policy. The concept of proxy signature scheme was first introduced by Mambo et al. in 1996 [5]. So far, many extensions of the basic proxy signature scheme have been proposed.

A straightforward extension of proxy signatures has been proposed in [3] known as multi-proxy signatures. In multi-proxy signature schemes, an original signer can authorize a group of proxy members and only the cooperation of all the signers in the proxy group can generate the proxy signatures on behalf of the original signer.

Zhang [9] and Kim et al. [4] independently proposed the first threshold proxy signature scheme by incorporating the idea of secret sharing into the context of proxy signatures. In a $(t,\ n)$-threshold proxy signature scheme, at least t out of n proxy signers must cooperate to generate signatures. Multi-proxy signature scheme can be regarded as a special case of the $(t,\ n)$-threshold proxy signature scheme for $t = n$. Security of most threshold proxy signature schemes in the literature is proven only in the random oracle model. The one proposed by Beheshti-Atashgah et al. [2] is secure in the standard model requires non-standard relativized assumption.

Existing extensions of proxy signatures do not cover more general policies for delegation, a general access structure. Suppose, for instance, a CEO of a company goes to mountains for vacation and wants to delegate the right to sign messages to subordinates according to the company's policy which is as complex as (TH3 (sales manager, finance manager, human resource manager, supply chain manager)) OR (the executive secretary AND a director) OR (a director AND two managers). Furthermore, for privacy, the company does not like anyone to see which delegates (or CEO) issued the the signature. All existing proxy signature schemes fell short for such policies.

1.2 Our Results

We introduce a new proxy signature scheme that allows considerably more general delegation policies described by general access structures. We thus call it *proxy signatures for general access structures* and denote by PS-GAS. We formalize the security requirements of PS-GAS as perfect privacy and (existentially) unforgeability in active attack scenarios.

We then present an efficient construction of PS-GAS, whose access structure is expressed by a general monotone formula (or a monotone span program), that naturally supports the type of policies in multi-proxy signatures and threshold proxy signatures as special cases.

Besides being the first scheme that supports general access structures, our PS-GAS scheme is fully secure under well studied cryptographic assumptions in the standard model, i.e., without random oracles, unlike most constructions of (t, n)-threshold proxy signature schemes in the literature. It is fully secure, i.e., perfectly private and existentially unforgeable under the *decisional linear* (DLIN) assumption (over prime order pairing groups).

The proposed scheme is efficient and practical. The size of a signature is 13ℓ (prime-order pairing) group element length, where ℓ is the size of a policy. For example, it is around 4K bytes, when the size of a prime-order pairing group element is around the underlying fi-

* Graduate School of Informatics, Kyoto University
† NTT Secure Platform Laboratries

nite field size (e.g., 256 bits), and the policy with $\ell = 10$ expresses a formula with 4 AND and 5 OR operations over atomic equality terms in $\mathbb{F}_q$.

1.3 Key Techniques

The techniques of attribute-based encryption(ABE) are applicable to proxy signature schemes and our construction follows the approach as well. However, while known constructions bases on so-called *ciphertext-policy* (CP) ABE schemes, our PS-GAS scheme bases on a *key-policy* (KP) ABE for the first time. Our construction follows several established key ideas; dual pairing vector spaces (DPVS) [6, 8], fully secure functional encryption scheme [6], and decentralized attribute-based signature scheme [7].

In our construction, a signing key is shared according to the access structure so that only a set of proxy signers collecting necessary shares of the key can perform signature generation. This is a quite natural approach in the context of proxy signatures. Note that no central party exists among proxy signers in this approach.

To realize it, we apply the technique of decentralized attribute-based signature (DMA-ABS) scheme [7] where no central authority exists. Unfortunately, the DMA-ABS scheme is a *ciphertext-policy* (CP) type, where a secret key of a user is associated with attributes, not policies. (To the best of our knowledge, all ABS schemes in the literature are CP-type.) In contrast, our approach requires that a secret (signing) key of each user (proxy signer) should be associated with a policy (access structure), i.e., a *key-policy* (KP) type.

No existing signature scheme satisfies the both requirements simultaneously, decentralized key-collaboration (no central party among signers exists) and key-policy (signing keys are associated with a policy).

In this paper, we, for the first time, achieve a signature scheme that satisfies the both requirements, or a decentralized policy-based signature scheme. The DPVS-based techniques by the DMA-ABS scheme [7] are key tools for our construction, but it is still a challenging task to extend them to KP type signatures. We have developed some new techniques, a distributed signing protocol among multiple proxy signers, a new re-randomization technique, and a new type of signature verification where attributes are only used as dummies.

2 Preliminaries

2.1 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. A function $f : \mathbb{N} \to \mathbb{R}$ is *negligible* in λ (denoted by $f(\lambda) < \epsilon(\lambda)$), if for every constant $c > 0$, there exists an integer n such that $f(\lambda) < \lambda^{-c}$ for all $\lambda > n$.

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

2.2 Dual Pairing Vector Spaces by Direct Product of Bilinear Pairing Groups

In this paper, for simplicity of description, we focus on the *symmetric* version of dual pairing vector spaces (DPVS) [6] constructed by using symmetric bilinear pairing groups. The *asymmetric* version of DPVS can be constructed on asymmetric bilinear pairing groups, and see Appendix A.2 in the full version of [6] for the description of the asymmetric version of DPVS, $(q, \mathbb{V}, \mathbb{V}^*, \mathbb{G}_T, \mathbb{A}, \mathbb{A}^*, e)$. The symmetric version of DPVS is a special case of the asymmetric one with identifying $\mathbb{V} = \mathbb{V}^*$ and $\mathbb{A} = \mathbb{A}^*$.

Definition 1 (Symmetric bilinear pairing groups). *$(q, \mathbb{G}, \mathbb{G}_T, G, e)$ are a tuple of a prime q, cyclic additive group $\mathbb{G}$ and multiplicative group $\mathbb{G}_T$ of order q, $G \neq 0 \in \mathbb{G}$, and a polynomial-time computable nondegenerate bilinear pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ i.e., $e(sG, tG) = e(G, G)^{st}$ and $e(G, G) \neq 1$. Let $\mathcal{G}_{\mathsf{bpg}}$ be an algorithm that takes input 1^λ and outputs a description of bilinear pairing groups $(q, \mathbb{G}, \mathbb{G}_T, G, e)$ with security parameter λ.*

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

2.3 Decisional Linear (DLIN) Assumption

Definition 3 (DLIN: Decisional Linear 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* $\mathcal{G}_\beta^{\mathsf{DLIN}}(1^\lambda)$: $\mathsf{param}_{\mathbb{G}} := (q, \mathbb{G}, \mathbb{G}_T, G, e) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{bpg}}(1^\lambda)$, $\kappa, \delta, \xi, \sigma \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $Y_0 := (\delta+\sigma)G, Y_1 \xleftarrow{\mathsf{U}} \mathbb{G}$, return $(\mathsf{param}_{\mathbb{G}}, G, \xi G, \kappa G, \delta\xi G, \sigma\kappa G, Y_\beta)$*, 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:*

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

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.

2.4 Collision Resistant (CR) Hash Functions

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

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

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

2.5 Monotone Span Programs and Access Structures

Definition 4 (Monotone Span Programs [1]). *Let* $\mathcal{K}$ *be a field, and* $\{x_1,\ldots,x_n\}$ *be a set of variables. A monotone span program over* $\mathcal{K}$ *is a labeled matrix* $\hat{M} := (M, \rho)$ *where* M *is a* $(\ell \times r)$ *matrix over* $\mathcal{K}$ *and* ρ *is a labeling of the rows of* M *by literals from* $\{x_1,\ldots,x_n\}$ *(every row is labeled by one literal).*

A monotone span program accepts or rejects an input by the following criterion. For every input sequence $\delta \in \{0,1\}^n$ *define the submatrix* M_δ *of* M *consisting of those rows whose labels are set to 1 by the input* δ*, i.e., rows labeled by some* x_i *such that* $\delta_i = 1$.

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

We assume that no row M_i $(i = 1,\ldots,\ell)$ of the matrix M is $\vec{0}$. We now introduce a monotone access structure with evaluating map ρ by using the inner-product of attribute vectors, that is employed in the proposed PS-GAS scheme.

We now construct a secret-sharing scheme for a monotone span program.

Definition 5. *A secret-sharing scheme for a monotone span program* $\hat{M} := (M, \rho)$ *is:*

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

3 Proxy Signatures for General Access Structures (PS-GAS)

3.1 Definitions for PS-GAS

Definition 6 (Proxy Signatures for General Access Structures: PS-GAS).

Setup *This is a randomized algorithm. An original signer runs the algorithm* $\mathsf{Setup}(1^\lambda)$ *which outputs a public parameters* mpk *and a master secret key* msk*. The original signer publishes* mpk *and stores* msk.

PSetup *This is a randomized algorithm. Let* $N := \{1,\ldots,n\}$ *be the set of secret key holders. The original signer sets an access structure (monotone span program)* $\mathbb{S} := (M, \rho)$ *and a "text" which includes information about delegation as well as the related information* (aux, σ)*, and runs* $\mathsf{PSetup}(\mathbb{S}, \mathsf{mpk}, \mathsf{msk}, text)$ *which outputs public parameters* $\mathsf{pk}_{\mathsf{psetup}} := (aux, text, \sigma, \mathbb{S})$ *and secret keys* $\{\mathsf{sk}_{\mathsf{psetup}_j}\}_{j\in N} := \{(\mathsf{pk}_{\mathsf{psetup}}, \mathsf{sk}_j)\}_{j\in N}$ *to delegate the right to sign messages. The original signer gives* $\mathsf{sk}_{\mathsf{psetup}_j}$ *to the*

j-th secret key holder.

Sig *This is a randomized algorithm. Let K be a subset of N, which is a set of proxy signers. For simplicity of description, we let $K := \{1, \ldots, k\} \subseteq N$. The first signer U_1 runs Sig_1 which outputs $\tilde{\boldsymbol{s}}_1$. U_1 sends $\tilde{\boldsymbol{s}}_1$ to the second signer U_2. The j-th signer $U_j (2 \leq j \leq k-1)$ receives $\tilde{\boldsymbol{s}}_{j-1}$ from U_{j-1} and runs Sig_j which outputs $\tilde{\boldsymbol{s}}_j$. U_j sends $\tilde{\boldsymbol{s}}_j$ to the next signer U_{j+1}. The last signer U_k receives $\tilde{\boldsymbol{s}}_{k-1}$ from U_{k-1} and runs Sig_k which outputs $\tilde{\boldsymbol{s}}_k$. Finally output $\boldsymbol{s}^* := (\tilde{\boldsymbol{s}}_k, aux, text, \sigma)$ as $\mathsf{Sig}(\mathsf{mpk}, \{\mathsf{sk}_j\}_{j\in K}, K, m, \mathsf{pk}_{\mathsf{psetup}})$.*

Ver *To verify signature $\boldsymbol{s}^*$ on message m, using a set of public parameters mpk, a user runs $\mathsf{Ver}(\mathsf{mpk}, m, \boldsymbol{s}^*)$ which outputs boolean value $\mathsf{accept} := 1$ or $\mathsf{reject} := 0$.*

Let $\delta := (x_1, \ldots, x_n) \in \{0,1\}^n$ such that $x_i = 1$ if and only if $i \in \bigcup_{j\in K} I_j$, where $I_j := \{i \in \{1, \ldots, \ell\} | \rho(i) = j\}$. We say that $\mathbb{S}$ accepts K if and only if $\mathbb{S}$ accepts δ (as defined in Definition 3).

A PS-GAS scheme should have the following correctness property: for all $(\mathsf{mpk}, \mathsf{msk}) \xleftarrow{\mathsf{R}} \mathsf{Setup}(1^\lambda)$, all message m, all proxy signer sets K, all signing keys $\{\mathsf{sk}_{\mathsf{psetup}_j}\}_{j\in N} \xleftarrow{\mathsf{R}} \mathsf{PSetup}(\mathbb{S}, \mathsf{mpk}, \mathsf{msk}, text)$ all access structure $\mathbb{S}$ such that $\mathbb{S}$ accepts K, and all signatures $\boldsymbol{s}^* \xleftarrow{\mathsf{R}} \mathsf{Sig}(\mathsf{mpk}, \{\mathsf{sk}_j\}_{j\in K}, K, m, \mathsf{pk}_{\mathsf{psetup}})$, it holds that $\mathsf{Ver}(\mathsf{mpk}, \mathrm{m}, \boldsymbol{s}^*) = 1$ with probability 1.

Remark The information "*text*" can be the null string, where the proxy signatures can be indistinguishable from the signature of the original signer. The usage of "*text*" depends on the applications.

Similarly, σ can be the null string, when $(aux, text)$ is a part of mpk.

Definition 7 (Perfect Privacy of PS-GAS). *A PS-GAS scheme is perfectly private, if, for all $\mathsf{msk}, \mathsf{mpk} \xleftarrow{\mathsf{R}} \mathsf{Setup}(1^\lambda)$, for all $(\mathsf{pk}^{(1)}_{\mathsf{psetup}}, \{\mathsf{sk}^{(1)}_{\mathsf{psetup}_j}\}_{j\in N}) \xleftarrow{\mathsf{R}} \mathsf{PSetup}(\mathbb{S}^{(1)}, \mathsf{mpk}, \mathsf{msk}, text)$, for all $(\mathsf{pk}^{(2)}_{\mathsf{psetup}}, \{\mathsf{sk}^{(2)}_{\mathsf{psetup}_j}\}_{j\in N}) \xleftarrow{\mathsf{R}} \mathsf{PSetup}(\mathbb{S}^{(2)}, \mathsf{mpk}, \mathsf{msk}, text)$, all messages m, all proxy signer sets $K^{(1)} \subseteq N$ and $K^{(2)} \subseteq N$, all access structures $\mathbb{S}^{(1)}$ and $\mathbb{S}^{(2)}$ such that $\mathbb{S}^{(1)}$ accepts $K^{(1)}$ and $\mathbb{S}^{(2)}$ accepts $K^{(2)}$, the distributions of $\mathsf{Sig}(\mathsf{mpk}, \{\mathsf{sk}^{(1)}_j\}_{j\in K^{(1)}}, K^{(1)}, m, \mathsf{pk}^{(1)}_{\mathsf{psetup}})$ and $\mathsf{Sig}(\mathsf{mpk}, \{\mathsf{sk}^{(2)}_j\}_{j\in K^{(2)}}, K^{(2)}, m, \mathsf{pk}^{(2)}_{\mathsf{psetup}})$ are equal.*

Definition 8 (Unforgeability of PS-GAS). *For an adversary, we define $\mathsf{Adv}^{\mathsf{PS\text{-}GAS,UF}}_{\mathcal{A}}(\lambda)$ to be the success probability in the following experiment for any security parameter λ. A PS-GAS scheme is existentially unforgeable if the success probability of any polynomial-time adversary is negligible:*

1. *The challenger runs the setup algorithm, $\mathsf{mpk}, \mathsf{msk} \xleftarrow{\mathsf{R}} \mathsf{Setup}(1^\lambda)$. The challenger gives public parameters mpk to the adversary $\mathcal{A}$ and stores master secret key msk.*
2. *$\mathcal{A}$ may adaptively makes a polynomial number of queries of the following types ($\iota = 1, 2, \ldots$):*
   - *[PSetup query] $\mathcal{A}$ asks the challenger to create signing keys for an access structure $\mathbb{S}_\iota$ and $text_\iota$. The challenger runs $\mathsf{PSetup}(\mathbb{S}_\iota, \mathsf{mpk}, \mathsf{msk}, text_\iota)$ and records its output, $(\mathsf{pk}_{\mathsf{psetup}}, \{\mathsf{sk}_j\}_{j\in N})$, without giving them to $\mathcal{A}$.*
   - *[Signing query] $\mathcal{A}$ specifies a key, $(\{\mathsf{sk}_j\}_{j\in N}, \mathsf{pk}_{\mathsf{psetup}})$, with $(\mathbb{S}_\iota, text_\iota)$ that has already been created, and asks the challenger to perform a signing operation to create a signature for a message m, and a subset K such that $\mathbb{S}_\iota$ accepts K. The challenger computes the signature, $\boldsymbol{s}^* := \mathsf{Sig}(\mathsf{mpk}, \{\mathsf{sk}_j\}_{j\in K}, K, m, \mathsf{pk}_{\mathsf{psetup}})$, without giving them to $\mathcal{A}$.*
   - *[(Key or Signature) Reveal query] $\mathcal{A}$ asks the challenger to reveal a part of an already-created key specified by $(\mathbb{S}_\iota, text_\iota)$ and a subset K_ι (i.e., $(\mathsf{pk}_{\mathsf{psetup}}, \{\mathsf{sk}_j\}_{j\in K_\iota})$), or an already-created signature specified by $(\mathbb{S}_\iota, text_\iota, m, K)$.*

   Note that when key or signature creation requests are made, $\mathcal{A}$ does not automatically see the created key or signature. $\mathcal{A}$ sees it only when it makes a reveal query. Here, $text_\iota$ is used as a tag or identifier for a reveal query to specify an already-issued Psetup or signing query.
3. *At the end, the adversary outputs $(\boldsymbol{s}'^*, m')$.*

We say the adversary succeeds, if the following three conditions hold, (1) a signature created by a query with $(m', text')$ was never revealed (by a signature reveal query) to the adversary, (2) $\mathbb{S}_\iota$ does not accept K_ι in any key reveal query with $(\mathbb{S}_\iota, text_\iota, K_\iota)$, and (3) $\mathsf{Ver}(\mathsf{mpk}, m', \boldsymbol{s}'^*) = 1$.

3.2 Proposed Proxy Singature Scheme

We define function $\rho : \{1, \ldots, \ell\} \to \{1, \ldots, n\}$ by $\rho(i) = j$, where ρ is given in access structure $\mathbb{S} := (M, \rho)$. We refer to Section 2.1 for notations on DPVS, e.g., $(x_1, \ldots, x_N)_{\mathbb{B}}$, $(y_1, \ldots, y_N)_{\mathbb{B}^*}$ for $x_i, y_i \in \mathbb{F}_q$. Let $\Sigma^A := (KeyGen^A, Sign^A, Ver^A)$ be a digital signature scheme.

$\mathsf{Setup}(1^\lambda)$: $\mathsf{param}_{\mathbb{G}} := (q, \mathbb{G}, \mathbb{G}_T, G, e) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{bpg}}(1^\lambda)$,

$\kappa \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times$, $g_T := e(G,G)^\kappa, H := \mathsf{hk} \xleftarrow{\mathsf{R}} \mathsf{KH}_\lambda$ (Section 2.4),

for $t = 1, \ldots, d$,

(d is the maximum value of ℓ for an access structure)

$\mathsf{param}_{\mathbb{V}_t} := (q, \mathbb{V}_t, \mathbb{G}_T, \mathbb{A}_t, e) := \mathcal{G}_{\mathsf{dpvs}}(1^\lambda, 8, \mathsf{param}_{\mathbb{G}})$,

$X_t := (\chi_{t,i,j})_{i,j} \xleftarrow{\mathsf{U}} GL(8, \mathbb{F}_q)$, $(\vartheta_{t,i,j})_{i,j} := \kappa \cdot (X_t^{\mathrm{T}})^{-1}$,

$\boldsymbol{b}_{t,i} := (\chi_{t,i,1}, \ldots, \chi_{t,i,8})_{\mathbb{A}_t} = \sum_{j=1}^{8} \chi_{t,i,j} \boldsymbol{a}_{t,j}$,

$\mathbb{B}_t := (\boldsymbol{b}_{t,1}, \ldots, \boldsymbol{b}_{t,8})$,
$\boldsymbol{b}^*_{t,i} := (\vartheta_{t,i,1}, \ldots, \vartheta_{t,i,8})_{\mathbb{A}_t} = \sum_{j=1}^{8} \vartheta_{t,i,j} \boldsymbol{a}_{t,j}$,
$\mathbb{B}^*_t := (\boldsymbol{b}^*_{t,1}, \ldots, \boldsymbol{b}^*_{t,8})$,
$\pi,\ \widetilde{\varphi}_{t,1}, \xleftarrow{\mathsf{U}} \mathbb{F}_q$,
$\widetilde{\boldsymbol{b}}^*_{t,1} = (\overbrace{\pi}^{1}, \overbrace{0}^{1}, \overbrace{0^4}^{4}, \overbrace{\widetilde{\varphi}_{t,1}}^{1}, \overbrace{0}^{1})_{\mathbb{B}^*_t}$,
$\widehat{\mathbb{B}}_t := (\boldsymbol{b}_{t,1}, \ldots, \boldsymbol{b}_{t,4}, \boldsymbol{b}_{t,8})$,
$\widehat{\mathbb{B}}^*_t := (\widetilde{\boldsymbol{b}}^*_{t,1}, \boldsymbol{b}^*_{t,2}, \ldots, \boldsymbol{b}^*_{t,4}, \boldsymbol{b}^*_{t,7})$,
$(\mathsf{sk}^A, \mathsf{vk}^A) \xleftarrow{\mathsf{R}} KeyGen^A(1^\lambda)$.
return $\mathsf{msk} := (\{\mathbb{B}^*_t\}_{t\in[d]}, \mathsf{sk}^A)$,
$\mathsf{mpk} := (\{\mathsf{param}_{\mathbb{V}_t}, \widehat{\mathbb{B}}_t, \widehat{\mathbb{B}}^*_t\}_{t\in[d]},\ g_T,\ \mathsf{vk}^A, H)$

$\mathsf{PSetup}(\mathbb{S} := (M, \rho),\ \mathsf{mpk},\ \mathsf{msk},\ text)$:
$\vec{f} \xleftarrow{\mathsf{U}} \mathbb{F}_q^r,\ \vec{s} := (s_1, \ldots, s_\ell) := M \cdot \vec{f},\ \ s_0 := \vec{1} \cdot \vec{f}$,
$\sigma = Sign^A(\mathsf{sk}^A,\ g_T^{s_0} || text)$,
$\vec{f'} \xleftarrow{\mathsf{R}} \mathbb{F}_q^r$ s.t. $\vec{1} \cdot \vec{f'}^{\mathrm{T}} = 0$,
$\vec{s'}^{\mathrm{T}} := (s'_1, \ldots, s'_\ell)^{\mathrm{T}} := M \cdot \vec{f'}^{\mathrm{T}}$,
$I_j := \{i \in [\ell] \mid \ \rho(i) = j\}$ for $j = 1, \ldots, n$,
for $i = 1, \ldots, \ell$, $\varphi_i \xleftarrow{\mathsf{U}} \mathbb{F}_q$,
$\boldsymbol{k}^*_i := (\overbrace{s_i}^{1}, \overbrace{s'_i}^{1}, \overbrace{0^4}^{4}, \overbrace{\varphi_i}^{1}, \overbrace{0}^{1})_{\mathbb{B}^*_i}$,
$\mathsf{sk}_j = \{\boldsymbol{k}^*_i\}_{i\in I_j}$ for $j = 1, \ldots, n$,
$\mathsf{sk}_{\mathsf{psetup}\,j} := (\mathsf{pk}_{\mathsf{psetup}}, \mathsf{sk}_j)$,
return $\mathsf{pk}_{\mathsf{psetup}} := (\mathbb{S} := (M, \rho),\ g_T^{s_0},\ \sigma,\ text)$,
$\{\mathsf{sk}_{\mathsf{psetup}\,j}\}_{j\in N}$,

$\mathsf{Sig}(\mathsf{mpk}, \{\mathsf{sk}_j\}_{j\in K},\ K,\ m,$
$\mathsf{pk}_{\mathsf{psetup}} := (\mathbb{S} := (M, \rho),\ g_T^{s_0},\ \sigma,\ text))$:
For simplicity of description, we let $K := \{1, \ldots, k\}$.
Let $\mathbb{S}$ accept K, then
compute $\{\alpha_i\}_{i\in\bigcup_{i\in K} I_j}$ s.t. $\sum_{i\in\bigcup_{i\in K} I_j} \alpha_i M_i = \vec{1}$,
where M_i is the i-th row of M,
and $I_j := \{i \in \{1, \ldots, \ell\} \mid \ \rho(i) = j\}$.
The values of $\{\alpha_i\}$ are shared
among the signers, $\{U_j\}_{j\in K}$.
The j-th signer $U_j (1 \le j \le k)$ runs Sig_j
and generates signatures $\{\boldsymbol{s}^*_{i,j}\}_{i\in[\ell]}$.
Here, $\boldsymbol{s}^*_{i,0} := \boldsymbol{0}$ $(i = 1, .., \ell)$.
Then U_j sends $\{\boldsymbol{s}^*_{i,j}\}_{i\in[\ell]}$ to the next signer U_{j+1}.
$\mathsf{Sig}_j(\mathsf{mpk}, \mathsf{sk}_j,\ m,\ \{\boldsymbol{s}^*_{i,j-1}\}_{i\in[\ell]})$:
$(r_{i,j})_{i\in[\ell]} \xleftarrow{\mathsf{R}} \mathbb{F}_q^\ell$ *s.t.* $r_{1,j} + \cdots + r_{\ell,j} = 0$,
$(r'_{i,j})_{i\in[\ell]} \xleftarrow{\mathsf{R}} \mathbb{F}_q^\ell$ *s.t.* $r'_{1,j} + \cdots + r'_{\ell,j} = 0$;
for $i = 1, \ldots, \ell$, $\tau_{i,j} \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $\boldsymbol{r}^*_i \xleftarrow{\mathsf{U}} \boldsymbol{b}^*_{i,7}$
if $i \in I_j$
$\boldsymbol{s}^*_{i,j} := \boldsymbol{s}^*_{i,j-1} + \alpha_i \cdot \boldsymbol{k}^*_i + r_{i,j} \widetilde{\boldsymbol{b}}^*_{i,1} + r'_{i,j} \boldsymbol{b}^*_{i,2}$
$+\tau_{i,j} \left(\boldsymbol{b}^*_{i,3} + H(m, text) \boldsymbol{b}^*_{i,4}\right) + \boldsymbol{r}^*_i$,
else
$\boldsymbol{s}^*_{i,j} := \boldsymbol{s}^*_{i,j-1} + r_{i,j} \widetilde{\boldsymbol{b}}^*_{i,1} + r'_{i,j} \boldsymbol{b}^*_{i,2}$
$+\tau_{i,j} \left(\boldsymbol{b}^*_{i,3} + H(m, text) \boldsymbol{b}^*_{i,4}\right) + \boldsymbol{r}^*_i$.
The last signer U_k outputs $\{\boldsymbol{s}^*_{i,k}\}_{i\in[\ell]}$.
$\boldsymbol{s}^*_i := \boldsymbol{s}^*_{i,k}$,
$\boldsymbol{s}^* := (\{\boldsymbol{s}^*_i\}_{i\in[\ell]}, g_T^{s_0}, text, \sigma)$.
return $\boldsymbol{s}^*$

$\mathsf{Ver}(\mathsf{mpk},\ m,\ \boldsymbol{s}^*)$:
Let $\delta \in \mathbb{F}_q \ \ \eta_i, \tau'_i \xleftarrow{\mathsf{U}} \mathbb{F}_q$,
for $i = 1, \ldots, \ell$
$\boldsymbol{c}_i = (\overbrace{1}^{1}, \overbrace{\delta}^{1}, \overbrace{\tau'_i(H(m, text),\ -1)}^{2}, \overbrace{0^2}^{2}, \overbrace{0}^{1}, \overbrace{\eta_i}^{1})_{\mathbb{B}_i}$
return 1
if $\prod_{i=1}^{\ell} e(\boldsymbol{c}_i, \boldsymbol{s}^*_i) = g_T^{s_0}$ and $Ver^A(\mathsf{vk}^A, \sigma,\ g_T^{s_0} || text) = 1$,
return 0 otherwise.

[Correctness] If $\mathbb{S} := (M, \rho)$ accepts K,

$$
\begin{aligned}
&\prod_{i=1}^{\ell} e\left(\boldsymbol{c}_i, \boldsymbol{s}^*_i\right) \\
&= \prod_{i=1}^{\ell} e\left((1, \delta, \tau'_i(H(m, text), -1), 0^3, \eta_i)_{\mathbb{B}_i}, \boldsymbol{s}^*_i\right) \\
&= \prod_{j=1}^{k} \prod_{i\in I_j} e\left((1, \delta, \tau'_i(H(m, text), -1), 0^3, \eta_i)_{\mathbb{B}_i}, \alpha_i \boldsymbol{k}^*_i\right) \\
&\quad \times \prod_{j=1}^{k} \prod_{i=1}^{\ell} e((1, \delta, \tau'_i(H(m, text), -1), 0^3, \eta_i)_{\mathbb{B}_i}, \\
&\qquad\qquad (\pi r_{i,j}, r'_{i,j}, 0^4, \widetilde{\varphi}_{i,1} r_{i,j}, 0)_{\mathbb{B}^*_i}) \\
&\quad \times \prod_{j=1}^{k} \prod_{i=1}^{\ell} e((1, \delta, \tau'_i(H(m, text), -1), 0^3, \eta_i)_{\mathbb{B}_i}, \\
&\qquad\qquad \tau_{i,j}(\boldsymbol{b}^*_{i,3} + H(m, text) \boldsymbol{b}^*_{i,4})) \\
&\quad \times \prod_{j=1}^{k} \prod_{i=1}^{\ell} ee\left((1, \delta, \tau'_i(H(m, text), -1), 0^3, \eta_i)_{\mathbb{B}_i}, \boldsymbol{r}^*_i\right) \\
&= \prod_{j=1}^{k} \prod_{i\in I_j} g_T^{\alpha_i(s_i + \delta s'_i)} \times \prod_{j=1}^{k} \prod_{i=1}^{\ell} g_T^{\pi r_i + \delta r'_i} \\
&\quad \times \prod_{j=1}^{k} \prod_{i=1}^{\ell} {g_T}^0 \times \prod_{j=1}^{k} \prod_{i\in I_j} {g_T}^0 \\
&= g_T^{s_0}, \\
&\text{since } \prod_{j=1}^{k} \prod_{i\in I_j} \alpha_i s_i = s_0 \text{ and } \prod_{j=1}^{k} \prod_{i\in I_j} \alpha_i s'_i = 0.
\end{aligned}
$$

3.3 Security of the Proposed PS-GAS

Theorem 1. *The proposed PS-GAS scheme is perfectly private.*

Theorem 2. *The proposed PS-GAS scheme is unforgeable if the DLIN assumption holds over the underlying bilinear pairing groups and the underlying hash function family is collision resistant.*

The proofs of the theorems are given in Section 4.

Remarks

1. The DLIN assumption implies the discrete logarithm assumption, which implies the existence of collision resistant hash function family. Therefore, Theorem 2 holds solely under the DLIN assumption (in the standard model), although we usually

employ a very efficient hash function like SHA for practical applications assuming the collision resistance of such a practical hash function. Note that the collision resistance of a practical hash function (e.g., SHA) is still one of widely accepted assumptions.

2. Digital signature Σ A is used for certifying the value of $(g_T^{s_0}, text)$. It can be replaced by a registry model, where $(g_T^{s_0}, text)$ is included in mpk. For simplicity, in our security proof, we assume the registry model. It is easy to prove the security of our scheme in the signature-based certification model under the security (existentially unforgeability against adaptively chosen message attacks) of the underlying signature scheme Σ A.

3.4 Performance

The signature length of the proposed scheme is the size of 8ℓ group elements. Here, since this scheme can be implemented over a *prime order* paring group, the size of a group element can be around the size of $\mathbb{F}_q$ (e.g., 256 bits), and ℓ represents 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 (i.e., the number of group elements) 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. In those cases, the signature lengths of our scheme are 80 and 800 group elements, respectively.

4 Security Proofs

4.1 Proof of Theorem 1

Proof. Theorem 1 is true if the following statement is true.

For all $\mathsf{msk}, \mathsf{mpk} \xleftarrow{\mathsf{R}} \mathsf{Setup}(1^\lambda)$, for all $(\mathsf{pk}^{(1)}_{\mathsf{psetup}}, \{\mathsf{sk}^{(1)}_{\mathsf{psetup}_j}\}_{j\in N}) \xleftarrow{\mathsf{R}} \mathsf{PSetup}(\mathbb{S}^{(1)}, \mathsf{mpk}, \mathsf{msk}, text)$, for all $(\mathsf{pk}^{(2)}_{\mathsf{psetup}}, \{\mathsf{sk}^{(2)}_{\mathsf{psetup}_j}\}_{j\in N}) \xleftarrow{\mathsf{R}} \mathsf{PSetup}(\mathbb{S}^{(2)}, \mathsf{mpk}, \mathsf{msk}, text)$, all messages m, all proxy signer sets $K^{(1)} \subseteq N$ and $K^{(2)} \subseteq N$, all access structures $\mathbb{S}^{(1)}$ and $\mathbb{S}^{(2)}$ such that $\mathbb{S}^{(1)}$ accepts $K^{(1)}$ and $\mathbb{S}^{(2)}$ accepts $K^{(2)}$, the distributions of $\mathsf{Sig}(\mathsf{mpk}, \{\mathsf{sk}^{(1)}_j\}_{j\in K^{(1)}}, K^{(1)}, m, \mathsf{pk}^{(1)}_{\mathsf{psetup}})$ and $\mathsf{Sig}(\mathsf{mpk}, \{\mathsf{sk}^{(2)}_j\}_{j\in K^{(2)}}, K^{(2)}, m, \mathsf{pk}^{(2)}_{\mathsf{psetup}})$ are equal.

In the proposed PS-GAS scheme, $(\boldsymbol{s}^*_1, \ldots, \boldsymbol{s}^*_\ell) \xleftarrow{\mathsf{R}} \mathsf{Sig}(\mathsf{mpk}, \{\mathsf{sk}^{(1)}_j\}_{j\in K^{(1)}} K^{(1)}, m, \mathsf{pk}^{(1)}_{\mathsf{psetup}})$ is expressed by the following equation.
For $i = 1, \ldots, \ell$,

$$\begin{aligned}\boldsymbol{s}^*_i &:= \alpha_i \cdot \boldsymbol{k}^*_i + r_i \widetilde{\boldsymbol{b}}^*_{i,1} + r'_i \boldsymbol{b}^*_{i,2} \\ &\quad + \tau_i \left(\boldsymbol{b}^*_{i,3} + H(m, text)\boldsymbol{b}^*_{i,4}\right) + \boldsymbol{r}^*_i \\ &= (\alpha_i s_i + \pi r_i, \alpha_i s'_i + r'_i, \tau_i, \tau_i H(m, text), 0^2, \\ &\quad \alpha_i \varphi_i + \widetilde{\varphi}_{i,1} r_i + \xi_i, 0)_{\mathbb{B}^*_i},\end{aligned}$$

where $\xi_i, \tau_i, \pi, \varphi_i, \widetilde{\varphi}_{i,1} \xleftarrow{\mathsf{U}} \mathbb{F}_q$,
$\vec{f} \xleftarrow{\mathsf{U}} \mathbb{F}_q^r$, $\vec{s} := (s_1, \ldots, s_\ell) := M \cdot \vec{f}$, $s_0 := \vec{1} \cdot \vec{f}$, $\vec{f'} \xleftarrow{\mathsf{R}} \mathbb{F}_q^r$ s.t. $\vec{1} \cdot \vec{f'}^{\mathrm{T}} = 0$, $\vec{s'}^{\mathrm{T}} := (s'_1, \ldots, s'_\ell)^{\mathrm{T}} := M \cdot \vec{f'}^{\mathrm{T}}$, $(r_1, \ldots, r_\ell) \xleftarrow{\mathsf{U}} \{(r_1, \ldots, r_\ell) \in \mathbb{F}_q^\ell \mid r_1 + \cdots + r_\ell = 0\}$, $(r'_1, \ldots, r'_\ell) \xleftarrow{\mathsf{U}} \{(r'_1, \ldots, r'_\ell) \in \mathbb{F}_q^\ell \mid r'_1 + \cdots + r'_\ell = 0\}$, if $i \in \bigcup_{j\in K^{(1)}} I_j$, $\{\alpha_i\}_{i\in\bigcup_{j\in K^{(1)}} I_j} \in \{\{\alpha_i\}_{i\in\bigcup_{j\in K^{(1)}} I_j} \mid \sum_{i\in\bigcup_{j\in K^{(1)}} I_j} \alpha_i M_i = \vec{1}\}$, if $i \notin \bigcup_{j\in K^{(1)}} I_j$, $\alpha_i = 0$.

Let $w_i := \alpha_i s_i + \pi r_i, w'_i := \alpha_i s'_i + r'_i, \xi'_i \xleftarrow{\mathsf{U}} \mathbb{F}_q$, then $(\boldsymbol{s}^*_1, \ldots, \boldsymbol{s}^*_\ell)$ can be rephrased by the following equation. For $i = 1 \ldots, \ell$

$$\boldsymbol{s}^*_i := (w_i,\ w'_i,\ \tau_i(1, H(m, text)),\ 0^2,\ \xi'_i, 0)_{\mathbb{B}^*_i}.$$

Here, $(w_1, \ldots, w_\ell) \xleftarrow{\mathsf{U}} \{(w_1, \ldots, w_\ell) \in \mathbb{F}_q^\ell \mid w_1 + \cdots + w_\ell = s_0\}$ since $\sum_{i=1}^\ell \alpha_i s_i = s_0$ and $(r_1, \ldots, r_\ell) \xleftarrow{\mathsf{U}} \{(r_1, \ldots, r_\ell) \in \mathbb{F}_q^\ell \mid r_1 + \cdots + r_\ell = 0\}$, and $(w'_1, \ldots, w'_\ell) \xleftarrow{\mathsf{U}} \{(w'_1, \ldots, w'_\ell) \in \mathbb{F}_q^\ell \mid w'_1 + \cdots + w'_\ell = 0\}$ since $\sum_{i=1}^\ell \alpha_i s'_i = 0$ and $(r'_1, \ldots, r'_\ell) \xleftarrow{\mathsf{U}} \{(r'_1, \ldots, r'_\ell) \in \mathbb{F}_q^\ell \mid r'_1 + \cdots + r'_\ell = 0\}$.

In a similar way, $(\boldsymbol{s}^*_1, \ldots, \boldsymbol{s}^*_\ell) \xleftarrow{\mathsf{R}} \mathsf{Sig}(\mathsf{mpk}, \{\mathsf{sk}^{(2)}_j\}_{j\in K^{(2)}}, K^{(2)}, m, \mathsf{pk}^{(2)}_{\mathsf{psetup}})$ are expressed by the form above. Therefore, the distributions of $\mathsf{Sig}(\mathsf{mpk}, \{\mathsf{sk}^{(1)}_j\}_{j\in K^{(1)}}, K^{(1)}, m, \mathsf{pk}^{(1)}_{\mathsf{psetup}})$ and $\mathsf{Sig}(\mathsf{mpk}, \{\mathsf{sk}^{(2)}_j\}_{j\in K^{(2)}}, K^{(2)}, m, \mathsf{pk}^{(2)}_{\mathsf{psetup}})$ are equivalent. □

4.2 Proof of Theorem 2

The proof of Theorem 2 follows the proof strategy of [7], which is based on the dual system encryption methodology by Waters and DPVS by Okamoto and Takashima. The major difference between the proof of this theorem and [7] is that our proxy signature scheme is related to (decentralized) key-policy (KP)-ABE, while the ABS [7] is to (decentralized) ciphertext-policy (CP)-ABE. Here note that we directly construct our scheme and prove its security without (explicitly) using the relation with a KP-ABE.

Proof : To prove Theorem 2, we consider the following $(\nu_S + 4\nu_H + 4)$ 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 security game.

Here, secret key $\{\boldsymbol{k}^{(h)*}_i\}_{i\in I^{(h)}_j} = \mathsf{sk}^{(\mathsf{h})}_{\mathsf{j}}$ for $j \in K^{(h)}$, which is a reply to the h-th key reveal query with an access structure $\mathbb{S}^{(h)} := (M^{(h)}, \rho^{(h)})$ and subset $K^{(h)} \subseteq N$ for $h = 1, \ldots, \nu_H$, is:

$$\boldsymbol{k}^{(h)*}_i := (\ \overbrace{s^{(h)}_i}^{1},\ \overbrace{s^{(h)\prime}_i}^{1},\ \overbrace{0^2}^{2},\ \overbrace{\boxed{0^2}}^{2},\ \overbrace{\varphi^{(h)}_i}^{1},\ \overbrace{0}^{1}\ )_{\mathbb{B}^*_i}, \quad (1)$$

where $\vec{f}^{(h)} \xleftarrow{\mathsf{U}} \mathbb{F}_q^r$, $\vec{s}^{(h)} := (s^{(h)}_1, \ldots, s^{(h)}_\ell) := M \cdot \vec{f}^{(h)}$, $s^{(h)}_0 := \vec{1} \cdot \vec{f}^{(h)}$, $\vec{f}^{(h)\prime} \xleftarrow{\mathsf{R}} \mathbb{F}_q^r$ s.t. $\vec{1} \cdot (\vec{f}^{(h)\prime})^{\mathrm{T}} = 0$, $(\vec{s}^{(h)\prime})^{\mathrm{T}} := (s^{(h)\prime}_1, \ldots, s^{(h)\prime}_\ell)^{\mathrm{T}} := M \cdot (\vec{f}^{(h)\prime})^{\mathrm{T}}, \varphi^{(h)}_i \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $\theta^{(h)}_i, \theta^{(h)\prime}_i \xleftarrow{\mathsf{U}} \mathbb{F}_q$.

Basis $\{\widetilde{\boldsymbol{b}}^*_{i,\iota}\}_{i=1,..,\ell,\ \iota=1,2}$, which is a part of mpk, is:

$$\widetilde{\boldsymbol{b}}^*_{i,\iota} := (\overbrace{\ \pi,}^{1}\ \overbrace{\ 0,}^{1}\ \overbrace{\ 0^2,}^{2}\ \overbrace{\ 0,\boxed{0}\ ,}^{2}\ \overbrace{\ \widetilde{\varphi}_{t,\iota},}^{1}\ \overbrace{\ 0\ }^{1})_{\mathbb{B}^*_i}, \tag{2}$$

where $\pi \xleftarrow{\mathsf{U}} \mathbb{F}_q,\ \widetilde{\varphi}_{t,\iota} \xleftarrow{\mathsf{U}} \mathbb{F}_q$.

Signature $\{\boldsymbol{s}^*_i\}_{i=1,\ldots,\ell}$, which is a reply to the h-th signature reveal query with $(\mathbb{S}^{(h)}, text^{(h)}, m^{(h)}, K^{(h)})$ for $h = 1, \ldots, \nu_S$, is:

$$\boldsymbol{s}^{(h)*}_i := (\overbrace{\ w^{(h)}_i,}^{1}\ \overbrace{\ w^{(h)\prime}_i,}^{1}\ \overbrace{\ \tau^{(h)}_i(1, H(m^{(h)}, text^{(h)})),}^{2}\ \overbrace{\boxed{0^2},}^{2}\ \overbrace{\ \xi^{(h)}_i,}^{1}\ \overbrace{\ 0\ }^{1})_{\mathbb{B}^*_i}, \tag{3}$$

where $\zeta^{(h)}_i, \zeta^{(h)\prime}_i, \tau^{(h)}_i,\ \xi^{(h)}_i \xleftarrow{\mathsf{U}} \mathbb{F}_q,\ (w^{(h)}_i, \ldots, w^{(h)}_\ell) \xleftarrow{\mathsf{U}} \{(w_i, \ldots, w_\ell) \in \mathbb{F}^\ell_q \mid w_1 + \cdots + w_\ell = s_0\}, (w^{(h)\prime}_i, \ldots, w^{(h)\prime}_\ell) \xleftarrow{\mathsf{U}} \{(w'_i, \ldots, w'_\ell) \in \mathbb{F}^\ell_q \mid w'_1 + \cdots + w'_\ell = 0\}$.

Verification text $\{\boldsymbol{c}_i\}_{i=1,\ldots,\ell}$, for $(m',\ text',aux' := g_T^{s_0})$, which is used for verification of the output of the adversary $\mathcal{A}$ at the end of the game, is:

$$\boldsymbol{c}_i := (\overbrace{\ 1,}^{1}\ \overbrace{\ \delta,}^{1}\ \overbrace{\ \gamma_i(H(m', text'),\ -1),}^{2}\ \overbrace{\boxed{0^2}\ ,}^{2}\ \overbrace{\ 0,}^{1}\ \overbrace{\ \eta_i\ }^{1})_{\mathbb{B}_i}, \tag{4}$$

where $\delta,\ \eta_i,\ \gamma_i \xleftarrow{\mathsf{U}} \mathbb{F}_q$.

Game 1 : Same as Game 0 except that (a part of) the verification text, $\{\boldsymbol{c}_i\}_{i=1,\ldots,\ell}$, is:

$$\boldsymbol{c}_i := (\overbrace{\ 1,}^{1}\ \overbrace{\ \delta,}^{1}\ \overbrace{\ \gamma_i(H(m', text'),\ -1),}^{2}\ \overbrace{\boxed{\vec{z}_i}\ ,}^{2}\ \overbrace{\ 0,}^{1}\ \overbrace{\ \eta_i\ }^{1})_{\mathbb{B}_i}, \tag{5}$$

where $\vec{z}_i \xleftarrow{\mathsf{U}} \mathbb{F}^2_q$, and the other variables are generated as in Game 0.

Game 2-h ($h = 1, \ldots, \nu_S$): Game 2-0 is Game 1. Game 2-h is the same as Game 2-$(h-1)$ except that the reply $\{\boldsymbol{s}^{(h)*}_i\}_{i=1,\ldots,\ell}$ to the h-th signature reveal query is:

$$\boldsymbol{s}^{(h)*}_i := (\overbrace{\ w^{(h)}_i,}^{1}\ \overbrace{\ w^{(h)\prime}_i,}^{1}\ \overbrace{\ \tau^{(h)}_i(1, H(m^{(h)}, text^{(h)})),}^{2}\ \overbrace{\boxed{\vec{u}^{(h)}_i},}^{2}\ \overbrace{\ \xi^{(h)}_i,}^{1}\ \overbrace{\ 0\ }^{1})_{\mathbb{B}^*_i}, \tag{6}$$

where $\vec{u}^{(h)}_i \xleftarrow{\mathsf{U}} \mathbb{F}^2_q$, and the other variables are generated as in Game 2-$(h-1)$.

Game 3-h-1 ($h = 1, \ldots, \nu_H$) : Game 3-0-4 is Game 2-ν_S. Game 3-h-1 is the same as Game 3-$(h-1)$-4 except that the verification text, $\{\boldsymbol{c}_i\}_{i=1,\ldots,\ell}$, is:

$$\boldsymbol{c}_i := (\overbrace{\ 1,}^{1}\ \overbrace{\ \delta,}^{1}\ \overbrace{\ \gamma_i(H(m', text'),\ -1),}^{2}\ \overbrace{\boxed{\tau \cdot Z_i}\ ,}^{2}\ \overbrace{\boxed{\tau'}\ ,}^{1}\ \overbrace{\ 0\ ,}^{1}\ \overbrace{\ \eta_i\ }^{1})_{\mathbb{B}_i}, \tag{7}$$

where $Z_i \xleftarrow{\mathsf{U}} \mathbb{F}^\times_q,\ \tau, \tau' \xleftarrow{\mathsf{U}} \mathbb{F}_q$, and the other variables are generated as in Game 3-$(h-1)$-4.

Game 3-h-2 ($h = 1, \ldots, \nu_H$): Game 3-h-2 is the same as Game 3-h-1 except that secret key $\{\boldsymbol{k}^{(h)*}_i\}_{i \in I^{(h)}_j} = \mathsf{sk}^{(h)}_j$ for $j \in K^{(h)}$ is:

$$\boldsymbol{k}^{(h)*}_i := (\overbrace{\ s^{(h)}_i,}^{1}\ \overbrace{\ s^{(h)\prime}_i,}^{1}\ \overbrace{\ 0^2,}^{2}\ \overbrace{\boxed{r^{(h)}_i \cdot U_i},}^{1}\ \overbrace{\ 0,}^{1}\ \overbrace{\ \varphi^{(h)}_i,}^{1}\ \overbrace{\ 0\ }^{1})_{\mathbb{B}^*_i}, \tag{8}$$

where $\vec{g}^{(h)} \xleftarrow{\mathsf{U}} \{\vec{g}^{(h)} \in \mathbb{F}^r_q \mid \vec{1} \cdot (\vec{g}^{(h)})^{\mathrm{T}} = 0\}, r^{(h)}_i := M^{(h)}_i \cdot (\vec{g}^{(h)})^{\mathrm{T}},\ \omega^{(h)}_i \xleftarrow{\mathsf{U}} \mathbb{F}_q, U_i := Z_i^{-1}$ for $Z_i \xleftarrow{\mathsf{U}} \mathbb{F}^\times_q$, and the other variables are generated as in Game 3-h-1.

Game 3-h-3 ($h = 1, \ldots, \nu_H$) : Game 3-h-3 is the same as Game 3-h-2 except that secret key $\{\boldsymbol{k}^{(h)*}_i\}_{i \in I^{(h)}_j} = \mathsf{sk}^{(h)}_j$ for $j \in K^{(h)}$ is:

$$\boldsymbol{k}^{(h)*}_i := (\overbrace{\ s^{(h)}_i,}^{1}\ \overbrace{\ s^{(h)\prime}_i,}^{1}\ \overbrace{\ 0^2,}^{2}\ \overbrace{\boxed{r^{(h)\prime}_i \cdot U_i},}^{1}\ \overbrace{\ 0,}^{1}\ \overbrace{\ \varphi^{(h)}_i,}^{1}\ \overbrace{\ 0\ }^{1})_{\mathbb{B}^*_i}, \tag{9}$$

where $\vec{g}^{(h)\prime} \xleftarrow{\mathsf{U}} \mathbb{F}^r_q, r^{(h)\prime}_i := M^{(h)}_i \cdot (\vec{g}^{(h)\prime})^{\mathrm{T}}$, and the other variables are generated as in Game 3-h-2.

Game 3-h-4 ($h = 1, \ldots, \nu_H$) : Game 3-h-4 is the same as Game 3-h-3 except that secret key $\{\boldsymbol{k}^{(h)*}_i\}_{i \in I^{(h)}_j} = \mathsf{sk}^{(h)}_j$ for $j \in K^{(h)}$ is:

$$\boldsymbol{k}^{(h)*}_i := (\overbrace{\ s^{(h)}_i,}^{1}\ \overbrace{\ s^{(h)\prime}_i,}^{1}\ \overbrace{\ 0^2,}^{2}\ \overbrace{\ 0,}^{1}\ \overbrace{\boxed{r^{(h)\prime}_i},}^{1}\ \overbrace{\ \varphi^{(h)}_i,}^{1}\ \overbrace{\ 0\ }^{1})_{\mathbb{B}^*_i}, \tag{10}$$

where the variables are generated as in Game 3-h-3.

Game 4: Game 4 is the same as Game 3-ν_H-4 except that a part of mpk, $\{\widetilde{\boldsymbol{b}}^*_{i,\iota}\}_{i=1,..,\ell,\ \iota=1,2}$ is:

$$\widetilde{\boldsymbol{b}}^*_{i,\iota} := (\overbrace{\ \pi,}^{1}\ \overbrace{\ 0,}^{1}\ \overbrace{\ 0^2,}^{1}\ \overbrace{\ 0,\boxed{\eta}\ ,}^{2}\ \overbrace{\ \widetilde{\varphi}_{t,\iota},}^{1}\ \overbrace{\ 0\ }^{1})_{\mathbb{B}^*_t}, \tag{11}$$

where $\eta \xleftarrow{\mathsf{U}} \mathbb{F}_q$, and the other variables are generated as in Game 3-ν_H-4.

Game 5 : Game 5 is the same as Game 4 except that the verification text, $\{\boldsymbol{c}_i\}_{i=1,\ldots,\ell}$, is:

$$\boldsymbol{c}_i := (\overbrace{\boxed{\delta'}}^{1}, \overbrace{\delta}^{1}, \overbrace{\gamma_i(H(m', text'), -1)}^{2}, \overbrace{\tau \cdot Z_i}^{2}, \overbrace{\tau'}^{1}, \overbrace{0}^{1}, \overbrace{\eta_i}^{1})_{\mathbb{B}_i}, \tag{12}$$

where $\delta' \xleftarrow{\mathsf{U}} \mathbb{F}_q$, and the other variables are generated as in Game 4.

Let $\mathsf{Adv}^{(0)}_{\mathcal{A}}(\lambda)$ be $\mathsf{Adv}^{\mathsf{PS\text{-}GAS,UF}}_{\mathcal{A}}(\lambda)$ in Game 0, and $\mathsf{Adv}^{(1)}_{\mathcal{A}}(\lambda), \mathsf{Adv}^{(2\text{-}h)}_{\mathcal{A}}(\lambda), \mathsf{Adv}^{(3\text{-}h\text{-}1)}_{\mathcal{A}}(\lambda), \ldots, \mathsf{Adv}^{(3\text{-}h\text{-}4)}_{\mathcal{A}}(\lambda), \mathsf{Adv}^{(4)}_{\mathcal{A}}(\lambda), \mathsf{Adv}^{(5)}_{\mathcal{A}}(\lambda)$ be the advantage of $\mathcal{A}$ in Games $1, 2\text{-}h, 3\text{-}h\text{-}1, \ldots, 3\text{-}h\text{-}4, 4, 5$, respectively.

By evaluatning the gaps between the neighboring advantages, we obtain that for any PPT adversary $\mathcal{A}$ there exist PPT algorithms $\mathcal{B}_1, .., \mathcal{B}_4$ such that

$$\mathsf{Adv}^{(0)}_{\mathcal{A}}(\lambda) < \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{B}_1}(\lambda) + \sum_{h=1}^{\nu_S}(\mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{B}_{2\text{-}h\text{-}1}}(\lambda) + \mathsf{Adv}^{\mathsf{H,CR}}_{\mathcal{B}_{2\text{-}h\text{-}2}}(\lambda))$$
$$+ \sum_{h=1}^{\nu_H}\left(\mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{B}_{3\text{-}h\text{-}1}}(\lambda) + \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{B}_{3\text{-}h\text{-}2}}(\lambda)\right) + \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{B}_4}(\lambda) + \epsilon(\lambda).$$

Thus, $\mathsf{Adv}^{\mathsf{DMA\text{-}FS,UF}}_{\mathcal{A}}(\lambda) < \epsilon(\lambda)$, assuming that $\mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{B}}(\lambda) < \epsilon(\lambda)$ and $\mathsf{Adv}^{\mathsf{H,CR}}_{\mathcal{B}}(\lambda) < \epsilon(\lambda)$ for any probabilistic polynomial-time algorithm $\mathcal{B}$. □

References

1. Beimel, A., Secure schemes for secret sharing and key distribution. PhD Thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.
2. Beheshti-Atashgah, M., Bayat, M.: Designated verifier threshold proxy signature scheme without random oracles. ePrint, IACR, http://eprint.iacr.org/2012/488
3. S.J. Hwang and C.H. Shi, A simple multi-proxy signature scheme, Proceedings of the 10th National Conference on Information Security, Hualien, Taiwan, 2000, pp. 134-138.
4. Kim, S.J., Park, S.J.,Won, D.H.: Proxy signatures, revisited. ICICS'97, LNCS 1334, Springer-Verlag, Berlin. (1997) 223-232
5. Mambo, M., Usuda, K., Okamoto, E.: Proxy signature: delegation of the power to sign messages. IEICE Transactions on Fundamentals. 76(1996) 1338-1353
6. 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
7. Okamoto, T., Takashima, K.: Decentralized attribute-based signatures, In: PKC 2013, LNCS, vol. 7778, pp. 125-142. Sprinter Heidelberg (2013)
8. Okamoto, T., Takashima, K.: Efficient attribute-based signatures for non-monotone predicates in the standard model, In: PKC 2011, LNCS, vol. 6571, pp. 35-52. Springer Heidelberg (2011)
9. Zhang, K.: Threshold proxy signature schemes. Information Security Workshop, Japan. (1997) 191-197

A Lemmas for the Proof of Theorem 2

We will show 9 lemmas for the proof of Theorem 2. These lemmas can be proven in a manner similar to Lemmas 7–15 via Lemmas 1–6 shown in the full version of [7].

Lemma 1. *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, $|\mathsf{Adv}^{(0)}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(1)}_{\mathcal{A}}(\lambda)| < \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{B}_1}(\lambda) + \epsilon(\lambda)$.*

Lemma 2. *For any adversary $\mathcal{A}$, there exists probabilistic machines $\mathcal{B}_{2\text{-}1}$ and $\mathcal{B}_{2\text{-}2}$, whose running time is essentially the same as that of $\mathcal{A}$, such that for any security parameter λ, $|\mathsf{Adv}^{(2\text{-}(h-1))}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(2\text{-}h)}_{\mathcal{A}}(\lambda)| < \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{B}_{2\text{-}h\text{-}1}}(\lambda) + \mathsf{Adv}^{\mathsf{H,CR}}_{\mathcal{B}_{2\text{-}h\text{-}2}}(\lambda) + \epsilon(\lambda)$. where $\mathcal{B}_{2\text{-}h\text{-}\iota}(\cdot) := \mathcal{B}_{2\text{-}\iota}(h, \cdot)$, $\iota = 1, 2$.*

Lemma 3. *For any adversary $\mathcal{A}$, for any security parameter λ, $|\mathsf{Adv}^{(3\text{-}(h-1)\text{-}4)}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(3\text{-}h\text{-}1)}_{\mathcal{A}}(\lambda)| < \epsilon(\lambda)$.*

Lemma 4. *For any adversary $\mathcal{A}$, there exists a probabilistic machine $\mathcal{B}_{3\text{-}1}$, whose running time is essentially the same as that of $\mathcal{A}$, such that for any security parameter λ, $|\mathsf{Adv}^{(3\text{-}h\text{-}1)}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(3\text{-}h\text{-}2)}_{\mathcal{A}}(\lambda)| < \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{B}_{3\text{-}h\text{-}1}}(\lambda), \epsilon(\lambda)$., where $\mathcal{B}_{3\text{-}h\text{-}1}(\cdot) := \mathcal{B}_{3\text{-}1}(h, \cdot)$.*

Lemma 5. *For any adversary $\mathcal{A}$, for any security parameter λ, $\mathsf{Adv}^{(3\text{-}h\text{-}2)}_{\mathcal{A}}(\lambda) = \mathsf{Adv}^{(3\text{-}h\text{-}3)}_{\mathcal{A}}(\lambda)$.*

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

Lemma 7. *For any adversary $\mathcal{A}$, there exists a probabilistic machine $\mathcal{B}_4$, whose running time is essentially the same as that of $\mathcal{A}$, such that for any security parameter λ, $|\mathsf{Adv}^{(3\text{-}\nu_H\text{-}4)}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(4)}_{\mathcal{A}}(\lambda)| < \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{B}_4}(\lambda) + \epsilon(\lambda)$.*

Lemma 8. *For any adversary $\mathcal{A}$, for any security parameter λ, $\mathsf{Adv}^{(4)}_{\mathcal{A}}(\lambda) = \mathsf{Adv}^{(5)}_{\mathcal{A}}(\lambda)$.*

Lemma 9. *For any adversary $\mathcal{A}$, $\mathsf{Adv}^{(5)}_{\mathcal{A}}(\lambda) = 1/q < \epsilon(\lambda)$.*

Proof. Since the value of $\delta' \xleftarrow{\mathsf{U}} \mathbb{F}_q$ in $\{\boldsymbol{c}_i\}_{i=1,..,\ell}$ is independent from all the other variables, the verification equation, $\prod_{i=1}^{\ell} e(\boldsymbol{c}_i, \boldsymbol{s}^*_i) = g_T^{s_0}$, holds with probability $1/q$ for any $\{\boldsymbol{s}^*_i\}_{i=1,..,\ell}$ in Game 5. Hence, $\mathsf{Adv}^{(5)}_{\mathcal{A}}(\lambda) = 1/q$. □