Multi-Input Functional Encryption for Unbounded Inner Products

Bishnu Charan Behera and Somindu C. Ramanna

Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur, India
bishnu_charan_behera@iitkgp.ac.in, somindu@cse.iitkgp.ac.in

Abstract. In this work, we propose a construction for *Multi-Input Inner Product Encryption* (MIPFE) that can handle vectors of variable length in different encryption slots. This construction is the first of its kind, as all existing MIPFE schemes allow only equal length vectors. The scheme is constructed in the private key setting, providing privacy for both message as well as the function, thereby achieving the so-called *full-hiding* security. Our MIPFE scheme uses bilinear groups of prime order and achieves security under well studied cryptographic assumptions, namely, the symmetric external Diffie-Hellman assumption.

Keywords: Functional encryption (FE) · inner-product FE · multi-input FE · unbounded vectors

1 Introduction

Functional encryption (FE) [1,2,3,4,5] is a modern cryptographic primitive that generalizes *public key encryption* (PKE). Compared to traditional cryptographic approaches, FE offers more flexibility in sharing and dispersing information. As the name suggests, FE allows users to retrieve some function of the message, where an owner of a master secret key can generate a secret key sk_f for any function f, which can be used to recover $f(m)$ from the ciphertext ct_m of the message m.

In a *multi-input functional encryption* (MIFE) [6,7,8,9], there are multiple encryption slots to encrypt messages in different slots independently. The decryption key of MIFE decrypts n ciphertexts simultaneously to evaluate the joint functionality of the n messages. MIFE system is useful in scenarios when information to be processed together is supplied at different points of time or by multiple parties. Applications of MIFE include data mining over encrypted data coming from multiple sources, the multi-client delegation of computations to external servers, processing encrypted streaming data, non-interactive differentially private data releases, etc. The research on MIFE can be broadly characterized into two categories. The first category mainly emphasizes on the construction of MIFE for general multi-input functionalities such as Turing machines or arbitrary polynomial-size circuits. However, such construction relies on very strong cryptographic primitives like indistinguishability obfuscation, single-input FE for general circuits, etc. On the other hand, the second approach focuses on the construction of efficient MIFE based on standard cryptographic primitives like comparison or multi-input inner product.

The inner product version of MIFE is called *multi-input inner product functional encryption* (MIPFE) [10,11,12,13,14]. In MIPFE, there are several encryption slots to encrypt vectors $\{\vec{x}_\iota\}_{\iota \in S}$ in different slots independently. The decryption key $\mathrm{sk}_{\{\mathbf{y}_\iota\}_{\iota \in S}}$ of MIPFE can reveal $\sum_{\iota \in S} \langle \vec{x}_\iota, \mathbf{y}_\iota \rangle$ from the ciphertext ct_ι for all $\iota \in S$. The FE schemes can be broadly categorized into two types. The first type specifically provides message confidentiality. On the other hand, the second type defines a unified notion called *full-hiding security* where function privacy in addition to message privacy is guaranteed. To motivate the utility of function privacy, we cite an example from [14]: Consider a scenario when a hospital subscribes to an external cloud server to store its patient's medical records. To ensure the confidentiality of the medical record while performing various computations on the outsourced data remotely from time to time, a promising option for the hospital is to use a FE scheme where data can be encrypted locally before uploading to the cloud server. Now, when the hospital wants the information of all patients suffering from a certain disease, it must provide a functional decryption key to the cloud server to retrieve all the required information. However, if function privacy is not ensured, it may leak certain confidential information of some public figures (if someone is on the hospital record). It may damage their publicity resulting in financial loss. This situation is clearly undesirable from a privacy perspective.

To address this issue, many recent works have been initiated both in the single-input and multi-input settings. However, it has been observed that the function privacy in private key settings provides better security in comparison to the public key settings. In order to achieve it in a public key setting, the function must be chosen from certain high-entropy distribution. Instead, the extent of function privacy is much stronger in the private key setting. Even though it is a potential tool for function privacy, only a handful number of research works exist in the literature.

The work of Lin [15] computes the inner products of arbitrary polynomial degrees, where the standard inner product is a degree 2 function. But, it is a multilinear map-based construction. However, the work of Datta et al. [14] is much more practical. They proposed two constructions of MIPFE. The first design is a MIPFE function private scheme supporting a polynomial number of encryption slots. And, the second design is capable of handling an apriori unbounded number of encryption slots and multi-input inner product functions with arbitrary slot index sets of any polynomial size. Both the constructions obtained security under standard k-LIN assumptions.

In all existing MIPFE construction, the size of vectors at different slots is pre-determined and all public parameters of the system are chosen based on that. This makes them incapable of handling variable-length vectors at different slots. A layman approach to overcome this problem is to fix the size m to be arbitrarily large. This, however, would lead to large parameters whose size typically grows linearly in m. A natural question is whether there exists an MIPFE scheme with the parameters being completely unconstrained by the lengths of the vectors in keys and ciphertexts. In fact, MIPFE with variable-length vectors has many real-life applications. onsider n hospitals holding already shared private keys and each of these hospitals uses the FE system to store patient records on a cloud server. Suppose organization Z is curious about the weighted average of a specific medical outcome for all of the patients at these facilities. In that case, it can obtain a decryption key to bring up the needed outcome. But the existing constructions cannot handle this scenario, as it is not possible for these n hospitals to have the same number of patients. A solution to this problem is to use an MIPFE scheme that can handle variable length vectors at each slot.

Our Contributions In this work, we solve the above mentioned problem in the private key setting. Our construction of private-key MIPFE scheme is based on bilinear groups of prime order and achieves a unified notion of security called *full-hiding* security. We closely follow techniques from [16]. Security relies on the standard SXDH assumption. We consider two standard indexing methods from [16], namely *consecutive* and *separate*. Each vector element gets indexed automatically according to its position in consecutive settings. For instance, in (a, b, c), a is indexed to 1, b to 2 and c to 3. On the other hand, in separate indexing, each vector is specified with an index set. Let's assume (a, b, c) is indexed according to set $\{2, 6, 7\}$, that means a is indexed at 2, b at 6 and c at 7. For decryption, we use a form, namely *ct-dominant*. In a ct-dominant scheme, the decryption process works only if the index set of the decryption vector is a subset of the index set of the encryption vector. That is, if D_{sk} is the index set for secret key sk and D_{ct} is the index set for ciphertext ct, then we have $D_{\mathsf{sk}} \subseteq D_{\mathsf{ct}}$ for each encryption slot ι.

We now provide a brief overview of the construction. Let $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be an asymmetric bilinear map of prime order p. The master secret key is a pair of dual orthogonal bases $(\mathbf{B}_\iota, \mathbf{B}_\iota^*)$ for the vector spaces $\mathbb{G}_1^n$ and $\mathbb{G}_2^n$ along with scalars $\{s_\iota\}$ for each slot ι. For each slot ι, the ciphertext and secret key corresponding to vectors $\mathbf{x}_\iota$ and $\mathbf{y}_\iota$ respectively are of the form $(\pi_{\iota,i}(i, 1), s_\iota, x_{\iota,i}, z_\iota, 0, 0)\mathbf{B}_\iota$ and $(\rho_{\iota,i}(-1, i), u_\iota, y_{\iota,i}, r_{\iota,i}, 0, 0)\mathbf{B}_\iota^*$, respectively. The indexing technique from [4] is used in the first two prefixes. These two prefix dimensions specify the vector's index, and only if the indices of both the ciphertext and secret key are equal, correct decryption is possible. The last two dimensions are not used in the real scheme but reserved for defining semi-functional spaces in the proof of security. In [12], a secret key component of the form $k_T = e(P_1, P_2)^{\sum_\iota z_\iota r}$ is used to prevent partial leak of information. We follow the similar steps to avoid any partial leak, we use $k_T = e(P_1, P_2)^{-\sum_{\iota \in [n]} \left|D_\iota\right| s_\iota u_\iota}$, where $\left|D_\iota\right|$ is the cardinality of vectors in slot ι, s_ι is part of master secret and u_ι is the random scalar used for key generation.

The key and ciphertext are designed as shown above considering four main aspects. First, the slot matching is done through the dual orthogonal bases $(\mathbf{B}_\iota, \mathbf{B}_\iota^*)$. Next, index matching is done through the indexing technique in the first two prefixes. The fifth component is used to prevent the re-composition of ciphertext. Suppose, the slot size is 2 with encryption vectors $\vec{x}_1$ and $\vec{x}_2$ for slots 1 and 2, respectively. And decryption vectors $\mathbf{y}_1$ and $\mathbf{y}_2$. In the multi-input settings, the decryption algorithm should reveal $\langle \mathbf{x}_1, \mathbf{y}_1 \rangle + \langle \mathbf{x}_2, \mathbf{y}_2 \rangle$ but not the individual inner products $\langle \mathbf{x}_1, \mathbf{y}_1 \rangle$ and $\langle \mathbf{x}_2, \mathbf{y}_2 \rangle$. To prevent this partial leak of information, we add s_ι for each slot in the master secret key. And, there is a target $\mathbb{G}_T$-component k_T in the secret key as mentioned above, which cancels the other factors to reveal the desired result. More importantly, to obtain the joint evaluation over $\vec{x}_\iota$'s and $\mathbf{y}_\iota$'s, we need ciphertext for each slot $\iota \in [n]$.

Our indexing technique is inspired by the public key IPFE construction of [16]. On the other hand, the private key construction in [16] is more efficient. For each index, a dual orthonormal bases is generated using a pseudo-random function keyed by the master secret key k. It is natural to ask whether an extension of the same leads to a more efficient MIPFE scheme. Though we do not rule this out, observe that we need different dimensions in the dual orthonormal bases in order to match both slots and indices of vectors in each slot. We use n different dual orthonormal bases corresponding to the n slots in our construction, which are generated and stored as part of the master secret key in the setup phase.

2 Preliminaries

2.1 Notation

We write $x_1, \ldots, x_k \xleftarrow{\mathrm{R}} \mathcal{X}$ to indicate that $x_1, \ldots, x_k$ are sampled independently from a set $\mathcal{X}$ according to some distribution R (U denotes uniform distribution). For a (probabilicstic) algorithm $\mathcal{A}$, $y \longleftarrow \mathcal{A}(x)$ means that y is chosen according to the output distribution of $\mathcal{A}$ on input x. A function $f : \mathbb{N} \to \mathbb{R}$ is called *negligible* in $m (\in \mathbb{N})$ if for every constant $c > 0$, $\exists m_0 \in \mathbb{N}$ such that $f(m) < 1/m^c$ for all $m \geq m_0$. We write a negligible function in m as $\mathsf{negl}(m)$. For a natural number n, denote the set $\{1, 2, \ldots, n\}$ by $[n]$. For a prime p, we denote by $\mathbb{Z}_p$ the field of order p. Vectors over $\mathbb{Z}_p$ will be represented by bold face lower case letters (e.g. $\mathbf{v}$). $\mathbb{Z}_p^{n \times n}$ denotes the set of all $n \times n$ matrices over $\mathbb{Z}_p$ and $\mathsf{GL}_n(\mathbb{Z}_p)$, the general linear group of degree n over $\mathbb{Z}_p$ consisting of all invertible $n \times n$ matrices over $\mathbb{Z}_p$. We denote matrices over $\mathbb{Z}_p$ by bold-face upper case letters (e.g. $\mathbf{A}$). $\mathbf{A}^T$ denotes the transpose of matrix $\mathbf{A}$ and $\mathbf{A}^* = (\mathbf{A}^{-1})^T$ denotes the orthonormal dual basis of $\mathbf{A}$. $\mathbf{I}_n$ denotes the identity matrix of dimension n. For vectors $\mathbf{u} = (u_1, \ldots, u_n)$ and $\mathbf{v} = (v_1, \ldots, v_n)$, $\langle \mathbf{u}, \mathbf{v} \rangle$ denotes their inner product $\sum_{i=1}^{n} u_i v_i$.

Inner Products of Unbounded Vectors An unbounded vector is written as $\mathbf{x} = (x_i)_{i \in D}$ where D, a finite subset of $\mathbb{N}^*$ is called the domain of $\mathbf{x}$. In this paper, $x_i \in \mathbb{Z}_p$ for all $i \in D$, where p is defined by the bilinear map used in the construction of our encryption scheme. Given two vectors $\mathbf{x} = (x_i)_{i \in D}$ and $\mathbf{y} =$

$(y_i)_{i\in D'}$, the inner product $\langle \mathbf{x}, \mathbf{y} \rangle$ is a function defined as:

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i\in D\cap D'} x_i y_i$$

where the domains D and D' are non-empty finite subsets of $\mathbb{N}^*$. In the ct-dominant setting D' would be a subset of D. For simplicity we assume that $D = [m]$ for some $m \in \mathbb{N}$ and $D' \subseteq [m]$.

2.2 Bilinear Groups and Related Assumptions

A bilinear map $\mathcal{G} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, P_1, P_2, e, p)$ consists of cyclic groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ of prime order p with the first two groups given by generators P_1, P_2 respectively and an *efficiently computable* map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, with the following two properties:

Bilinearity: $e(aQ_1, bQ_2) = e(Q_1, Q_2)^{ab}$, for all $Q_1 \in \mathbb{G}_1$, $Q_2 \in \mathbb{G}_2$ and $a, b \in \mathbb{Z}_p$.
Non-degeneracy: $e(P_1, P_2)$ is a generator for $\mathbb{G}_T$ unless $P_1 = 0$ or $P_2 = 0$ where $P_1 \in \mathbb{G}_1$, $P_2 \in \mathbb{G}_2$.

The bilinear group generator $\mathsf{GroupGen}(\vartheta)$ takes a security parameter ϑ as input and returns a bilinear map $\mathcal{G}$ over a ϑ-bit prime p.

We represent an element $aP_\tau \in \mathbb{G}_\tau$ for $\tau \in \{1, 2\}$ as $[a]_\tau$ and an element $e(P_1, P_2)^a \in \mathbb{G}_T$ as $[a]_T$, where P_τ is a generator of G_τ. Given $[a]_\tau$ it is generally hard to obtain a. Observe that for $a, b \in \mathbb{Z}_p$, given $[a]_\tau, [b]_\tau$, one can compute $[a + b]_\tau$ as $[a]_\tau + [b]_\tau$. Furthermore, given $[a]_1, [b]_2$, one can compute $[ab]_T$ as $e([a]_1, [b]_2)$. For $\mathbf{A} = (a_{i,j})_{i,j\in[n]} \in \mathbb{Z}_p^{n\times n}$, $[\mathbf{A}]_\tau$ is defined as $([a_{i,j}]_\tau)_{i,j\in[n]}$. Similarly, for a vector $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{Z}_p^n$, $[\mathbf{x}]_\tau$ is defined as $([x_1]_\tau, \ldots, [x_n]_\tau)$.

Dual Pairing Vector Spaces. Let $n \in \mathbb{N}$ and let $(\mathbf{B}, \mathbf{B}^*)$ be dual orthonormal bases for $\mathbb{Z}_p^n$. Then $[\mathbf{B}]_1$ and $[\mathbf{B}^*]_2$ are dual orthonormal bases of vector spaces $\mathbb{G}_1^n$ and $\mathbb{G}_2^n$ respectively. The following two properties hold:

– For vectors $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_p^n$, $e([\mathbf{x}]_1, [\mathbf{y}]_2) = e(P_1, P_2)^{\langle \mathbf{x}, \mathbf{y} \rangle}$.
– Suppose that $\mathbf{B}$ is chosen at random from $\mathsf{GL}_n(\mathbb{Z}_p)$. Then for arbitrary vectors $\mathbf{x}_1, \ldots, \mathbf{x}_k, \mathbf{y}_1, \ldots, \mathbf{y}_l \in \mathbb{Z}_p^n$ and any matrix $\mathbf{M} \in \mathsf{GL}_n(\mathbb{Z}_p)$, the distributions $\big(\{\mathbf{x}_i\mathbf{B}\}_{i\in[k]}, \{\mathbf{y}_i\mathbf{B}^*\}_{i\in[l]}\big)$ and $\big(\{\mathbf{x}_i\mathbf{M}\mathbf{B}\}_{i\in[k]}, \{\mathbf{y}_i\mathbf{M}^*\mathbf{B}^*\}_{i\in[l]}\big)$ are identical.

Diffie-Hellman Assumption. Let $\tau \in \{1, 2\}$. Given an asymmetric bilinear map $\mathcal{G} \leftarrow \mathsf{GroupGen}(\vartheta)$, along with

$$[a]_\tau, [e]_\tau, [t_\beta]_\tau = [ae + \beta f]_\tau$$

where $a, e, f \xleftarrow{\text{U}} \mathbb{Z}_p$, the DDH$\tau$ problem asks to determine whether $\beta = 0$ or $\beta = 1$.
For a probabilistic polynnomial time adversary $\mathscr{A}$, define

$$\mathsf{Adv}_{\mathscr{A}}^{\mathcal{G}}(\text{DDH}\tau)(\vartheta) = \bigg| \Pr[\mathscr{A}(\mathcal{G}, [a]_\tau, [e]_\tau, [t_0]_\tau) = 1] - \Pr[\mathscr{A}(\mathcal{G}, [a]_\tau, [e]_\tau, [t_1]_\tau) = 1] \bigg|.$$

The decisional Diffie-Hellman assumption in group G_τ (DDHτ) assumption holds if for all PPT adversaries $\mathscr{A}$, $\mathsf{Adv}^{\mathcal{G}}_{\mathscr{A}}(\text{DDH}\tau)(\vartheta) \leq \mathsf{negl}(\vartheta)$.

The symmtric external Diffie-Hellman (SXDH) assumption is said to hold if both DDH1 and DDH2 hold.

2.3 Multi-Input IPFE with Variable Vector Size

Multi-Input Inner Product Functionality. A bounded-arity multi-input inner product function family $\mathcal{F}^{\mathcal{B}}_{\vartheta} = \{\mathcal{F}^{\mathcal{B}}_{n}\}$ for some $\mathcal{B} \in \mathbb{N}$, where each sub-families $\mathcal{F}^{\mathcal{B}}_{n}$ consists of bounded-arity multi-input inner product functions $f_{\{\mathbf{y}_\iota\}_{\iota\in[n]}}$. Each function $f_{\{\mathbf{y}_\iota\}_{\iota\in[n]}} : \mathbb{Z}^{m_1} \times \mathbb{Z}^{m_2} \times ... \times \mathbb{Z}^{m_n} \to \mathbb{Z}$, with the associated vectors $\{y_\iota\}_{\iota\in[n]}$ each belonging to $\mathbb{Z}^{m'_i}$, is defined as

$$f_{\{\mathbf{y}_\iota\}\iota\in[n]}(\mathbf{x}_1, \ldots, \mathbf{x}_n) = \sum_{\iota\in[n]} \langle \mathbf{x}_\iota, \mathbf{y}_\iota \rangle$$

for all sets of vectors $\{\mathbf{x}_\iota\}_{\iota\in[n]}$ of variable length over $\mathbb{Z}$ and the value of the inner product $\langle \mathbf{x}_\iota, \mathbf{y}_\iota \rangle \leq \mathcal{B}$. Since we work on the ct-dominant setting, we define the inner product $\langle \mathbf{x}_\iota, \mathbf{y}_\iota \rangle = \sum_{i\in m'_\iota} x_{\iota,i}.y_{\iota,i}$ assuming $[m'_\iota] \subseteq [m_\iota]$.

Private Key MIPFE over Variable-Length Vectors. A private key *bounded-arity* multi-input inner product function encryption scheme over variable-length vectors associated with function family $\mathcal{F}^{\mathcal{B}}_{n}$ is specified by the following polynomial-time algorithms.

Setup$(1^\vartheta, n, \mathcal{B})$**:** Takes the security parameter 1^ϑ, the arity $n \in \mathbb{N}$ for the multi-input inner product functionality, and the upper bound $\mathcal{B}$ on the values of inner products. It generates and outputs the master secret key msk and the public parameters pp.

KeyGen$(\mathsf{pp}, \mathsf{msk}, \{\mathbf{y}_\iota\}_{\iota\in[n]})$**:** It takes as input pp, msk and the set of vectors $\{\mathbf{y}_\iota\}_{\iota\in[n]}$ of variable lengths such that $\mathbf{y}_\iota \in \mathbb{Z}^{m'_\iota}$ for all $\iota \in [n]$. Finally, it outputs the decryption key sk corresponding to the given set of vectors.

Encrypt$(\mathsf{pp}, \mathsf{msk}, \iota, \mathbf{x}_\iota)$**:** Takes as input pp, the master secret msk, an index $\iota \in [n]$ and a vector $\mathbf{x}_\iota \in \mathbb{Z}^{m_\iota}$ for slot ι. It outputs the ciphertext ct_ι.

Decrypt$(\mathsf{pp}, \mathsf{sk}, \{\mathsf{ct}_\iota\}_{\iota\in[n]})$**:** Takes as input pp, a decryption key sk and set of n ciphertexts $\{\mathsf{ct}_\iota\}_{\iota\in[n]}$. It outputs $d \in \mathbb{Z}$ or special symbol $\perp$ to indicate failure.

Correctness. The above scheme is said to be *correct* if for all security parameters 1^ϑ, for all n polynomial in ϑ, for all sets of n vectors $\{\mathbf{x}_\iota\}_{\iota\in[n]}, \{\mathbf{y}_\iota\}_{\iota\in[n]}$ with $\langle \mathbf{x}_\iota, \mathbf{y}_\iota \rangle \leq \mathcal{B}$, we have

$$\Pr\left[d = \sum_{\iota\in[n]} \langle \mathbf{x}_\iota, \mathbf{y}_\iota \rangle \;\middle|\; \begin{array}{l} (\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\vartheta, n, \mathcal{B}) \\ \mathsf{sk} \leftarrow \mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, \{\mathbf{y}_\iota\}_{\iota\in[n]}) \\ \{\mathsf{ct}_\iota \leftarrow \mathsf{Encrypt}(\mathsf{pp}, \mathsf{msk}, \iota, \mathbf{x}_\iota)\}_{\iota\in[n]} \\ d \leftarrow \mathsf{Decrypt}(\mathsf{pp}, \mathsf{sk}, \{\mathsf{ct}_\iota\}_{\iota\in[n]}) \end{array} \right] \geq 1 - \mathsf{negl}(\vartheta)$$

for some negligible function negl.

Full-Hiding Security. The notion of full-hiding security for private key MIPFE arity n can be formalized through the following experiment $\mathsf{Expt}_{\mathscr{A}}(\beta)$, for $\beta \xleftarrow{\text{U}} \{0,1\}$, where $\mathscr{A}$ is the adversary and $\mathscr{C}$ is the challenger.

Setup: $\mathscr{C}$ generates $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^{\vartheta}, n, \mathcal{B})$ and passes the public parameter pp to $\mathscr{A}$. $\mathscr{C}$ generates $\beta \xleftarrow{\text{U}} \{0,1\}$.

Key Query Phase: $\mathscr{A}$ adaptively makes (polynomially many in ϑ) key extraction queries: for the j^{th} secret key query the $\mathscr{A}$ provides pair of vector sets $(\{\mathbf{y}_{j,\iota,0}\}_{\iota\in[n]}, \{\mathbf{y}_{j,\iota,1}\}_{\iota\in[n]})$ such that $\mathbf{y}_{j,\iota,0}, \mathbf{y}_{j,\iota,1} \in \mathbb{Z}^{m'_{j,\iota}}$; $\mathscr{C}$ then responds with the secret key $\mathsf{sk}^*_j \leftarrow \mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, \{\mathbf{y}_{j,\iota,\beta}\})$.

Ciphertext Query Phase: $\mathscr{A}$ makes adaptively makes a polynomial number of ciphertext queries. Each query consists of a pair of vectors $(\mathbf{x}_{\mu_\iota,\iota,0}, \mathbf{x}_{\mu_\iota,\iota,1}) \in (\mathbb{Z}^{m_{\mu_\iota,\iota}})^2$ for slot ι. In response to μ_ι ciphertext query numbered μ_ι, $\mathscr{C}$ returns $\mathsf{ct}^*_{\mu_\iota,\iota} \leftarrow \mathsf{Encrypt}(\mathsf{pp}, \mathsf{msk}, \mathbf{x}_{\mu_\iota,\iota,\beta})$. We assume that the total number of decryption key queries made by $\mathscr{A}$ is q_{sk} and the total number of ciphertext queries for index ι is $q_{\mathsf{ct},\iota}$ with the restriction that the number of queries for each index be atleast one i.e., $q_{\mathsf{ct},\iota} \geq 1$ for all $\iota \in [n]$. Also, For all $j \in [q_{\text{sk}}]$ and for all $(\mu_1, ..., \mu_n) \in [q_{\mathsf{ct},1}] \times ... \times [q_{\mathsf{ct},n}]$, we must have

$$\sum_{\iota\in[n]} \langle \mathbf{x}_{\mu_\iota,\iota,0}, \mathbf{y}_{j,\iota,0} \rangle = \sum_{\iota\in[n]} \langle \mathbf{x}_{\mu_\iota,\iota,1}, \mathbf{y}_{j,\iota,1} \rangle$$

Guess: $\mathscr{A}$ concludes the game with a guess $\beta' \in \{0,1\}$.

The MIPFE scheme is said to achieve fully hiding if for any PPT adversary $\mathscr{A}$ the advantage is,

$$\mathsf{Adv}^{\text{MIPFE}}_{\mathscr{A}}(\vartheta) = |\Pr[\mathsf{Expt}^{\text{MIPFE}}_{\mathscr{A}}(0) = 1] - \Pr[\mathsf{Expt}^{\text{MIPFE}}_{\mathscr{A}}(1) = 1]| \leq \mathsf{negl}(\vartheta)$$

for some negligible function negl.

3 Our Variable Vector Length MIPFE Scheme

3.1 Construction

$\mathsf{Setup}(1^{\vartheta})$**:** Takes a security parameter 1^{ϑ}, generates bilinear group $(\mathcal{G} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, P_1, P_2, e, p)) \leftarrow \mathsf{GroupGen}(1^{\vartheta})$, chooses $s_\iota \xleftarrow{\text{U}} \mathbb{Z}_p, \mathbf{B}_\iota \xleftarrow{\text{U}} \mathsf{GL}_7(\mathbb{Z}_p)$ $\forall \iota \in [n]$. And it sets,

$$\mathsf{pp} = \mathcal{G}, \mathsf{msk} = \{s_\iota, \mathbf{B}_\iota, \mathbf{B}^*_\iota\}_{\iota\in[n]}$$

$\mathsf{Encrypt}(\mathsf{pp}, \mathsf{msk}, \iota, \mathbf{x}_\iota = (x_{\iota,i})_{i\in[m_\iota]})$**:** The size of vector $\mathbf{x}_\iota$ is m_ι. Choose $\pi_{\iota,i}, z_\iota \xleftarrow{\text{U}} \mathbb{Z}_p$, $\forall i \in [m_\iota]$. Compute

$$\mathbf{c}_{\iota,i} = (\pi_{\iota,i}(i,1), s_\iota, x_{\iota,i}, z_\iota, 0, 0)\,\mathbf{B}_\iota, \;\; \forall i \in [m_\iota]$$

Output the ciphertext $\mathsf{ct}_\iota = (\iota, \{[\mathbf{c}_{\iota,i}]_1\}_{i\in m_\iota})$.

KeyGen$(\mathsf{pp}, \mathsf{msk}, \{D_\iota, y_\iota = (y_{\iota,i})_{i\in D_\iota}\}_{\iota\in[n]})$**:** On input a set of vectors $\mathbf{y}_\iota$ defined over index set D_ι for $\iota \in [n]$, choose $u_\iota, \rho_{\iota,i}, r_{\iota,i} \xleftarrow{\mathrm{U}} \mathbb{Z}_p,\ \forall\iota \in [n], i \in D_\iota$ such that $\sum_{i\in D_\iota} r_{\iota,i} = 0$. It sets

$$\mathbf{k}_{\iota,i} = (\rho_{\iota,i}(-1, i), u_\iota, y_{\iota,i}, r_{\iota,i}, 0, 0)\,\mathbf{B}^*_\iota,\ \ \forall\iota \in [n], i \in D_\iota$$
$$\hat{k} = e(P_1, P_2)^{-\sum_{\iota\in[n]}|D_\iota|u_\iota s_\iota}$$

where $|D_\iota|$ represents the cardinality of the domain D_ι. Output the secret key $\mathsf{sk} = (\hat{k}, \{D_\iota, \{[\mathbf{k}_{\iota,i}]_2\}_{i\in D_\iota}\}_{\iota\in[n]})$.

Decrypt$(\mathsf{pp}, \mathsf{sk}, \{\mathsf{ct}_\iota\}_{\iota\in[n]})$**:** If $D_\iota \subseteq [m_\iota], \forall\iota \in [n]$, then compute

$$h = \hat{k} \prod_{\iota\in[n]} \prod_{i\in D_\iota} e([\mathbf{c}_{\iota,i}]_1, [\mathbf{k}_{\iota,i}]_2)$$

Then compute and output the discrete logarithm of h to base $e(P_1, P_2)$.

Correctness. For any set of n ciphertexts $\{\mathsf{ct}_\iota = (\iota, \{[\mathbf{c}_{\iota,i}]_1\}_{i\in m_\iota})\}_{\iota\in[n]}$, and decryption key $\mathsf{sk} = (\hat{k}, \{D_\iota, \{[\mathbf{k}_{\iota,i}]_2\}_{i\in D_\iota}\}_{\iota\in[n]})$, if $D_\iota \subseteq [m_\iota]$ for all $\iota \in [n]$, we have

$$\begin{aligned}
h &= \hat{k} \prod_{\iota\in[n]} \prod_{i\in D_\iota} e([\mathbf{c}_{\iota,i}]_1, [\mathbf{k}_{\iota,i}]_2) \\
&= \hat{k} \prod_{\iota\in[n]} e(P_1, P_2)^{\sum_{i\in D_\iota}\langle \mathbf{c}_{\iota,i}, \mathbf{k}_{\iota,i}\rangle} \\
&= \hat{k} \prod_{\iota\in[n]} e(P_1, P_2)^{\sum_{i\in D_\iota} x_{\iota,i}y_{\iota,i}+u_\iota s_\iota+r_{\iota,i}z_\iota} \\
&= \hat{k} \prod_{\iota\in[n]} e(P_1, P_2)^{\sum_{i\in D_\iota} x_{\iota,i}y_{\iota,i}+u_\iota s_\iota} \\
&= \hat{k} \cdot e(P_1, P_2)^{\sum_{\iota\in[n]}\langle \mathbf{x}_\iota, \mathbf{y}_\iota\rangle + |D_\iota|u_\iota s_\iota} \\
&= e(P_1, P_2)^{-\sum_{\iota\in[n]}\left|D_\iota\right|u_\iota s_\iota} e(P_1, P_2)^{\sum_{\iota\in[n]}\langle \mathbf{x}_\iota, \mathbf{y}_\iota\rangle + \left|D_\iota\right|u_\iota s_\iota} \\
&= e(P_1, P_2)^{\sum_{\iota\in[n]}\langle \mathbf{x}_\iota, \mathbf{y}_\iota\rangle}
\end{aligned}$$

Computing dicrete logarithm produces the desired result which is $\sum_{\iota\in[n]}\langle \mathbf{x}_\iota, \mathbf{y}_\iota\rangle$. Given that $\sum_{\iota\in[n]}\langle \mathbf{x}_\iota, \mathbf{y}_\iota\rangle \leq n\mathcal{B}$ which is polynomial in ϑ, it is feasible to compute discete logarithm of h to base $e(P_1, P_2)$.

3.2 Proof of Security

Theorem 1. *Our* MIPFE *scheme is fully hiding under the restriction that the adversary makes atleast one ciphertext query for each slot provided the* SXDH *assumption holds in the underlying pairing groups. More formally, for any PPT* *adversary $\mathscr{A}$ against our* MIPFE *scheme, there exist a PPT adversary $\mathscr{B}$ for* SXDH *such that*

$$\mathsf{Adv}_{\mathscr{A}}^{\mathrm{MIPFE}}(\vartheta) \leq \left(2q_{\mathsf{sk}} + 3\sum_{\iota\in[n]} q_{\mathsf{ct},\iota}\right)\mathsf{Adv}_{\mathscr{B}}^{\mathrm{SXDH}}(\vartheta) + 2^{-\Omega(\vartheta)}$$

Proof. The proof starts with $\mathsf{Expt}_{\mathscr{A}}^{\mathrm{MIPFE}}(0)$ which is game 0 and ends with $\mathsf{Expt}_{\mathscr{A}}^{\mathrm{MIPFE}}(1)$. In the intermediate games, we change the ciphertext and secret key from $\beta = 0$ to $\beta = 1$ using both computational and information theoretic arguments. We assume that the adversary $\mathscr{A}$ makes atleast one ciphertext query for each slot ι. The sequence of games is as follows.

Game 0**:** This game is same as the real security game when $\beta = 0$. For all $\iota \in [n], \mu_\iota \in [q_{\mathsf{ct},\iota}]$, in response to μ_ι^{th} ciphertext query for the slot ι with a pair of vectors $(\mathbf{x}_{\mu_\iota,\iota,0}, \mathbf{x}_{\mu_\iota,\iota,1})$ of same length $m_{\mu_\iota,\iota}$, the challenger returns the ciphertext $\mathsf{ct}_{\mu_\iota,\iota} = (\iota, \left\{[\mathbf{c}_{\mu_\iota,\iota,i}]_1\right\}_{i\in[m_{\mu_\iota,\iota}]})$ where

$$\mathbf{c}_{\mu_\iota,\iota,i} = (\pi_{\mu_\iota,\iota,i}(i,1), s_\iota, x_{\mu_\iota,\iota,0,i}, z_{\mu_\iota,\iota}, 0, 0)\mathbf{B}_\iota,\ \ \forall i \in [m_{\mu_\iota,\iota}]$$

and for all $j \in [q_{\mathrm{sk}}]$, the j^{th} secret key query with two sets of n vectors each, $(\left\{\mathbf{y}_{j,\iota,0}\right\}_{\iota\in[n]}, \left\{\mathbf{y}_{j,\iota,1}\right\}_{\iota\in[n]})$ with vectors for slot ι defined over domain $D_{j,\iota}$, the adversary is provided $\mathsf{sk}_j = (\hat{k}_j, \left(D_{j,\iota}, \{[\mathbf{k}_{j,\iota,i}]_2\}_{i\in D_{j,\iota}}\right)_{\iota\in[n]})$, where

$$\mathbf{k}_{j,\iota,i} = (\rho_{j,\iota,i}(-1,i), u_{j,\iota}, y_{j,\iota,0,i}, r_{j,\iota,i}, 0, 0)\mathbf{B}_\iota^*, \forall\iota \in [n], i \in D_{j,\iota}$$
$$\hat{k}_j = e(P_1, P_2)^{-\sum_{\iota\in[n]}\left|D_{j,\iota}\right|s_\iota u_{j,\iota}}$$

where $\sum_{i\in D_{j,\iota}} r_{j,\iota,i} = 0$.

Game 1**-**v**-**1**,** $v \in [q_{\mathbf{sk}}]$**:** Game 1-0-2 is same as Game 0. This game is same as Game 1-$(v-1)$-2, except that the v-th secret key query is responded as

$$\mathbf{k}_{v,\iota,i} = (\rho_{v,\iota,i}(-1,i), u_{v,\iota}, y_{v,\iota,0,i}, r_{v,\iota,i}, 0, \boxed{\bar{r}_{v,\iota,i}})\mathbf{B}_\iota^*, \forall\iota \in [n], i \in D_{v,\iota}$$
$$\hat{k}_v = e(P_1, P_2)^{-\sum_{\iota\in[n]}\left|D_{v,\iota}\right|s_\iota u_{v,\iota}}$$

where $\sum_{i\in D_{v,\iota}} \bar{r}_{v,\iota,i} = 0$.

Game 1**-**v**-**2**,** $v \in [q_{\mathbf{sk}}]$**:** This game is same as Game 1-v-1 except that the v-th secret key query is responded as

$$\mathbf{k}_{v,\iota,i} = (\rho_{v,\iota,i}(-1,i), u_{v,\iota}, y_{v,\iota,0,i}, r_{v,\iota,i}, \boxed{y_{v,\iota,1,i}}, \bar{r}_{v,\iota,i})\mathbf{B}_\iota^*, \forall\iota \in [n], i \in D_{v,\iota}$$
$$\hat{k}_v = e(P_1, P_2)^{-\sum_{\iota\in[n]}\left|D_{v,\iota}\right|s_\iota u_{v,\iota}}$$

Game 2-ι-μ_ι-1, $\iota \in [n], \mu_\iota \in [q_{\mathsf{ct},\iota}]$**:** Game 2-0-$q_{\mathsf{ct},0}$-3 is same as Game 1-$q_{\mathrm{sk}}$-2. This game is same as Game 2-$(\iota-1)$-$q_{\mathsf{ct},\iota-1}$-3 if $\mu_\iota = 1$ or Game 2-ι-$(\mu_\iota - 1)$-3 if $\mu_\iota > 1$ except that the μ_ι-th ciphertext query for ι-th slot corresponding to pair of vectors $(\mathbf{x}_{\mu_\iota,\iota,0}, \mathbf{x}_{\mu_\iota,\iota,1})$ with same length $m_{\mu_\iota,\iota}$ is responded with $\mathsf{ct}_{\mu_\iota,\iota} = (\iota, \{[\mathbf{c}_{\mu_\iota,\iota,i}]_1\}_{i \in m_{\mu_\iota,\iota}})$, where

$$\mathbf{c}_{\mu_\iota,\iota,i} = (\pi_{\mu_\iota,\iota,i}(i,1), s_\iota, x_{\mu_\iota,\iota,0,i}, z_{\mu_\iota,\iota}, 0, \boxed{z'_{\mu_\iota,\iota}})\mathbf{B}_\iota, \ \ \forall i \in [m_{\mu_\iota,\iota}]$$

Here, the random scalars are chosen as in Game 2-$(\iota-1)$-$q_{ct,\iota-1}$-3 or Game 2-ι-$(\mu_\iota - 1)$-3 according as $\mu_\iota = 1$ or $\mu_\iota > 1$.

Game 2-ι-μ_ι-2, $\iota \in [n], \mu_\iota \in [q_{\mathsf{ct},\iota}]$**:** This game is same as Game 2-ι-μ_ι-1 except that

$$\mathbf{c}_{\mu_\iota,\iota,i} = (\pi_{\mu_\iota,\iota,i}(i,1), s_\iota, \boxed{0}, z_{\mu_\iota,\iota}, \boxed{x_{\mu_\iota,\iota,1,i}}, z'_{\mu_\iota,\iota})\mathbf{B}_\iota, \ \ \forall i \in [m_{\mu_\iota,\iota}]$$

Game 2-ι-μ_ι-3, $\iota \in [n], \mu_\iota \in [q_{\mathsf{ct},\iota}]$**:** This game is same as Game 2-ι-μ_ι-2 except that

$$\mathbf{c}_{\mu_\iota,\iota,i} = (\pi_{\mu_\iota,\iota,i}(i,1), s_\iota, 0, z_{\mu_\iota,\iota}, x_{\mu_\iota,\iota,1,i}, \boxed{0})\mathbf{B}_\iota, \ \ \forall i \in [m_{\mu_\iota,\iota}]$$

Game 3: This game is same as Game 2-n-$q_{\mathsf{ct},n}$-3 except that all the ciphertext and secret key are responded as

$$\mathbf{c}_{\mu_\iota,\iota,i} = (\pi_{\mu_\iota,\iota,i}(i,1), s_\iota, \boxed{x_{\mu_\iota,\iota,1,i}}, z_{\mu_\iota,\iota}, \boxed{0}, 0)\mathbf{B}_\iota, \ \ \forall i \in [m_{\mu_\iota,\iota}]$$
$$\mathbf{k}_{j,\iota,i} = (\rho_{v,\iota,i}(-1,i), u_{j,\iota}, \boxed{y_{j,\iota,1,i}}, r_{j,\iota,i}, \boxed{y_{j,\iota,0,i}}, \bar{r}_{j,\iota,i})\mathbf{B}^*_\iota, \ \ \forall \iota \in [n], i \in D_{j,\iota}$$
$$\hat{k}_j = e(P_1, P_2)^{-\sum_{\iota \in [n]} \left|D_{j,\iota}\right| s_\iota u_{j,\iota}}$$

Game 4: This game is same as the real security game when $\beta = 1$. For all $\iota \in [n], \mu_\iota \in [q_{\mathsf{ct},\iota}]$, in response to μ_ι^{th} ciphertext query for the slot ι with a pair of vectors $(\mathbf{x}_{\mu_{\iota,0},\iota}, \mathbf{x}_{\mu_\iota,\iota,1})$ of same length $m_{\mu_\iota,\iota}$, it returns the ciphertext $\mathsf{ct}_{\mu_\iota,\iota} = (\iota, \left\{[\mathbf{c}_{\mu_\iota,\iota,i}]_1\right\}_{i \in m_{\mu_\iota,\iota}})$ where

$$\mathbf{c}_{\mu_\iota,\iota,i} = (\pi_{t_\iota,\iota,i}(i,1), s_\iota, x_{\mu_\iota,\iota,1,i}, z_{\mu_\iota,\iota}, 0, 0)\mathbf{B}_\iota, \ \ \forall i \in [m_{\mu_\iota,\iota}]$$

and for all $j \in [q_{\mathrm{sk}}]$, the j^{th} secret key query on two sets of n vectors each $\left(\left\{\mathbf{y}_{j,\iota,0}\right\}_{\iota \in [n]}, \left\{\mathbf{y}_{j,\iota,1}\right\}_{\iota \in [n]}\right)$ defined over domain $D_{j,\iota}$ for slot ι, is responded with $\mathsf{sk}_j = (\hat{k}_j, \left(D_{j,\iota}, \{[\mathbf{k}_{j,\iota,i}]_2\}_{i \in D_{j,\iota}}\right)_{\iota \in [n]})$, where

$$\mathbf{k}_{j,\iota,i} = (\rho_{j,\iota,i}(-1,i), u_{j,\iota}, y_{j,\iota,1,i}, r_{j,\iota,i}, \boxed{0, 0})\mathbf{B}^*_\iota, \ \ \forall \iota \in [n], i \in D_{j,\iota}$$
$$\hat{k}_j = e(P_1, P_2)^{-\sum_{\iota \in [n]} \left|D_{j,\iota}\right| s_\iota u_{j,\iota}}$$

where $\sum_{i \in D_{j,\iota}} r_{j,\iota,i} = 0$.

Let E_X denote the probability that the adversary $\mathscr{A}$ wins in game X.

Lemma 1. *There exists a PPT algorithm* $\mathscr{B}$ *for* DDH2 *such that*

$$\left|\Pr[\mathsf{E}_{1\text{-}(v-1)\text{-}2}] - \Pr[\mathsf{E}_{1\text{-}v\text{-}1}]\right| \leq \mathsf{Adv}_{\mathscr{B}}^{\mathrm{DDH2}}(\vartheta) + 2^{-\Omega(\vartheta)}$$

Proof. Adversary $\mathscr{B}$ receives an instance of DDH i.e., $(\mathcal{G}, [a]_2, [e]_2, [t_\beta]_2)$, and it sets $\mathsf{pp} = \mathcal{G}$. It samples $\mathbf{W}_\iota \xleftarrow{\mathrm{U}} \mathsf{GL}_7(\mathbb{Z}_p),\ \forall \iota \in [n]$. It sets,

$$\mathbf{B}_\iota = \begin{pmatrix} \mathbf{I}_4 & & & \\ & 0 & 0 & 1 \\ & 0 & 1 & 0 \\ & 1 & 0 & -a \end{pmatrix} \mathbf{W}_\iota, \quad \mathbf{B}_\iota^* = \begin{pmatrix} \mathbf{I}_4 & & & \\ & a & 0 & 1 \\ & 0 & 1 & 0 \\ & 1 & 0 & 0 \end{pmatrix} \mathbf{W}_\iota^*, \forall \iota \in [n]$$

Then $\mathscr{B}$ simulates the ciphertext and secret key query in the following manner. All the ciphertext query is respond as follows:

$$[\mathbf{c}_{\mu_\iota,\iota,i}]_1 = [(\pi_{\mu_\iota,\iota,i}(i,1), s_\iota, x_{\mu_\iota,\iota,0,i}, z_{\mu_\iota,\iota}, 0, 0)\mathbf{B}_\iota]_1,\ \ \forall i \in [m_{\mu_\iota,\iota}]$$

And the secret key query is responded with $\hat{k}_j = e(P_1, P_2)^{-\sum_{\iota\in[n]} \left|D_{j,\iota}\right| s_\iota u_{j,\iota}}$ for all $j \in [q_{\mathrm{sk}}]$, and

$$[\mathbf{k}_{j,\iota,i}]_2 = \begin{cases} [(\rho_{j,\iota,i}(-1,i), u_{j,\iota}, y_{j,\iota,0,i}, r_{j,\iota,i}, y_{j,\iota,1,i}, \bar{r}_{j,\iota,i})\mathbf{B}_\iota^*]_2,\ \ \forall \in D_{j,\iota}, & (j < v) \\ [(\rho_{j,\iota,i}(-1,i), u_{j,\iota}, y_{j,\iota,0,i}, r_{j,\iota,i}, 0, 0)\mathbf{B}_\iota^*]_2,\ \ \forall \in D_{j,\iota}, & (j > v) \end{cases}$$

Now, for the v-th secret key query, $\mathscr{B}$ responds as follows:

$$r'_{v,\iota,i} \xleftarrow{\mathrm{U}} \mathbb{Z}_p \text{ such that } \sum_{i\in D_{v,\iota}} r'_{v,\iota,i} = 0$$

$$\begin{aligned} [\mathbf{k}_{v,\iota,i}]_2 &= [(\rho_{v,\iota,i}(-1,i), u_{v,\iota}, y_{v,\iota,0,i}, 0, 0, 0)\mathbf{B}_\iota^* + r'_{v,\iota,i}(0,0,0,0,t_\beta,0,e)\mathbf{W}_\iota^*]_2 \\ &= [(\rho_{v,\iota,i}(-1,i), u_{v,\iota}, y_{v,\iota,0,i}, er'_{v,\iota,i}, 0, \beta f r'_{v,\iota,i})\mathbf{B}_\iota^*]_2 \end{aligned}$$

implicitly setting $r_{v,\iota,i} = er'_{v,\iota,i}$. Now, $\mathscr{A}$'s view is same as Game 1-$(v-1)$-2 if $\beta = 0$, otherwise it is Game 1-v-1 with $\bar{r}_{v,\iota,i} = fr'_{v,\iota,i}$.

Lemma 2. $|\Pr[\mathsf{E}_{1\text{-}v\text{-}1}] - \Pr[\mathsf{E}_{1\text{-}v\text{-}2}]| \leq 2^{-\Omega(\vartheta)}$.

Proof. We choose $\mathbf{B}_\iota \xleftarrow{\mathrm{U}} \mathsf{GL}_7(\mathbb{Z}_p), \forall \iota \in [n]$. Now, all the ciphertext query are responded as:

$$[\mathbf{c}_{\mu_\iota,\iota,i}]_1 = [(\pi_{\mu_\iota,\iota,i}(i,1), s_\iota, x_{\mu_\iota,\iota,0,i}, z_{\mu_\iota,\iota}, 0, 0)\mathbf{B}_\iota]_1,\ \ \forall i \in [m_{\mu_\iota,\iota}]$$

And the secret key query is responded with $\hat{k}_j = e(P_1, P_2)^{-\sum_{\iota\in[n]} \left|D_{j,\iota}\right| s_\iota u_{j,\iota}}$ for all $j \in [q_{sk}]$, and

$$[\mathbf{k}_{j,\iota,i}]_2 = \begin{cases} [(\rho_{j,\iota,i}(-1,i), u_{j,\iota}, y_{j,\iota,0,i}, r_{j,\iota,i}, y_{j,\iota,1,i}, \bar{r}_{j,\iota,i})\mathbf{B}_\iota^*]_2,\ \ \forall \in D_{j,\iota}, & (j < v) \\ [(\rho_{j,\iota,i}(-1,i), u_{j,\iota}, y_{j,\iota,0,i}, r_{j,\iota,i}, 0, 0)\mathbf{B}_\iota^*]_2,\ \ \forall \in D_{j,\iota}, & (j > v) \end{cases}$$

Now, for the v-th secret key query, we sample $w \xleftarrow{\mathrm{U}} \{0,1\}$, and set

$$[\mathbf{k}_{v,\iota,i}]_2 = [(\rho_{v,\iota,i}(-1,i), u_{v,\iota}, y_{v,\iota,0,i}, r_{j,\iota,i}, wy_{v,\iota,1,i}, \bar{r}_{j,\iota,i})\mathbf{B}_\iota^*]_2$$

$\mathscr{A}$'s view is same as Game 1-v-1 if $w = 0$, otherwise it is Game 1-v-2.

Lemma 3. *There exist a PPT adversary* $\mathscr{B}$ *for* DDH1 *such that* $\forall\iota \in [n]$,

$$\begin{aligned}\big|\Pr[\mathsf{E}_{2\text{-}\iota\text{-}\mu_\iota\text{-}1}] - \Pr[\mathsf{E}_{2\text{-}(\iota-1)\text{-}q_{\mathsf{ct},\iota-1}\text{-}3}]\big| &\le \mathsf{Adv}_{\mathscr{B}}^{\mathrm{DDH1}}(\vartheta) + 2^{-\Omega(\vartheta)}, \text{if } \mu_\iota = 1\\ \big|\Pr[\mathsf{E}_{2\text{-}\iota\text{-}\mu_\iota\text{-}1}] - \Pr[\mathsf{E}_{2\text{-}\iota\text{-}(\mu_\iota-1)\text{-}3}]\big| &\le \mathsf{Adv}_{\mathscr{B}}^{\mathrm{DDH1}}(\vartheta) + 2^{-\Omega(\vartheta)}, \text{if } \mu_\iota > 1\end{aligned}$$

Proof. Adversary $\mathscr{B}$ receives an instance of DDH1 $(\mathcal{G}, [a]_1, [e]_1, [t_\beta]_1)$ and it sets $\mathsf{pp} = \mathcal{G}$. It samples $\mathbf{W}_\iota \xleftarrow{\mathrm{U}} \mathsf{GL}_7(\mathbb{Z}_p), \forall\iota \in [n]$ and defines

$$\mathbf{B}_\iota = \begin{pmatrix}\mathbf{I}_4 & \\ & \begin{matrix} a & 0 & 1\\ 0 & 1 & 0\\ 1 & 0 & 0\end{matrix}\end{pmatrix}\mathbf{W}_\iota, \quad \mathbf{B}^*_\iota = \begin{pmatrix}\mathbf{I}_4 & \\ & \begin{matrix} 0 & 0 & 1\\ 0 & 1 & 0\\ 1 & 0 & -a\end{matrix}\end{pmatrix}\mathbf{W}^*_\iota, \forall\iota \in [n].$$

All the secret key query are responded as,

$$r'_{j,\iota,i}, r''_{j,\iota,i} \xleftarrow{\mathrm{U}} \mathbb{Z}_p, \text{such that} \sum_{i \in D_{j,\iota}} r'_{j,\iota,i} = \sum_{i \in D_{j,\iota}} r''_{j,\iota,i} = 0$$

$$\begin{aligned}[\mathbf{k}_{j,\iota,i}]_2 &= [(\rho_{j,\iota,i}(-1, i), u_{j,\iota}, y_{j,\iota,0,i}, r'_{j,\iota,i}, y_{j,\iota,1,i}, 0)\mathbf{B}^*_\iota + (0,0,0,0,r''_{j,\iota,i},0,0)\mathbf{W}^*_\iota]_2\\ &= [(\rho_{j,\iota,i}(-1, i), u_{j,\iota}, y_{j,\iota,0,i}, r'_{j,\iota,i} + ar''_{j,\iota,i}, y_{j,\iota,1,i}, r''_{j,\iota,i})\mathbf{B}^*_\iota]_2\end{aligned}$$

And sets $\hat{k}_j = e(P_1, P_2)^{-\sum_{\iota\in[n]} \left|D_{j,\iota}\right| s_\iota u_{j,\iota}}$. We can implicitly set $r_{j,\iota,i} = r'_{j,\iota,i} + ar''_{j,\iota,i}$ and $\bar{r}_{j,\iota,i} = r''_{j,\iota,i}$.
Now, all the ciphertext queries are responded as follows,

(i) If $(\iota', q_\iota) < (\iota, u_\iota)$, then the ciphertext is returned as,

$$[\mathbf{c}_{q_\iota,\iota',i}]_1 = [(\pi_{q_\iota,\iota',i}(i,1), s_\iota, 0, z_{q_\iota,\iota'}, x_{q_\iota,\iota',1,i}, 0)\mathbf{B}_{\iota'}]_1, \ \forall i \in [m_{q_\iota,\iota'}]$$

(ii) If $(\iota', q_\iota) = (\iota, u_\iota)$, then $\forall i \in [m_{u_\iota,\iota}]$ the ciphertext is returned as,

$$\begin{aligned}[\mathbf{c}_{u_\iota,\iota,i}]_1 &= [(\pi_{u_\iota,\iota,i}(i,1), s_\iota, x_{u_\iota,\iota,0,i}, 0, 0, 0)\mathbf{B}_\iota + z'_{u_\iota,\iota}(0,0,0,0,t_\beta,0,e)\mathbf{W}_\iota]_1\\ &= [(\pi_{u_\iota,\iota,i}(i,1), s_\iota, x_{u_\iota,\iota,0,i}, ez'_{u_\iota,\iota}, 0, \beta f z'_{u_\iota,\iota})\mathbf{B}_\iota]_1\end{aligned}$$

(iii) If $(\iota', q_\iota) > (\iota, u_\iota)$, the ciphertext is returned as,

$$[\mathbf{c}_{q_\iota,\iota',i}]_1 = [(\pi_{q_\iota,\iota',i}(i,1), s_\iota, x_{q_\iota,\iota',0,i}, z_{q_\iota,\iota'}, 0, 0)\mathbf{B}_{\iota'}]_1, \ \forall i \in [m_{q_\iota,\iota'}]$$

Adversary $\mathscr{B}$ perfectly simulated the secret key and ciphertext queries. View of adversary $\mathscr{A}$ is equally distributed between Game 2-ι-μ_ι-1 and Game 2-$(\iota-1)$-$q_{\mathsf{ct},\iota-1}$-3 or Game 2-ι-$(\mu_\iota-1)$-3, depending on $\mu_\iota = 1$ or $\mu_\iota > 1$ according as $\beta = 0$ or $\beta = 1$.

Lemma 4. $\big|\Pr[\mathsf{E}_{2\text{-}\iota\text{-}\mu_\iota\text{-}1}] - \Pr[\mathsf{E}_{2\text{-}\iota\text{-}\mu_\iota\text{-}2}]\big| \le 2^{-\Omega(\vartheta)}$.

Proof. We choose $\mathbf{B}_\iota \xleftarrow{\text{U}} \mathsf{GL}_7(\mathbb{Z}_p), \forall \iota \in [n]$. Now the j-th secret key is responded as,

$$\begin{aligned} \mathbf{k}_{j,\iota,i} &= (\rho_{j,\iota,i}(-1,i), u_{j,\iota}, y_{j,\iota,0,i}, r_{j,\iota,i}, y_{j,\iota,1,i}, \bar{r}_{j,\iota,i})\mathbf{B}^*_\iota, \ \ \forall \iota \in [n], i \in D_{j,\iota} \\ \hat{k}_j &= e(P_1, P_2)^{-\sum_{\iota \in [n]} \left|D_{j,\iota}\right| s_\iota u_{j,\iota}} \end{aligned}$$

And for the μ_ι-th ciphertext query, we sample $\hat{w} \xleftarrow{\text{U}} \{0,1\}$ and set

$$[\mathbf{c}_{\mu_\iota,\iota,i}]_1 = [(\pi_{\mu_\iota,\iota,i}(i,1), s_\iota, (1-\hat{w})x_{\mu_\iota,\iota,0,i}, z_{\mu_\iota,\iota}, \hat{w}x_{\mu_\iota,\iota,1,i}, z'_{\mu_\iota,\iota})\mathbf{B}_\iota]_1, \ \ \forall i \in [m_{\mu_\iota,\iota}]$$

$\mathscr{A}$'s view is same as Game 2-ι-μ_ι-1 if $\hat{w} = 0$, otherwise it is Game 2-ι-μ_ι-2.

Lemma 5. *There exists a PPT adversary $\mathscr{B}$ for DDH1 s.t.*

$$\left| \Pr[\mathsf{E}_{2\text{-}\iota\text{-}\mu_\iota\text{-}2}] - \Pr[\mathsf{E}_{2\text{-}\iota\text{-}\mu_\iota\text{-}3}] \right| \leq \mathsf{Adv}^{\text{SXDH}}_{\mathscr{B}}(\vartheta) + 2^{-\Omega(\vartheta)}$$

Proof. $\mathscr{B}$ receives an instance of DDH1 $(\mathcal{G}, [a]_1, [e]_1, [t_\beta]_1)$ and it sets $\mathsf{pp} = \mathcal{G}$. It samples $\mathbf{W}_\iota \xleftarrow{\text{U}} \mathsf{GL}_7(\mathbb{Z}_p), \forall \iota \in [n]$ and defines

$$\mathbf{B}_\iota = \begin{pmatrix} \mathbf{I}_4 & & & \\ & a & 0 & 1 \\ & 0 & 1 & 0 \\ & 1 & 0 & 0 \end{pmatrix} \mathbf{W}_\iota, \quad \mathbf{B}^*_\iota = \begin{pmatrix} \mathbf{I}_4 & & & \\ & 0 & 0 & 1 \\ & 0 & 1 & 0 \\ & 1 & 0 & -a \end{pmatrix} \mathbf{W}^*_\iota, \forall \iota \in [n]$$

All the secret key query are responded as,

$$r'_{j,\iota,i}, r''_{j,\iota,i} \leftarrow \mathbb{Z}_p, \text{such that} \sum_{i \in D_{j,\iota}} r'_{j,\iota,i} = \sum_{i \in D_{j,\iota}} r''_{j,\iota,i} = 0$$

$$\begin{aligned} [\mathbf{k}_{j,\iota,i}]_2 &= [(\rho_{j,\iota,i}(-1,i), u_{j,\iota}, y_{j,\iota,0,i}, r'_{j,\iota,i}, y_{j,\iota,1,i}, 0)\mathbf{B}^*_\iota + (0,0,0,0,r''_{j,\iota,i},0,0)\mathbf{W}^*_\iota]_2 \\ &= [(\rho_{j,\iota,i}(-1,i), u_{j,\iota}, y_{j,\iota,0,i}, r'_{j,\iota,i} + ar''_{j,\iota,i}, y_{j,\iota,1,i}, r''_{j,\iota,i})\mathbf{B}^*_\iota]_2 \end{aligned}$$

And sets $\hat{k}_j = e(P_1, P_2)^{-\sum_{\iota \in [n]} \left|D_{j,\iota}\right| s_\iota u_{j,\iota}}$. We can implicitly set $r_{j,\iota,i} = r'_{j,\iota,i} + ar''_{j,\iota,i}$ and $\bar{r}_{j,\iota,i} = r''_{j,\iota,i}$.
Now, all the ciphertext queries are responded as follows,

$$\begin{aligned} [\mathbf{c}_{u_\iota,\iota,i}]_1 &= [(\pi_{u_\iota,\iota,i}(i,1), s_\iota, 0, 0, x_{u_{\iota,\iota,1,i}}, 0)\mathbf{B}_\iota + \bar{z}_{u_\iota,\iota}(0,0,0,0,t_\beta,0,e)\mathbf{W}_\iota]_1 \\ &= [(\pi_{u_\iota,\iota,i}(i,1), s_\iota, 0, e\bar{z}_{u_\iota,\iota}, x_{u_{\iota,\iota,1,i}}, \beta f \bar{z}_{u_\iota,\iota})\mathbf{B}_\iota]_1 \end{aligned}$$

implicitly setting $z_{\mu_\iota,\iota} = e\bar{z}_{u_\iota,\iota}$. Now $\mathscr{A}'s$ view is same as Game 2-ι-μ_ι-2 if $\beta = 1$ implicitly setting $z'_{u_\iota,\iota} = \beta f \bar{z}_{u_\iota,\iota}$; otherwise it is Game 2-ι-μ_ι-3.

Lemma 6. $\Pr[\mathsf{E}_{2\text{-}n\text{-}q_{ct,n}\text{-}3}] = \Pr[\mathsf{E}_3]$.

Proof. We choose $\mathbf{B}_\iota \xleftarrow{\text{U}} \mathsf{GL}_7(\mathbb{Z}_p), \forall \iota \in [n]$. And define,

$$\mathbf{W}_\iota = \begin{pmatrix} \mathbf{I}_3 & & & & \\ & & & 1 & \\ & & 1 & & \\ & 1 & & & \\ & & & & 1 \end{pmatrix} \mathbf{B}_\iota, \quad \mathbf{W}^*_\iota = \begin{pmatrix} \mathbf{I}_3 & & & & \\ & & & 1 & \\ & & 1 & & \\ & 1 & & & \\ & & & & 1 \end{pmatrix} \mathbf{B}^*_\iota, \forall \iota \in [n]$$

Then all the ciphertext and secret key query are responded as,

$$\begin{aligned}
\mathbf{c}_{\mu_\iota,\iota,i} &= (\pi_{\mu_\iota,\iota,i}(i,1), s_\iota, 0, z_{\mu_\iota,\iota}, x_{\mu_\iota,\iota,1,i}, 0)\mathbf{B}_\iota \\
&= (\pi_{\mu_\iota,\iota,i}(i,1), s_\iota, x_{\mu_\iota,\iota,1,i}, z_{\mu_\iota,\iota}, 0, 0)\mathbf{W}_\iota \\
\mathbf{k}_{j,\iota,i} &= (\rho_{j,\iota,i}(-1,i), u_{j,\iota}, y_{j,\iota,0,i}, r_{j,\iota,i}, y_{j,\iota,1,i}, \bar{r}_{j,\iota,i})\mathbf{B}^*_\iota \\
&= (\rho_{j,\iota,i}(-1,i), u_{j,\iota}, y_{j,\iota,1,i}, r_{j,\iota,i}, y_{j,\iota,0,i}, \bar{r}_{j,\iota,i})\mathbf{W}^*_\iota
\end{aligned}$$

So, $\mathscr{A}$'s view is identical to both the games.

Lemma 7. *For any* PPT *adversary $\mathscr{A}$, there exists a* PPT *adversary $\mathscr{B}$ for SXDH s.t.*

$$\left|\Pr[\mathrm{E}_3] - \Pr[\mathrm{E}_4]\right| \leq q_{\mathrm{sk}}\mathrm{Adv}^{\mathrm{SXDH}}_{\mathscr{B}} + 2^{-\Omega(\vartheta)}$$

Proof. The proof is done the same way as Game 0 to Game 1-q_{sk}-2 but in the reverse order.

4 Conclusion

In this work, we have proposed a construction of private-key MIPFE with an apriori bounded slot size that can handle variable-length vectors at each slot. We proved security under the SXDH assumption. It would be interesting to further explore whether there exist constructions of MIPFE that can withstand an *arbitrary* number of encryption slots with variable-length vectors at each slot.

Acknowledgments

The first author expresses thanks to University Grants Commission (UGC) for their support.

References

1. Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In *Theory of Cryptography Conference*, pages 253–273. Springer, 2011.
2. Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: a new vision for public-key cryptography. *Communications of the ACM*, 55(11):56–64, 2012.
3. Adam O'Neill. Definitional issues in functional encryption. *Cryptology ePrint Archive*, 2010.
4. Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In *Annual cryptology conference*, pages 191–208. Springer, 2010.
5. Shweta Agrawal, Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption: New perspectives and lower bounds. In *Annual Cryptology Conference*, pages 500–518. Springer, 2013.

6. Shafi Goldwasser, S Dov Gordon, Vipul Goyal, Abhishek Jain, Jonathan Katz, Feng-Hao Liu, Amit Sahai, Elaine Shi, and Hong-Sheng Zhou. Multi-input functional encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 578–602. Springer, 2014.
7. Dan Boneh, Kevin Lewi, Mariana Raykova, Amit Sahai, Mark Zhandry, and Joe Zimmerman. Semantically secure order-revealing encryption: Multi-input functional encryption without obfuscation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 563–594. Springer, 2015.
8. Saikrishna Badrinarayanan, Divya Gupta, Abhishek Jain, and Amit Sahai. Multi-input functional encryption for unbounded arity functions. In *Advances in Cryptology–ASIACRYPT 2015: 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29–December 3, 2015, Proceedings, Part I*, pages 27–51. Springer, 2016.
9. Zvika Brakerski, Ilan Komargodski, and Gil Segev. Multi-input functional encryption in the private-key setting: Stronger security from weaker assumptions. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 852–880. Springer, 2016.
10. Junichi Tomida. Tightly secure inner product functional encryption: Multi-input and function-hiding constructions. *Theor. Comput. Sci.*, 833:56–86, 2020.
11. Michel Abdalla, Fabrice Benhamouda, and Romain Gay. From single-input to multi-client inner-product functional encryption. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part III*, volume 11923 of *Lecture Notes in Computer Science*, pages 552–582. Springer, 2019.
12. Michel Abdalla, Romain Gay, Mariana Raykova, and Hoeteck Wee. Multi-input inner-product functional encryption from pairings. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 601–626. Springer, 2017.
13. Michel Abdalla, Dario Catalano, Dario Fiore, Romain Gay, and Bogdan Ursu. Multi-input functional encryption for inner products: Function-hiding realizations and constructions without pairings. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 597–627. Springer, 2018.
14. Pratish Datta, Tatsuaki Okamoto, and Junichi Tomida. Full-hiding (unbounded) multi-input inner product functional encryption from the k-linear assumption. *IACR Cryptol. ePrint Arch.*, page 61, 2018.
15. Huijia Lin. Indistinguishability obfuscation from sxdh on 5-linear maps and locality-5 prgs. Cryptology ePrint Archive, Paper 2016/1096, 2016. `https://eprint.iacr.org/2016/1096`.
16. Junichi Tomida and Katsuyuki Takashima. Unbounded inner product functional encryption from bilinear maps. *Japan Journal of Industrial and Applied Mathematics*, 37(3):723–779, 2020.