SCIS 2013 The 30th Symposium on
Cryptography and Information Security
kyoto, Japan, Jan. 22 - 25, 2013
The Institute of Electronics,
Information and Communication Engineers

Fully Secure Ciphertext-Policy Functional Encryption with $O(1)$ Pairings

Reo Yoshida * Fumitaka Hoshino* Tetsutaro Kobayashi*

Abstract— Functional encryption is an advanced notion of encryption that includes identity-based encryption, hidden-vector encryption, predicate encryption, and attribute-based encryption. The existing adaptively secure and ciphertext-policy functional encryption scheme has $O(n)$ pairings in its decryption algorithm where the scheme deals with the relation class of access structure, inner-products and n is a dimension of a vector space for the inner-product relation. We propose a ciphertext-policy functional encryption scheme with a decryption that has the constant times pairings for n. The scheme is adaptively secure under the decisional linear assumption and deals with the relation class of access structure and inner-products. Furthermore, we implement the proposed scheme and compare with Okamoto-Takashima functional encryption scheme.

Keywords: Functional encryption, Key compression, Decisional linear assumption, Adaptive security

1 Introduction

1.1 Background

Functional encryption (FE) is an advanced notion of encryption that includes identity-based encryption (IBE) [4, 5, 6, 9, 10], hidden-vector encryption (HVE) [8], predicate encryption (PE) [12, 14, 16], and attribute-based encryption (ABE) [2, 11, 18, 19, 20], as special cases. In FE [13, 15, 16, 21], a secret key, $\mathtt{sk}_\Gamma$ is associated with a parameter, Γ, and message m is encrypted to a ciphertext $\mathtt{Enc}(m, \mathtt{pk}, \Psi)$ using public key $\mathtt{pk}$, another parameter Ψ. Ciphertext $\mathtt{Enc}(m, \mathtt{pk}, \Psi)$ can be decrypted by secret key $\mathtt{sk}_\Gamma$ if and only if a relation $R(\Gamma, \Psi)$ holds.

One of the most promising applications of FE is an access control system on the cloud which has a data sharing function. In this system, data can be shared with many people without leakage even via cloud providers, simply by encrypting the data using FE under their attributes Ψ and sending the ciphertext $\mathtt{Enc}(m, \mathtt{pk}, \Psi)$ to the recipients. The only recipients who have $\mathtt{sk}_\Gamma$ in which the parameter Γ holds $R(\Gamma, \Psi)$ can decrypt it. The significant advantage of using FE is that only one ciphertext needs to be prepared. This is in contrast to IBE including public key encryption (PKE), which requires making the ciphertexts for each recipient. Therefore, a functional encryption scheme reduces times of the encryption in this application, while it does not reduce times of the decryption.

In this paper, we focus on ciphertext-policy functional encryption scheme for the above applications. The existing fully secure functional encryption scheme with the relation class of access structure and inner-products [15] has $O(n)$ pairings in which n is the dimension of a vector space for the inner-products relation. We propose a functional encryption scheme with the ciphertext-policy payload-hiding security under DLIN assumption and a decryption algorithm with a constant pairing computation for the dimension n. To construct this scheme, we take a existing key compression technique [16, 17]. Special type matrix for a dual orthonormal basis of the dual pairing vector spaces make it possible to use the bilinear property of a pairing for reducing it. Okamoto and Takashima describe the possibility of the construction of this scheme. However, they do not explicitly describe it.

1.2 Our Results

- We take an existing key compression technique [16], and propose a ciphertext-policy functional encryption scheme with a decryption in which pairing computation is constant times for the dimension n of a inner-product vector space. Intuitively, this technique makes pairing computation converse to scalar multiplication in the decryption algorithm. Therefore, this technique works effectively if there are some differences of computational time between pairing and scalar multiplication.
- Our scheme can deal with access structures in addition to inner-products of attribute vectors and achieve adaptively payload-hiding security from the DLIN assumption as well as the ciphertext-policy scheme of [15].
- We implement the proposed scheme and compare the timings of each algorithms of it with that of [15]. We show almost all algorithms of our scheme

* NTT Secure Platform Laboratories, 3-9-11, Midori-cho, Musashino-shi, Tokyo.

are faster than that of [15] on various literals and attribute, predicate vectors.

1.3 Related Works

Lewko et.al. [13] proposed fully secure ABE and PE with a standard model. They constructed their schemes from a non-standard assumption, subgroup decision assumption for three primes and the n-eDDH assumption. Okamoto and Takashima [15] proposed a fully secure functional encryption in the standard model from the standard assumption, the decisional linear (DLIN) assumption. Their schemes have many pairing computations in their `Dec` algorithm. Waters [21] describe a methodology for realizing piphertext-policy attribute encryption (CP-ABE) under concrete and noninteractive cryptographic assumptions in the standard model. Boneh, Sahai and Waters [7] provide a natural generalization of the syntax of FE schemes, for capturing concrete primitives such as IBE, ABE, and PE as particular cases. They present an elegant indistinguishability-based security definition for this new primitive.

1.4 Organization

In Section 2, we give some notations for describing our scheme and the definitions for Functional Encryption. In Section 3, we describe the proposed scheme and its security. In Section 4, we compare the proposed scheme and cipher-text policy scheme of [15].

2 Definitions for Functional Encryption

In this section, we give some notations, the definition of the dual pairing vector spaces by direct product of symmetric pairing groups, span programs, inner-product of attribute vectors and access structures for the definitions of ciphertext-policy functional encryption and its security [1, 15].

2.1 Notations

We give the notations of this paper as well as [15]. When S is a random variable or distribution, $x \overset{R}{\leftarrow} S$ denotes that x is randomly selected from A according to its distribution. When S is a set, $x \overset{U}{\leftarrow} S$ denotes that x is uniformly selected from S. The notation $x := y$ denotes that x is set, defined or substituted by y. When a is a fixed value, $A(x) \rightarrow a$ (e.g. $A(x) \rightarrow 1$) denotes the event that machine (algorithm) A outputs a on input x. A function $f : \mathbb{N} \rightarrow \mathbb{R}$ 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 $\mathbb{F} \setminus \{0\}$ by $\mathbb{F}_q^{\times}$. A vector symbol denotes a vector representation over $\mathbb{F}_q$, e.g. $\overrightarrow{x}$ denotes $(x_1, x_2, \ldots, x_n) \in \mathbb{F}_q^n$. $\overrightarrow{x} \cdot \overrightarrow{y}$ denotes the inner-product $\sum_{i=1}^{n} x_i v_i$ for $\overrightarrow{x} := (x_1, \ldots, x_n)$, $\overrightarrow{y} := (y_1, \ldots, y_n) \in \mathbb{F}_q^n$. The symbol $\overrightarrow{0}$ denotes the zero vector in $\mathbb{F}_q^n$ for any n. X^T denotes the transpose of matrix X. I_ℓ and 0_ℓ denote the $\ell \times \ell$ identity matrix and the $\ell \times \ell$ zero matrix, respectively. A bold face letter denotes an element of vector space $\mathbb{V}$, e.g., $\boldsymbol{x} \in \mathbb{V}$. When $\boldsymbol{b}_i \in \mathbb{V}$ $(i = 1, \ldots, n)$, $\mathbf{span}\langle \boldsymbol{b}_1, \cdots, \boldsymbol{b}_n \rangle \subseteq \mathbb{V}$ denotes the subspace generated by $\boldsymbol{b}_1, \cdots, \boldsymbol{b}_n$. For vectors $\overrightarrow{x} := (x_1, \cdots, x_N)$, $\overrightarrow{y} := (y_1, \cdots, c_N) \in \mathbb{F}_q^N$ and bases $\mathbb{B} := (\boldsymbol{b}_1, \cdots, \boldsymbol{b}_N)$, $\mathbb{B}^* := (\boldsymbol{b}_1, \cdots, \boldsymbol{b}_N)$, $(\overrightarrow{x})_{\mathbb{B}} = (x_1, \cdots, x_N)_{\mathbb{B}}$ denotes the linear combination $\sum_{i=1}^{N} x_i \boldsymbol{b}_i$, and $(\overrightarrow{y})_{\mathbb{B}^*} = (y_1, \cdots, y_N)_{\mathbb{B}^*}$ denotes $\sum_{i=1}^{N} y_i \boldsymbol{b}_i^*$. For a format of attribute vectors $\overrightarrow{n} := (d; n_1, \cdots, n_d)$ that indicates the dimensions of vector spaces, $\overrightarrow{e}_{t,j}$ denotes the canonical basis vector $(\overbrace{0, \cdots, 0}^{j-1}, 1, \overbrace{0, \cdots, 0}^{n_t - j}) \in \mathbb{F}_q^{n_t}$ for $t = 1, \cdots, d$ and $j = 1, \cdots, n_t$. We describe the special matrix subgroups [16] below,

$$\mathcal{H}(n, \mathbb{F}_q) := \left\{ \begin{pmatrix} u & & & u'_1 \\ & \ddots & & \vdots \\ & & u & u'_{n-1} \\ & & & u'_n \end{pmatrix} \middle| \begin{array}{l} u, u'_l \in \mathbb{F}_q \text{ for } l = 1, \ldots, n, \\ \text{a blank element in the matrix} \\ \text{denotes } 0 \in \mathbb{F}_q. \end{array} \right\},$$

$$\mathcal{L}(w, n, \mathbb{F}_q) := \left\{ X := \begin{bmatrix} X_{1,1} & \cdots & X_{1,w} \\ \vdots & \ddots & \vdots \\ X_{w,1} & \cdots & X_{w,w} \end{bmatrix} \middle| \; X_{i,j} := \begin{bmatrix} u_{i,j} & & & u'_{i,j,1} \\ & \ddots & & \vdots \\ & & u_{i,j} & u'_{i,j,n-1} \\ & & & u'_{i,j,n} \end{bmatrix} \begin{array}{c} \in \mathcal{H}(n, \mathbb{F}_q) \\ \text{for } i, j = \\ 1, \ldots, w \end{array} \right\} \cap GL(wn, \mathbb{F}_q). \tag{1}$$

2.2 Definitions

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

For the asymmetric definition of DPVS, $(q$, $\mathbb{V}$, $\mathbb{V}^*$, $\mathbb{G}_T$, $\mathbb{A}$, $\mathbb{A}^*, e)$, see Appendix A.2 in [15].

Definition 2 *Dual pairing vector spaces (DPVS) $(q$, $\mathbb{V}$, $\mathbb{G}_T$, $\mathbb{A}$, $e)$ by a direct product of symmetric pairing groups $\mathtt{param}_{\mathbb{G}} := (q, \mathbb{G}, \mathbb{G}_T, G, e)$ 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 $e : \mathbb{V} \times \mathbb{V} \rightarrow \mathbb{G}_T$.*

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* $\boldsymbol{y} := (H_1,\ldots,H_N) \in \mathbb{V}$*. DPVS also has linear transformations* $\phi_{i,j}$ *on* $\mathbb{V}$ *s.t.* $\phi_{i,j}(\boldsymbol{a}_j) = \boldsymbol{a}_i$ *and* $\phi_{i,j}(\boldsymbol{a}_k) = 0$ *if* $k \neq j$*.*

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

We describe the random dual orthonormal basis generator $\mathcal{G}_{\mathrm{ob}}$.

$\mathcal{G}_{\mathrm{ob}}(1^\lambda, \vec{n})$:

$\mathtt{param}_{\mathbb{G}} := (q,\mathbb{G},\mathbb{G}_T,G,e) \leftarrow \mathcal{G}_{\mathrm{bpg}}(1^\lambda)$,

$N_0 := 5, N_t := 4n_t, \psi_t \xleftarrow{U} \mathbb{F}_q^\times, g_T := e(G,G)^\psi$

$\mathtt{param}_{\mathbb{V}_t} := (q,\mathbb{V}_t,\mathbb{G}_T,\mathbb{A}_t,e) := \mathcal{G}_{\mathrm{dpvs}}(1^\lambda, N_t, \mathtt{param}_{\mathbb{G}})$ $(t = 0,1,\ldots,d)$,

$\mathtt{param}_{\vec{n}} := (\{\mathtt{param}_{\mathbb{V}_t}\}_{t=0,\ldots,d}, g_T)$.

$X_0 := (\chi_{0,i,j}) \xleftarrow{U} GL(N_0,\mathbb{F}_q)$,

$X_t \xleftarrow{U} \mathcal{L}(4,n_t,\mathbb{F}_q)$

where $\{\mu^*_{t,i,j}, \mu'^*_{t,i,j,l}\}_{i,j=1,\ldots,4;l=1,\ldots,n_t}$

denotes nonzero entries X_t.

$(\theta_{t,i,j}) := \psi_t \cdot (X_t)^{-1}$,

$\boldsymbol{b}_{t,i} := (\theta_{t,i,1},\ldots,\theta_{t,i,N_t})_{\mathbb{A}_t}$,

$\mathbb{B}_t := (\boldsymbol{b}_{t,1},\cdots,\boldsymbol{b}_{t,N_t})$ for $t = 0,\ldots,d; i,j = 1,\ldots,N_t$.

$\boldsymbol{b}^*_{0,i} := (\chi_{0,i,1},\ldots,\chi_{0,i,5})_{\mathbb{A}_0}$ for $i = 1,\ldots,5$,

$\mathbb{B}^*_0 := (\boldsymbol{b}^*_{0,1},\ldots,\boldsymbol{b}^*_{0,5})$.

$B^*_{t,i,j} := \mu^*_{t,i,j}G, B'^*_{t,i,j,l} := \mu'^*_{t,i,j,l}G$,

for $t = 1,\ldots,d; i,j = 1,\ldots,4; l = 1,\ldots,N_t$.

return $(\mathtt{param}_{\vec{n}}, \mathbb{B}_0, \mathbb{B}^*_0, \{\mathbb{B}_t,$
$\{B^*_{t,i,j}, B'^*_{t,i,j,l}\}_{i,j=1,\ldots,4;l=1,\ldots,N_t}\}_{t=1,\ldots,d})$.

Definition 3 (**Inner-Products of Attribute vectors and Access Structures** [15]) *Under the notations of span programs in [15],* $\mathcal{U}_t$ $(t = 1,\ldots,d$ *and* $\mathcal{U}_t \subseteq \{0,1\}^*)$ *is a sub-universe, a set of attributes, each of which is expressed by a pair of a sub-universe id and an* n_t *dimensional vector, i.e.,* $(t,\vec{v})$*, where* $t \in \{1,\ldots,d\}$ *and* $\vec{v} \in \mathbb{F}_q^{n_t} \backslash \{\vec{0}\}$*.*

We define such an attribute to be a variable p *of a span program* $\hat{M} := (M,\rho)$*,* $p := (t,\vec{x})$*. An access structure* $\mathbb{S}$ *is span program* $\hat{M} := (M,\rho)$ *along with variables* $p := (t,\vec{x})$*,* $p' := (t',\vec{x}')$*, …, i.e.,* $\mathbb{S} := (M,\rho)$ *such that* $\rho : \{t,\ldots,\vec{x}), \neg(t',\vec{x}'),\ldots$*. Let* Γ *be a set of attributes, i.e.,* $\Gamma := \{(t,\vec{v}_t) | \vec{v}_t \in \mathbb{F}_q^{n_t} \backslash \{\vec{0}\}, 1 \le t \le d\}$*, where* $1 \le t \le d$ *means that* t *is an element of some subset of* $\{1,\ldots,d\}$*.*

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

For the definition of secret-sharing scheme for span program, see [13, 15].

Definition 4 (**Ciphertext-Policy Functional Encryption : CP-FE**) *A ciphertext-policy functional encryption scheme consists of four algorithms.*

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

`KeyGen` *takes as input a set of attributes,* $\Gamma := \{(t,\vec{v}_t) | \vec{v}_t \in \mathbb{F}_q^{n_t}, 1 \le t \le d\}$*,* `pk` *and* `sk`*. It outputs a decryption key.*

`Enc` *takes as input message* m*, access structure* $\mathbb{S} := (M,\rho)$*, and the public parameters* `pk`*. It outputs the ciphertext.*

`Dec` *takes as input ciphertext that was encrypted under access structure* $\mathbb{S}$*, the decryption key for a set of attributes* Γ*, and the public parameters* `pk`*. It outputs either plaintext* m *or the distinguished symbol* $\perp$*.*

Definition 5 *The model for proving the adaptively payload-hiding security of CP-FE under chosen plaintext attack is:*

`Setup` *The challenger runs the setup algorithm,* $(\mathtt{pk},\mathtt{sk}) \xleftarrow{R} \mathtt{Setup}(1^\lambda, \vec{n})$*, and gives the public parameters* `pk` *to the adversary.*

`Phase 1` *The adversary is allowed to issue a polynomial number of queries,* Γ*, to the challenger or oracle* $\mathtt{KeyGen}(\mathtt{pk},\mathtt{sk},\cdot)$ *for private keys,* $\mathtt{sk}_\Gamma$ *associated with* Γ*.*

`Challenge` *The adversary submits two messages* m_0*,* m_1 *and an access structure,* $\mathbb{S} := (M,\rho)$*, provided that the* $\mathbb{S}$ *does not accept any* Γ *sent to the challenger in Phase 1. The challenger flips a random coin* $b \xleftarrow{U} \{0,1\}$*, and computes* $\mathtt{ct}^{(b)}_{\mathbb{S}} \xleftarrow{R} \mathtt{Enc}(\mathtt{pk}, m_b, \mathbb{S})$*. It gives* $\mathtt{ct}^{(b)}_{\mathbb{S}}$ *to the adversary.*

`Phase 2` *The adversary is allowed to issue a polynomial number of queries,* Γ*, to the challenger or oracle* $\mathtt{KeyGen}(\mathtt{pk},\mathtt{sk},\cdot)$ *for private keys,* $\mathtt{sk}_\Gamma$ *associated with* Γ*, provided that* $\mathbb{S}$ *does not accept* Γ*.*

`Geuss` *The adversary outputs a guess* b' *of* b*.*

The advantage of an adversary $\mathcal{A}$ *in the above game is defined as* $\mathrm{Adv}^{\mathrm{CP-FE,PH}}_{\mathcal{A}}(\lambda) := \Pr[b' = b] - 1/2$ *for any security parameter* λ*. A CP-FE scheme is adaptively payload-hiding secure if all polynomial time adversaries have at most a negligible advantage in the above game.*

Definition 6 (**DLIN : Decisional Linear Assumption**) *The DLIN problem is to guess $\beta \in \{0,1\}$, given* ($\texttt{param}$, G, ξG, κG, $\delta\xi G$, $\sigma\kappa G$, Y_β) $\xleftarrow{R} \mathcal{G}_\beta^{\rm DLIN}(1^\lambda)$, *where*

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

for $\beta \xleftarrow{U} \{0,1\}$. *For a probabilistic machine* $\mathcal{E}$, *we define the advantage of* $\mathcal{E}$ *for the DLIN problem as:*

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

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

Definition 7 (Problem 1) *Problem 1 is to guess* β, *given* ($\texttt{param}_{\vec{n}}$, $\mathbb{B}_0$, $\hat{\mathbb{B}}_0^*$, $\boldsymbol{e}_{\beta,0}$, $\{\mathbb{B}_t$, $\hat{\mathbb{B}}_t^*$, $\{\boldsymbol{e}_{\beta,t,i}\ \}_{i=1,\ldots,n_t}$ $\}_{t=1,\ldots,d}) \xleftarrow{R} \mathcal{G}_\beta^{\rm P1}(1^\lambda, \vec{n})$ }) *where*

$$\mathcal{G}_{ob}(1^\lambda, \vec{n}) :$$
$$\hat{\mathbb{B}}_0^* := (\boldsymbol{b}_{0,1}^*, \boldsymbol{b}_{0,3}^*, \ldots, \boldsymbol{b}_{0,5}),$$
$$\omega, \tau, \gamma_0 \xleftarrow{U} \mathbb{F}_q^*,$$
$$\boldsymbol{e}_{0,0} := (\omega, 0, 0, 0, \gamma_0)_{\mathbb{B}_0}, \boldsymbol{e}_{1,0} := (\omega, \tau, 0, 0, \gamma_0)_{\mathbb{B}_0},$$
$$\text{for } t = 1, \ldots, d; i = 1, \ldots, n_t,$$
$$\vec{e}_{t,i} := (0^{i-1}, 1, 0^{n_t - i}) \in \mathbb{F}_q^{n_t},$$
$$U_t \leftarrow \mathcal{H}(n_t, \mathbb{F}_q) \cap GL(n_t, \mathbb{F}_q), \gamma_t \xleftarrow{U} \mathbb{F}_q,$$
$$\boldsymbol{e}_{0,t,i} := (\omega\vec{e}_{t,i}, 0^{n_t}, 0^{n_t}, \gamma_t\vec{e}_{t,i})_{\mathbb{B}_t},$$
$$\boldsymbol{e}_{1,t,i} := (\omega\vec{e}_{t,i}, \tau\vec{e}_{t,i}(U_t^{-1})^T, 0^{n_t}, \gamma_t\vec{e}_{t,i})_{\mathbb{B}_t}$$
$$\hat{\mathbb{B}}_t^* := (\boldsymbol{b}_{t,1}^*, \ldots, \boldsymbol{b}_{tn_t}^*, \boldsymbol{b}_{t,2n_t+1}^*, \ldots, \boldsymbol{b}_{t,4n_t}^*)$$
$$\text{return } (\texttt{param}_{\vec{n}}, \mathbb{B}_0, \hat{\mathbb{B}}_0^*, \boldsymbol{e}_{\beta,0},$$
$$\{\mathbb{B}_t, \hat{\mathbb{B}}_t^*, \{\boldsymbol{e}_{\beta,t,i}\}_{i=1,\ldots,n_t}\}_{t=1,\ldots,d}\}),$$

For a probabilistic machine $\mathcal{C}$, *we define the advantage of* $\mathcal{C}$ *for Problem 1*, $\text{Adv}_{\mathcal{C}}^{\rm P1}(\lambda)$, *as*

$$\text{Adv}_{\mathcal{C}}^{\rm P1}(\lambda) := \Big| \Pr[\mathcal{C}(1^\lambda, \varrho) \to 1 \mid \varrho \xleftarrow{R} \mathcal{G}_0^{\rm P1}(1^\lambda, \vec{n})]$$
$$- \Pr[\mathcal{C}(1^\lambda, \varrho) \to 1 \mid]\varrho \xleftarrow{R} \mathcal{G}_1^{\rm P1}(1^\lambda, \vec{n}) \Big|$$

Lemma 1 *For any adversary* $\mathcal{B}$, *there exists a probabilistic machine* $\mathcal{E}$, *whose running times are essentially the same as that of* $\mathcal{B}$, *such that for any security parameter* λ, $\text{Adv}_{\mathcal{B}}^{\rm P1}(\lambda) \leq \text{Adv}_{\mathcal{E}}^{\rm DLIN} + 5/q$.

The proof of Lemma 1 is similar to the security proof of Problem 1 in [15]. The main difference is that special form matrices Equation (1) as well as Lemma 4 of [16]. □

Problem 2 is to guess β, given ($\texttt{param}_{\vec{n}}$, $\mathbb{B}_0$, $\hat{\mathbb{B}}_0^*$, $\boldsymbol{h}_{\beta,0}^*$, $\{\mathbb{B}_t, \mathbb{B}_t^*, \{\boldsymbol{h}_{\beta,t,i}^*, \boldsymbol{e}_{t,i}\}_{i=1,\ldots,n_t}\}_{t=1,\ldots,d}) \xleftarrow{R} \mathcal{G}_\beta^{P2}(1^{\lambda,\vec{n}})$, where

$$\mathcal{G}_\beta^{\rm P2} : (\texttt{param}_{\vec{n}}, \mathbb{B}_0, \mathbb{B}_0^*, \{\mathbb{B}_t, \mathbb{B}_t^*\}_{t=1,\ldots,d}) \xleftarrow{R} \mathcal{G}_{ob}(1^\lambda, \vec{n}),$$
$$\hat{\mathbb{B}}_0 := (\boldsymbol{b}_{0,1}, \boldsymbol{b}_{0,3}, \boldsymbol{b}_{0,5}),$$
$$\delta, \rho, \varphi_0 \xleftarrow{U} \mathbb{F}_q, U_t \xleftarrow{U} \mathcal{H}(n_t, \mathbb{F}_q) \cap GL(n_t, \mathbb{F}_q), Z_t = (U_t^{-1})^T,$$
$$\boldsymbol{h}_{0,0}^* := (\delta, 0, 0, \varphi_0, 0)_{\mathbb{B}_0^*}, \boldsymbol{h}_{1,0}^* := (\delta, \rho, 0, \varphi_0, 0)_{\mathbb{B}_0^*},$$
$$\boldsymbol{e}_0 := (\omega, \tau, 0, 0, 0)_{\mathbb{B}_0},$$
$$\vec{e}_{t,i_t} := (0^{i_t - 1}, 1, 0^{i_t - 1}) \in \mathbb{F}_q^{n_t} \text{ for } i_t = 1, \ldots, n_t,$$
$$\boldsymbol{h}_{0,t,i}^* := (\delta\vec{e}_{t,i}, 0^{n_t}, \vec{\varphi}_{t,i}, 0^{n_t})_{\mathbb{B}_t^*},$$
$$\boldsymbol{h}_{1,t,i}^* := (\delta\vec{e}_{t,i}, \rho\vec{e}_{t,i}Z_t, \vec{\varphi}_{t,i}, 0^{n_t})_{\mathbb{B}_t},$$
$$\boldsymbol{e}_{t,i} := (\omega\vec{e}_{t,i}, \tau\vec{e}_{t,i}, 0^{n_t}, 0^{n_t})_{\mathbb{B}_t}$$
$$\text{return } (\texttt{param}_{\vec{n}}, \mathbb{B}_0, \hat{\mathbb{B}}_0^*, \boldsymbol{h}_{\beta,0}^*,$$
$$\{\hat{\mathbb{B}}_t, \mathbb{B}_t^*, \{\boldsymbol{h}_{\beta,t,i}^*, \boldsymbol{e}_{t,i}\}_{i=1,\ldots,n_t}\}_{t=1,\ldots,n_t}),$$

for $\beta \xleftarrow{U} \{0,1\}$. For a probabilistic adversary $\mathcal{B}$, the advantage of $\mathcal{B}$ for Problem 2, $\text{Adv}_{\mathcal{B}}^{\rm P2}(\lambda)$, is similarly defined as in Definition 7.

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

*Proof.*The proof of Lemma 2 is similar to the security proof of Problem 2 of [16]. The main difference is that the number of the dual orthonormal basis, t. However, this does not affect that security proof, we obtain the above advantage same as [16]. □

3 Our Scheme

3.1 Construction

In this section, we give the concrete construction of our scheme, below. To construct this scheme, we apply the key compression technique [16] to the chipertext-policy functional encryption scheme [15]. Special type matrix, Equation (1), for a dual orthonormal basis of the dual pairing vector spaces make it possible to use the bilinear property of a pairing for reducing it.

We assume that input vector, $\vec{v}_t := (v_{t,1}, v_{t,2}, \ldots, v_{t,n_t})$, has the index $v_{t,1} = 1$, and that input vector, $\vec{x_i} := (x_{i,1}, \ldots, x_{i,n_i})$, satifies $x_{i,n_i} \neq 0$.

```
Setup(1^λ, n⃗):
    $(\texttt{param}_{\vec{n}}, \mathbb{B}_0, \mathbb{B}_0^*, \{\mathbb{B}_t, \{B_{t,i,j}^*, B'^*_{t,i,j,l}\}_{i,j=1,\ldots,4;l=1,\ldots,n_t}$
    $\}_{t=1,\ldots,d}) \xleftarrow{R} \mathcal{G}_{\rm ob}(1^\lambda, \vec{n}), \hat{\mathbb{B}}_0 := (\boldsymbol{b}_{0,1}, \boldsymbol{b}_{0,3}, \boldsymbol{b}_{0,5}),$
    $\hat{\mathbb{B}}_0^* := (\boldsymbol{b}_{0,1}^*, \boldsymbol{b}_{0,3}^*, \boldsymbol{b}_{0,4}),$
    $\hat{\mathbb{B}}_t := (\boldsymbol{b}_{t,1}, \ldots, \boldsymbol{b}_{t,n_t}, \boldsymbol{b}_{t,3n_t+1}, \boldsymbol{b}_{t,4n_t})$ for $t = 1, \ldots, d$,
    return $\texttt{pk} := (1^\lambda, \texttt{param}_{\vec{n}}, \{\hat{\mathbb{B}}_t\}_{t=0,\ldots,d}),$
        $\texttt{sk} := (\hat{\mathbb{B}}_0^*, \{B_{i,j}^*, B'^*_{t,i,j,l}\}_{i=1,3;j=1,\ldots,4;l=1,\ldots,n_t;t=1,\ldots,d})$

KeyGen(pk, sk, $\Gamma = \{t, \vec{v}_t := (v_{t,1}, \ldots, v_{t,n_t}) \in \mathbb{F}_q^{n_t} \backslash \{\vec{0}\}$
    $\mid t = 1, \ldots, d, v_{t,1} = 1\})$ :
    $\delta, \varphi_0, \varphi_t \xleftarrow{U} \mathbb{F}_q,$
    $\boldsymbol{k}_0^* := (\delta,\, 0,\, 1,\, \varphi_0,\, 0)_{\mathbb{B}_0}$,
    $K^*_{t,1,j} := \delta B^*_{t,1,j} + \varphi_t B^*_{t,3,j}$,
    $K'^*_{t,2,j} := \sum_{l=1}^{n_t} v_{t,l}\ (\delta B'^*_{t,1,j,l} + \varphi_t B'^*_{t,3,j,l})$ for $j = 1,\ldots,4,\ t = 1,\ldots,d$,
    return $\mathtt{sk} := (\Gamma, \boldsymbol{k}_0^*, \{K^*_{t,1,j}, K^*_{t,2,j}\}_{j=1,\ldots,4;t=1,\ldots,d})$
```

```
$\mathtt{Enc}(\mathtt{pk},\, m,\, \mathbb{S} := (M,\rho))$:
    $\overrightarrow{s}^T := (s_1,\ldots,s_l)^T := M\overrightarrow{f}^T,\ \overrightarrow{f} \xleftarrow{U} \mathbb{F}_q^r$,
    $s_0 := \overrightarrow{1} \cdot \overrightarrow{f}^T,\ \overrightarrow{\eta}_0,\ \overrightarrow{\eta}_i,\ \zeta \xleftarrow{U} \mathbb{F}_q$,
    $\boldsymbol{c}_0 := (-s_0, 0, \zeta, 0, \eta_0)_{\mathbb{B}_0}$,
    if $\rho(i) = (t,\, \overrightarrow{x}_i := (x_{i,1},\ldots,x_{i,n_t}) \in \mathbb{F}_q^{n_t}\backslash\{\overrightarrow{0}\}$ $s.t.$ $x_{i,n_t} \neq 0)$,
        $\boldsymbol{c}_i := (s_i\overrightarrow{e}_{t,1} + \tau_i\overrightarrow{x}_i, 0^{n_t}, 0^{n_t}, \overrightarrow{\eta}_i)_{\mathbb{B}_t}$ for $i = 1,\cdots,l$.
    if $\rho(i) = \neg(t,\, \overrightarrow{x}_i)$,
        $\boldsymbol{c}_i := (s_i\overrightarrow{x}_i, 0^{n_t}, 0^{n_t}, \overrightarrow{\eta}_i)_{\mathbb{B}_t}$.
    $c_{d+1} := g_T^{\zeta} \cdot m$,
    return $\mathrm{ct}_{\overrightarrow{x}_i} := (\mathbb{S}, \boldsymbol{c}_0, \boldsymbol{c}_1, \ldots, \boldsymbol{c}_l, c_{d+1})$.
```

```
$\mathtt{Dec}(\mathtt{pk},\, \mathtt{sk}_\Gamma := (\Gamma, \boldsymbol{k}_0^*, \{K^*_{t,i,j}, K^*_{t,2,j}\}_{i,j=1,\ldots,4;t=1,\ldots,d}$,
    $\mathtt{ct}_{\overrightarrow{x}_i} := (\mathbb{S},\, \boldsymbol{c}_0,\, \boldsymbol{c}_1,\, \ldots,\, \boldsymbol{c}_l,\, \boldsymbol{c}_{d+1}))$ :
    If $\mathbb{S} := (M,\rho)$ accepts $\Gamma := \{(t,\, \overrightarrow{v}_t)\}$, then computes $I$ and $\{\alpha_i\}_{i\in I}$ s.t. $\overrightarrow{1} = \sum_{i\in I}\alpha_i M_i$ where $M_i$ is the $i$-th row of $M$, and $I \subseteq \{i \in \{1,\ldots,l\}| [\rho(i) = (t,\, \overrightarrow{x}_i) \cap (t,\, \overrightarrow{v}_t) \in \Gamma \cap \overrightarrow{x}_i \cdot \overrightarrow{v}_t = 0] \cup [\rho(i) = \neg(t,\, \overrightarrow{x}_i) \cap (t,\, \overrightarrow{v}_t) \in \Gamma \cup \overrightarrow{x}_i \cdot \overrightarrow{v}_t \neq 0]\}$.
    Parse $\boldsymbol{c}_i$ as a $4n_i$-tuple $(C_{i,1},\, \ldots,\, C_{i,4n_i})$,
     $D_{i,j} := \sum_{f=1}^{n_i-1} v_{t,f}C_{i,(j-1)n_i+f}$ for $j = 1,\ldots,4$; $i = 1,\ldots,l$,
     $F := e(\boldsymbol{c}_0, \boldsymbol{k}_0^*)\cdot\prod_{i\in I\cup\rho(i)=(t,\overrightarrow{x}_i)} (\prod_{j=1}^4 e(D_{i,j}, K^*_{t,1,j})\cdot e(C_{i,jn_t}, K^*_{t,2,j}))^{\alpha_i}\ \cdot$
                 $\prod_{i\in I\cup\rho(i)=\neg(t,\overrightarrow{x}_i)}(\prod_{j=1}^4 e(D_{i,j}, K^*_{t,1,j})\cdot e(C_{i,jn_t}, K^*_{t,2,j}))^{\alpha_i/(\overrightarrow{x}_i\cdot\overrightarrow{v}_t)}$
    return $m' := c_{d+1}/F$.
```

Remark. A partial output of $\mathtt{Setup}(1^\lambda, \overrightarrow{n})$, $\{B^*_{t,i,j}, B'^*_{t,i,j,l}\}_{t=1\ldots,d;i=1,3;j=1,\ldots,4;l=1,\ldots,n_t}$ can be identified with $\hat{\mathbb{B}}^*_t := (\boldsymbol{b}^*_{t,1},\ldots,\boldsymbol{b}^*_{4n_t})$ as well as the Remark 6 in [16]. Decryption $\mathtt{Dec}$ can be alternatively described as:

$$\mathtt{Dec}'(\mathtt{pk}, \mathtt{sk}_\Gamma := (\Gamma, \boldsymbol{k}_0^*, \{K^*_{t,i,j}, K^*_{t,2,j}\}_{t=1,\ldots,d;i,j=1,\ldots,4}),$$

$$\mathtt{ct}_{\overrightarrow{x}_i} := (\mathbb{S}, \boldsymbol{c}_0, \boldsymbol{c}_1, \ldots, \boldsymbol{c}_l, \boldsymbol{c}_{d+1})) :$$

$$\boldsymbol{k}_t^* := (\overbrace{v_{t,1}K^*_{t,1,1}, \ldots, v_{t,n_t-1}K^*_{t,1,1}, K^*_{t,2,1}}^{n_t}, \ldots,$$

$$\overbrace{v_{t,1}K^*_{t,1,4}, \ldots, v_{t,n_t-1}, K^*_{t,2,4}, K^*_{t,2,4}}^{n_t}),$$

that is, $\boldsymbol{k}_t^* = (\delta\overrightarrow{v}_t, 0^{n_t}, \varphi_t\overrightarrow{v}_t, 0^{n_t})_{\mathbb{B}_t^*}$,

$$F := e(\boldsymbol{c}_0, \boldsymbol{k}_0^*) \cdot \prod_{i\in I\cup\rho(i)=(t,\overrightarrow{x}_i)} e(\boldsymbol{c}_i, \boldsymbol{k}_t^*)^{\alpha_i}\cdot$$

$$\prod_{i\in I\cup\rho(i)=\neg(t,\overrightarrow{x}_i)} e(\boldsymbol{c}_i, \boldsymbol{k}_t^*)^{\alpha_i/(\overrightarrow{x}_i\cdot\overrightarrow{v}_t)},$$

return $m' := c_{d+1}/F$.

[Correctness] The above F can be $g_T^{\delta(-s_0+\sum_{i\in I}\alpha_i s_i)+\zeta} = g_T^{\zeta}$.

3.2 Security

In this section, we give the proof of our scheme.

Theorem 1 *The proposed scheme is adaptively payload-hiding against chosen plaintext attacks under the DLIN assumption. For any adversary $\mathcal{A}$, there exist probabilistic machines $\mathcal{E}_1$ and $\mathcal{E}_2$, whose runnig times are essentially the same as that of $\mathcal{A}$, such that for any security parameter λ, $\mathrm{Adv}^{\mathrm{CP-FE,PH}}_{\mathcal{A}}(\lambda) \leq \mathrm{Adv}^{\mathrm{DLIN}}_{\mathcal{E}_1} + \sum_{h=1}^{\nu}(\mathrm{Adv}^{\mathrm{DLIN}}_{\mathcal{E}_{2-h-1}}(\lambda) + \mathrm{Adv}^{\mathrm{DLIN}}_{\mathcal{E}_{2-h-2}}(\lambda) + \epsilon$, where $\mathcal{E}_{2-h-1} := \mathcal{E}_{2-1}(h,\cdot)$, $\mathcal{E}_{2-h-2}(\cdot) := \mathcal{E}_{2-2}(h,\cdot)$, ν is the maximum number of $\mathcal{A}'s$ key queries and $\epsilon := (11\nu+6)/q$.*

Proof. To prove Theorem 1, we consider the following $(3\nu+3)$ games. In Game 0, the components that are framed by in boxes indicate coefficients to be changed in a subsequent game. In the other games, the components framed in boxes indicate coefficients that were changed from the previous game.

Game 0 : Original game. That is, the reply to a key query for $\Gamma := \{(t, \overrightarrow{v})\}$ is :

$$\boldsymbol{k}_0 := (\delta, 0, 1, \varphi_0, 0)_{\mathbb{B}_0}, \boldsymbol{k}_t := (\delta\overrightarrow{v}_t, 0^{n_t}, \varphi_t\overrightarrow{v}_t, 0^{n_t})_{\mathbb{B}_0^*}$$

where $\delta, \varphi_0 \xleftarrow{U} \mathbb{F}_q$, $\overrightarrow{\varphi}_t \xleftarrow{U} \mathbb{F}_q^{n_t}$, for $(t, \overrightarrow{x}_t) \in \Gamma$. The challenge ciphertext for challenge plaintexts (m^0, m^1) and access structure $\mathbb{S} := (M,\rho)$ is :

$$\boldsymbol{c}_0 := (-s_0, \boxed{0}, \boxed{\zeta}, 0, \eta)_{\mathbb{B}_0},$$

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

if $\rho(i) = (t, \overrightarrow{x}_i), \boldsymbol{c}_i := (s_i\overrightarrow{e}_{t,1} + \theta\overrightarrow{x}_i, 0^{n_t}, \boxed{0^{n_t}}, \overrightarrow{\eta}_i)_{\mathbb{B}_t}$

if $\rho(i) = \neg(t, \overrightarrow{x}_i), \boldsymbol{c}_i := (s_i\overrightarrow{x}_i, \boxed{0^{n_t}}, 0^{n_t}, \overrightarrow{\eta}_i)_{\mathbb{B}_t}$,

$c_{d+1} := g_T^{\zeta}m^{(b)}$

where $\overrightarrow{f} \xleftarrow{U} \mathbb{F}_q^r$, $\overrightarrow{s}^T := (s_1,\ldots,s_l)^T := M\cdot\overrightarrow{f}^T$, $s_0 := \overrightarrow{1}\cdot\overrightarrow{f}^T$, $\zeta \xleftarrow{U} \mathbb{F}_q$, $\overrightarrow{\eta}_0$, $\overrightarrow{\eta}_t \xleftarrow{U} \mathbb{F}_q^{n_t}$, $\overrightarrow{e}_{t,1} := (1,0,\ldots,0) \in \mathbb{F}_q^{n_t}$.

Game 1 : This is same as **Game 0** except that the challenge ciphertext $(\boldsymbol{c}_0,\ldots,\boldsymbol{c}_l,c_{d+1})$ is,

$$\boldsymbol{c}_0 := (-s_0, \boxed{\omega_0}, \zeta, 0, \eta)_{\mathbb{B}_0},$$

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

if $\rho(i) = (t, \overrightarrow{x}_i), \boldsymbol{c}_i := (s_i\overrightarrow{e}_{t,1} + \theta_i\overrightarrow{x}_i, \boxed{\tau_i\overrightarrow{x}_iU_t}, 0^{n_t}, \overrightarrow{\eta}_i)_{\mathbb{B}_t}$

if $\rho(i) := \neg(t, \overrightarrow{x}_i), \boldsymbol{c}_i := (s_i\overrightarrow{x}_i, \boxed{\tau_i\overrightarrow{x}_iU_t}, 0^{n_t}, \overrightarrow{\eta}_i)_{\mathbb{B}_t}$

where ω_0, $\tau_i \xleftarrow{U} \mathbb{F}_q$ $(i = 1,\ldots,l)$, $U_t \xleftarrow{U} \mathcal{H}(n_t,\mathbb{F}_q) \cup GL(n_t,\mathbb{F}_q)$ $(t = 1,\ldots,n_t)$ and all the other variables are generated as in Game 0.

Game 2-h-1 ($h = 1,\ldots,\nu$) : Game 2-0-3 is Game 1. Game 2-h-1 is the same as Game 2-$(h-1)$-3 except that the reply to the h-th key query for $(t, \overrightarrow{v}) \in \Gamma$, $\boldsymbol{k}_0^*$, $\boldsymbol{k}_1^*$, is

$$\boldsymbol{k}_0^* := (\delta, \boxed{\rho}, 1, \varphi_0, 0)_{\mathbb{B}_0^*},$$

$$\boldsymbol{k}_t^* := (\delta\overrightarrow{v}_t, \boxed{\rho_t\overrightarrow{v}_tZ_t}, \overrightarrow{\varphi}_t, 0^{n_t})_{\mathbb{B}_t^*},$$

where $\rho \xleftarrow{U} \mathbb{F}_q$, $Z_t := (U_t^{-1})^T$ for $U \xleftarrow{U} \mathcal{H}(n_t, \mathbb{F}_q) \cup GL(n_t, \mathbb{F}_q)$ and all the other variables are generated as in Game 2-$(h-1)$-3.

Game 2-h-2 (h=1, . . ., ν) : Game 2-h-2 is the same as Game 2-h-1 except that a part of the reply to the h-th key query for $(t, \vec{v}_t) \in \Gamma$, $(\boldsymbol{k}_0, \boldsymbol{k}_t)_{t=1,\ldots,d}$ is,

$$\boldsymbol{k}_0^* := (\delta, \boxed{\omega}, 1, \varphi_0, 0)_{\mathbb{B}_0^*},$$
$$\boldsymbol{k}_t^* := (\delta \vec{v}_t, \rho_t \vec{v}_t Z_t, \vec{\varphi}_t, 0^{n_t})_{\mathbb{B}_t^*},$$

where $\omega \xleftarrow{U} \mathbb{F}_q$ and all the other variables are generated as in Game 2-h-1.

Game 2-h-3 ($h = 1, \ldots, \nu$) : Game 2-h-3 is the same as Game 2-h-2 except that the reply to the h-th key query for $\vec{v}$, $\boldsymbol{k}_0^*, \{\boldsymbol{k}_t^*\}$, is,

$$\boldsymbol{k}_0 := (\delta, \omega, 1, \varphi_0, 0)_{\mathbb{B}_0^*},$$
$$\boldsymbol{k}_t^* := (\delta \vec{v}_t, \boxed{0^{n_t}}, \vec{\varphi}_t, 0^{n_t})_{\mathbb{B}_t^*}$$

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

Game 3 : This is same as Game 2-ν-3 except that $\boldsymbol{c}_0$ and c_{d+1} of the challenge cipher text are

$$\boldsymbol{c}_0 := (-s_0, \omega_0, \zeta', 0, \eta_0)_{\mathbb{B}_0},$$
$$c_{d+1} := g_T^{\zeta} m^{(b)},$$

where $\zeta' \xleftarrow{U} \mathbb{F}_q$ (i.e., independent form $\zeta \xleftarrow{U} \mathbb{F}_q$), and all the other variables are generated as in Game 2-ν-3.

Let $\mathrm{Adv}_{\mathcal{A}}^{(0)}(\lambda)$, $\mathrm{Adv}_{\mathcal{A}}^{(1)}(\lambda)$, $\mathrm{Adv}_{\mathcal{A}}^{2-h-\iota}(\lambda)$ ($h = 1, \ldots, \nu$; $\iota = 1, 2, 3$) and $\mathrm{Adv}_{\mathcal{A}}^{(3)}(\lambda)$ be the advantage of $\mathcal{A}$ in Game 0, 1, 2-h-ι and 3, respectively. $\mathrm{Adv}_{\mathcal{A}}^{(0)}(\lambda)$ is equivalent to $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{CP-FE,PH}}(\lambda)$ and it is obtained that $\mathrm{Adv}_{\mathcal{A}}^{(3)}(\lambda) = 0$. We will show six Lemmas 3-7 that evaluate the gaps between pairs of $\mathrm{Adv}_{\mathcal{A}}^{(0)}(\lambda)$, $\mathrm{Adv}_{\mathcal{A}}^{(1)}(\lambda)$, $\mathrm{Adv}_{\mathcal{A}}^{(2-h-\iota)}(\lambda)$ for $(h = 1, \ldots, \nu; \iota = 1, 2, 3)$ and $\mathrm{Adv}_{\mathcal{A}}^{(3)}(\lambda)$. □

Theorem 1 can be true for the asymmetric version of DPVS because Our scheme does not use the symmetry property of DPVS.

Lemma 3 *For any adversary $\mathcal{A}$, there exists a probabilistic machine athcalB_1, whose running tme is essentially the same as that of $\mathcal{A}$, such that for any security parameter λ, $|\mathrm{Adv}_{\mathcal{A}}^{(0)}(\lambda) - \mathrm{Adv}_{\mathcal{A}}^{(1)}(\lambda)| \leq \mathrm{Adv}_{\mathcal{B}_1}^{\mathrm{P1}}(\lambda)$*

Lemma 4 *For any adversary $\mathcal{A}$, there exists a probabilistic machine $\mathcal{B}_{2-1}$, whose running tie is essentially the same as that of $\mathcal{A}$, such that for any security parameter λ, $|\mathrm{Adv}_{\mathcal{A}}^{(2-(h-1)-3)}(\lambda) - \mathrm{Adv}_{\mathcal{A}}^{(2-h-1)}(\lambda)| \leq \mathrm{Adv}_{\mathcal{B}_{2-h-1}}^{\mathrm{P2}}(\lambda)$, where $\mathcal{B}_{2-h-1}(\cdot) := \mathcal{B}_{2-1}(h, \cdot)$.*

Lemma 5 *For any adversary $\mathcal{A}$, for any security parameter λ, $|\mathrm{Adv}_{\mathcal{A}}^{(2-h-1)}(\lambda) - \mathrm{Adv}_{\mathcal{A}}^{(2-h-2)}(\lambda)| \leq 1/q$.*

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

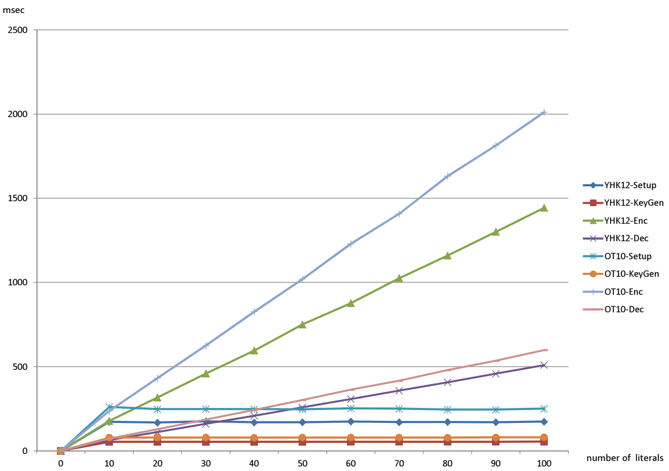

Figure 1: AND

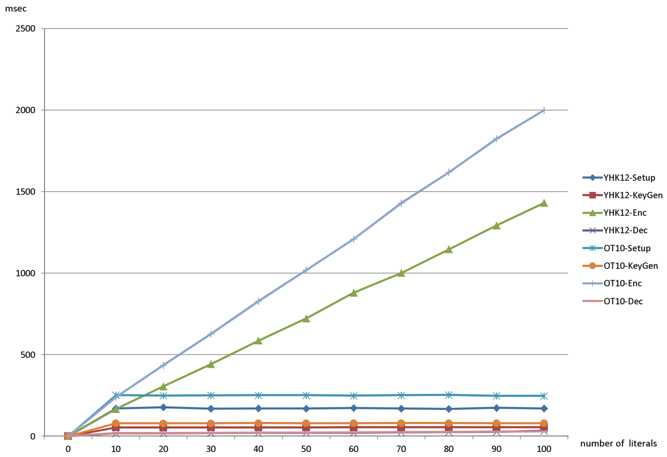

Figure 2: OR

Lemma 7 *For any adversary $\mathcal{A}$, for any security parameter λ, $|\mathrm{Adv}_{\mathcal{A}}^{(2-\nu-3)}(\lambda) - \mathrm{Adv}_{\mathcal{A}}^{(3)}(\lambda)| \leq 1/q$.*

Lemma 8 *For any adversary $\mathcal{A}$, for any security parameter λ, $\mathrm{Adv}_{\mathcal{A}}^{(3)}(\lambda) = 0$.*

*Proof.*Lemma 3, 4, 5, 6, and 7 is proven by the same manner as the proof of Lemma 7, 8, 9,10, and 11 in [16], respectively. The main difference is the number of the orthonormal basis, t, however, this does not affect their advantage because the part, $t = 1, \ldots, d$, of ciphertext and secret key have same distribution between these game changes. Therefore, all we have to is to evaluate the $t = 0$ part. In Lemma 8, the value of b is independent from the adversary's view in Game 3. Hence, $\mathrm{Adv}_{\mathcal{A}}^{3}(\lambda) = 0$. □

4 Comparison

We compare the CP scheme of [15] and [13] with the proposed one in Table 1. In order to show the decryption efficiency of our scheme, we consider asymmetric

Table 1: Comparison with CP scheme with in [15], where M_s , P, $|I|$ and $|\Gamma|$ represent scalar multiplication on the group $\mathbb{G}_s$ for $s = 1, 2$, pairing, the size of I and the size of Γ. SDA, Access st., inner-prod., means subgroup decision assumption, Access structure, and inner-products respectively.

	CP scheme of [15]	CP scheme of [13]	Proposed scheme
`KeyGen`	$10M_2 + 2n^2M_2$	$2M_1 + M_2 + \|\Gamma\|M_2$	$(12n + 10)M_2$
`Enc`	$(3n + 1)(n + 1)M_1$	$2\|I\|M_1$+M_2	$8n^2M_1$
`Dec`	$5P$+$(3n + 1)\|I\|P$	$(2\|I\| + 1)P$	$8\|I\|P$+ $4(n − 1)M_1$
Assumption	DLIN	SDA for 3 primes	DLIN
Relation	Access st. and inner-prod.	Access st.	Access st. and inner-prod.

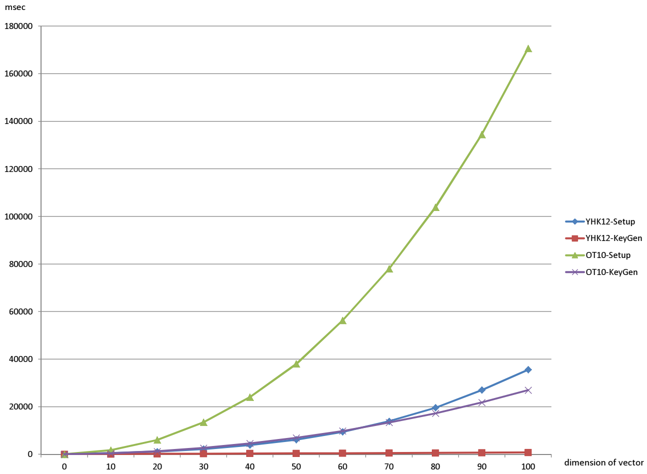

Figure 3: Vector-Setup, KeyGen

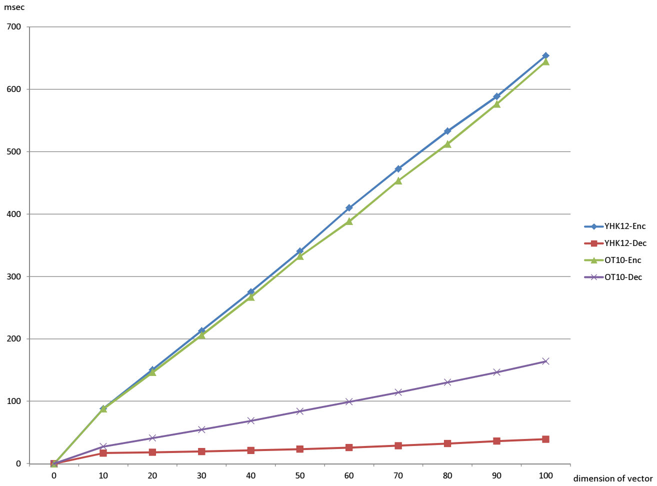

Figure 4: Vector-Enc, Dec

DPVS case for using prime fields computation at decryption. That is, pairing e is on $\mathbb{G}_1 \times \mathbb{G}_2$ s.t. $G_1 \neq G_2$. In `Dec` of the Table 1, the proposed scheme takes the constant times pairings for n, $8|I|P$+ $4(n−1)M_1$, while CP scheme of [15] takes $5P$+$(3n + 1)|I|P$. CP scheme of [13] takes the constant times for n, $(2|I|+1)P$, however the scheme is secure under subgroup decision assumption for 3 primes and has only the relation class of access structure.

Moreover, we compare the computation time of the CP scheme of [15] wtih the proposed one in Figure 1, 2, 3 and 4. We implemented and measured the elliptic computation, addition, scalar multiplication and pairing computation based on the parameters [3], and the scheme of [15], the proposed one using a PC with an Intel Core i7-3770K (3.50GHz) processor, 16GB memory. In the computation, we use only one core and thread.

Figure 1 shows the timings of each algorithm, `Setup`, `KeyGen`, `Enc` and `Dec` of [15] and the proposed one, increasing the times of AND literals. OT10 means the cipher-text policy scheme of [15] and YHK12 means the proposed scheme. The encryption of our scheme is about 25% faster than that of [15]. The decryption of our scheme is about 13% faster than that of [15].

Figure 2 shows the timings of each algorithm, `Setup`, `KeyGen`, `Enc` and `Dec` of [15] and the proposed one, increasing the times of OR literals. The encryption of our scheme is about 30% faster than that of [15]. The decryption of our scheme is almost same with [15].

Figure 3 shows the each timings of `Setup` and `KeyGen` of [15] and the proposed one, increasing the dimension of attribute and predicate vectors. The timings of the setup of our scheme is about 80% faster than that of [15]. at the 100 dimension. The key generation timings of our scheme is about 10% faster than that of [15] at the same dimension.

Figure 4 shows the each timings of `Enc` and `Dec` of [15] and the proposed one, increasing the dimension of attribute and predicate vectors. The timings of the encryption of our scheme is almost same with [15]. The decryption timings of our scheme is about 75% faster than that of [15]. at the 100 dimension.

5 Conclusion

In this paper, we propose a adaptively secure functional encryption scheme with $O(1)$ pairings. We take an existing key compression technique [16], and propose a ciphertext-policy functional encryption scheme with a decryption in which there is constant times pairing computation for the dimension n of a inner-product vector space. Moreover. we implement our scheme and compare with [15] scheme. We show almost all algorithms of the proposed scheme is faster than that of [15].

References

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

[2] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *IEEE Symposium on Security and Privacy*, pp. 321–334, 2007.

[3] Jean-Luc Beuchat, Jorge Enrique González-Díaz, Shigeo Mitsunari, Eiji Okamoto, Francisco Rodríguez-Henríquez, and Tadanori Teruya. High-speed software implementation of the optimal ate pairing over barreto-naehrig curves. In *Pairing*, pp. 21–39, 2010.

[4] D. Boneh and X. Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *EUROCRYPT*, pp. 223–238, 2004.

[5] D. Boneh and X. Boyen. Secure identity based encryption without random oracles. In *CRYPTO*, pp. 443–459, 2004.

[6] D. Boneh and M. K. Franklin. Identity-based encryption from the weil pairing. In *CRYPTO*, pp. 213–229, 2001.

[7] D. Boneh, A. Sahai, and B. Waters. Functional encryption: Definitions and challenges. In *TCC*, pp. 253–273, 2011.

[8] D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data. In *TCC*, pp. 535–554, 2007.

[9] C. Cocks. An identity based encryption scheme based on quadratic residues. In *IMA Int. Conf.*, pp. 360–363, 2001.

[10] C. Gentry. Practical identity-based encryption without random oracles. In *EUROCRYPT*, pp. 445–464, 2006.

[11] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM Conference on Computer and Communications Security*, pp. 89–98, 2006.

[12] J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *EUROCRYPT*, pp. 146–162, 2008.

[13] A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In *EUROCRYPT*, pp. 62–91, 2010.

[14] T. Okamoto and K. Takashima. Hierarchical predicate encryption for inner-products. In *ASIACRYPT*, pp. 214–231, 2009.

[15] T. Okamoto and K. Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In *CRYPTO*, pp. 191–208, 2010.

[16] T. Okamoto and K. Takashima. Achieving short ciphertexts or short secret-keys for adaptively secure general inner-product encryption. In *CANS*, pp. 138–159, 2011.

[17] T. Okamoto and K. Takashima. Achieving short ciphertexts or short secret-keys for adaptively secure general inner-product encryption. *IACR Cryptology ePrint Archive*, Vol. 2011, p. 648, 2011.

[18] R. Ostrovsky, A. Sahai, and B. Waters. Attribute-based encryption with non-monotonic access structures. In *ACM Conference on Computer and Communications Security*, pp. 195–203, 2007.

[19] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters. Secure attribute-based systems. In *ACM Conference on Computer and Communications Security*, pp. 99–112, 2006.

[20] A. Sahai and B. Waters. Fuzzy identity-based encryption. In *EUROCRYPT*, pp. 457–473, 2005.

[21] B. Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *Public Key Cryptography*, pp. 53–70, 2011.