Unbounded Inner Product Functional Encryption from Bilinear Maps

Junichi Tomida[1] and Katsuyuki Takashima[2]

[1] NTT
tomida.junichi@lab.ntt.co.jp
[2] Mitubishi Electric
Takashima.Katsuyuki@aj.MitsubishiElectric.co.jp

Abstract. Inner product functional encryption (IPFE), introduced by Abdalla et al. (PKC2015), is a kind of functional encryption supporting only inner product functionality. All previous IPFE schemes are bounded schemes, meaning that the vector length that can be handled in the scheme is fixed in the setup phase. In this paper, we propose the first unbounded IPFE schemes, in which we do not have to fix the lengths of vectors in the setup phase and can handle (a priori) unbounded polynomial lengths of vectors. Our first scheme is private-key based and fully function hiding. That is, secret keys hide the information of the associated function. Our second scheme is public-key based and provides adaptive security in the indistinguishability based security definition. Both our schemes are based on SXDH, which is a well-studied standard assumption, and secure in the standard model. Furthermore, our schemes are quite efficient, incurring an efficiency loss by only a small constant factor from previous bounded function hiding schemes.

Keywords: functional encryption, inner product, function hiding, unbounded, bilinear maps

1 Introduction

Functional encryption (FE) [9, 27] is an advanced cryptographic paradigm that is expected to drastically enhance the availability of encrypted data. Traditional encryption schemes can provide only "all-or-nothing" decryption capability over encrypted data, i.e., an owner of a legitimate decryption key can learn the entire data from a ciphertext and the others can learn nothing. In contrast, FE allows a legitimate user to learn some computed results from encrypted data without revealing any other information. More precisely, FE supporting a function class $\mathcal{F}$ allows an owner of a master secret key msk to issue a secret key sk_f for any function $f \in \mathcal{F}$, and decrypting a ciphertext ct_m of a message m with sk_f reveals only $f(m)$ and nothing else.

Although there are several constructions of FE for all circuits [17, 18, 30], all are based on currently impractical cryptographic primitives such as indistinguishability obfuscation [17] or multi-linear maps [16]. As a result, such general purpose FEs are far from practical, and this is why Abdalla et al. [1] initiated the study of a more specific and practical FE, i.e., inner product functional encryption (IPFE). In IPFE, an owner of a master secret key msk can issue a secret key $\mathsf{sk}_{\mathbf{y}}$ for a vector $\mathbf{y}$ and decrypting a ciphertext $\mathsf{ct}_{\mathbf{x}}$ of a vector $\mathbf{x}$ with $\mathsf{sk}_{\mathbf{y}}$ reveals only the inner product $\langle \mathbf{x}, \mathbf{y} \rangle$. The inner product is a simple but interesting function, because it is sufficient to directly compute weighted means over numerical data and useful for statistical computations. Furthermore, we can evaluate any polynomial over the data by encrypting all monomials appearing in the desired family of polynomials beforehand.

Following the work of Abdalla et al., there arose two main streams of works on IPFE. The first stream is for public-key based IPFE [2, 5], aiming to obtain the adaptive security, and the second stream is for private-key based IPFE [8,13, 21, 22, 24, 28], aiming to obtain function privacy and better efficiency. Function privacy is an important property of FE when it is used to delegate computation to another party. Recently, a multi-input version of IPFE has also been considered in [3, 4, 14].

Although most above (single-input) IPFE schemes are efficient and based on standard assumptions, all have one inconvenient property: they are *bounded*. That is, we need to fix the maximum length of vectors to be handled in the scheme at the beginning. After fixing the maximum length, we cannot handle vectors whose lengths exceed it. This is very inconvenient because it is almost impossible in the setup phase to predict which data will be encrypted. One may think that we can solve the problem by setting the maximum length to a quite large value. However, the size of a public parameter of bounded schemes expands at least linearly with the fixed maximum length, and such a solution incurs an unnecessary efficiency loss. Hence, it is desirable that we do not need to declare the maximum length of vectors to be handled in the scheme at the beginning and can make encryption or key generation for vectors with unbounded lengths. In the context of inner product predicate encryption (IPPE) [20] and attribute-based encryption [19], there exist unbounded schemes [10,11,23,26], whose public parameters do not impose a limit on the maximum length of vectors or number of attributes used in the scheme. Thus, we naturally have the following question:

Can we construct IPFE schemes that can handle vectors with unbounded lengths?

Our contributions We answer the question affirmatively. More precisely, we construct two concrete unbounded IPFE (UIPFE) schemes on the basis of the standard SXDH assumption that are both secure in the standard model.

1. The first scheme is private-key IPFE with fully function hiding, which is the strongest indistinguishability based security notion when considering function privacy [13].
2. The second scheme is public-key IPFE with adaptive security, which is a standard and desirable indistinguishability based security notion [5].

private-key scheme					
scheme	$\lvert\mathsf{msk}\rvert$	$\lvert\mathsf{ct}\rvert$	$\lvert\mathsf{sk}\rvert$	pairing	assumption
DDM16 [13]	$(8m^2 + 12m + 28)\lvert\mathbb{Z}_p\rvert$	$(4m + 8)\lvert G_1\rvert$	$(4m + 8)\lvert G_2\rvert$	Yes	SXDH
TAO16 [28]	$(4m^2 + 18m + 20)\lvert\mathbb{Z}_p\rvert$	$(2m + 5)\lvert G_1\rvert$	$(2m + 5)\lvert G_2\rvert$	Yes	XDLIN
KKS17 [22]	$(6m + 8)\lvert\mathbb{Z}_p\rvert$	$(2m + 8)\lvert G_1\rvert$	$(2m + 8)\lvert G_2\rvert$	Yes	SXDH
Ours 1	|PRF key|	$4m\lvert G_1\rvert$	$4m\lvert G_2\rvert + \alpha$	Yes	SXDH

public-key scheme						
scheme	$\lvert\mathsf{pk}\rvert$	$\lvert\mathsf{msk}\rvert$	$\lvert\mathsf{ct}\rvert$	$\lvert\mathsf{sk}\rvert$	pairing	assumption
ALS16 [5]	$(m + 1)\lvert G\rvert$	$2m\lvert\mathbb{Z}_p\rvert$	$(m + 2)\lvert G\rvert$	$2\lvert\mathbb{Z}_p\rvert + \beta$	No	DDH
Ours 2	$28\lvert G_1\rvert$	$28\lvert\mathbb{Z}_p\rvert$	$7m\lvert G_1\rvert$	$7m\lvert G_2\rvert + \alpha$	Yes	SXDH

Table 1. Comparison among private-key schemes that are fully function hiding and public-key schemes with adaptive security in the standard model. Although Lin also presented a construction of function hiding scheme [24], her scheme is the selective secure one and we do not adopt it here. A natural number $m \in \mathbb{N}$ denotes a length of a vector associated with the ciphertext or secret key. In our schemes, α denotes a bit length that is necessary to specify an index set associated with a vector. In the ALS16 scheme, β denotes a bit length that is necessary to specify a vector to be embed into a secret key. In this table, we omit a group description in a public key.

Table 1 compares efficiency among private-key schemes that are fully function hiding and public-key schemes with adaptive security in the standard model. Both our schemes achieve almost the same efficiency as the previous bounded fully function hiding IPFE schemes except the small constant factor. Note that previous public-key based schemes do not need pairing when instantiated from a cyclic group [1, 5]. However, we do not know how to construct unbounded public-key based IPFE schemes *without pairing*.

In UIPFE schemes, we can consider various conditions about encryption, key generation, and decryption. It is another important merit of UIPFE. For encryption and key generation, we can consider two cases, *consecutive* and *separate*. In the consecutive setting, each element of a vector is automatically indexed to its position when the vector is input to an encryption or key generation algorithm, i.e., for a vector (a, b, c), a's index is set to 1, b's index to 2, and c's index to 3. On the other hand, in the separate setting, an index set is attached to a vector and encryption and key generation are executed correspondingly to its index set. In other words, a vector (a, b, c) is indexed by some set, e.g., $\{1, 5, 6\}$, and the indices of a, b and c are set to 1,5, and 6, respectively. A separate scheme obviously suggests a consecutive scheme with respect to encryption or key generation. Next, we focus on the conditions of decryption. Similar to [26], we can classify the decryptable condition of IPFE schemes into three types: *ct-dominant*, *sk-dominant*, and *equal*. Let S_{ct} be an index set of a ciphertext ct and S_{sk} be an index set of a secret key sk. Then ct is decryptable with sk iff $S_{\mathsf{ct}} \supseteq S_{\mathsf{sk}}$ in ct-dominant schemes, $S_{\mathsf{ct}} \subseteq S_{\mathsf{sk}}$ in sk-dominant schemes, and $S_{\mathsf{ct}} = S_{\mathsf{sk}}$ in equal schemes. We denote the type of the schemes described above as (E:xx, K:yy, D:zz) where xx, yy $\in$ {con, sep}, and zz $\in$ {ct-dom, sk-dom, eq}, which means that encryption is xx setting, key generation is yy setting, and decryption is zz setting. It is not difficult to observe that the setting (E:sep, K:con, D:ct-dom) is meaningless because only the *consecutive* part of *separate* ciphertexts can be decrypted with any *consecutive* secret key. For example, for a ciphertext with an index set $\{1, 2, 4\}$, the element indexed as 4 is never used for decryption in the K:con setting. Hence, it is the same as the (E:con, K:con, D:ct-dom) setting. Similarly, (E:con, K:sep, D:sk-dom), (E:con, K:sep, D:eq), and (E:sep, K:con, D:eq) are also meaningless. Thus, we can consider eight types of UIPFE schemes.

In this paper, we focus on the D:ct-dom setting because we believe it is the most convenient for real applications. Consider the situation where Alice holds a huge encrypted database in an untrusted server. When she wants the server to make some computation over the database, she can obtain the result by sending a corresponding secret key to the server. If the necessary part of the database for the computation is very small, the D:ct-dom setting allows Alice to issue a compact secret key. This is because the size of a secret key of IPFE schemes typically grows linearly to the length of the corresponding vector. In the other settings, Alice needs to issue a secret key that is at least larger than some constant multiple of the size of the database, and this incurs a big efficiency loss.

Both our schemes are the (E:con, K:sep, D:ct-dom) setting, which suggests (E:con, K:con, D:ct-dom). Some readers may wonder why we do not consider the most general setting of D:ct-dom, (E:sep, K:sep, D:ct-dom), which suggests all D:ct-dom schemes. The reason is we can prove the security of our schemes against adaptive adversaries only in the (E:con, K:sep, D:ct-dom) setting. The intuitive reason for this limitation is that, in security proofs, reduction algorithms need to guess the contents of an index set with which an adversary queries an encryption oracle. This is possible in the E:con setting because the length of vectors queried by an adversary is a polynomial and a reduction algorithm can correctly guess the length with a non-negligible probability. In the E:sep setting, however, the possibility of index sets is exponential and is unpredictable for reduction algorithms. For this reason, our schemes are secure against selective adversaries in the (E:sep, K:sep, D:ct-dom) setting. In particular, our public-key scheme is semi-adaptively secure in the (E:sep, K:sep, D:ct-dom) setting, which means that the adversary declares a challenge message right after obtaining a public key in a security game [12]. Note that the fully function hiding private-key IPFE scheme in the (E:sep, K:con, D:sk-dom) setting is trivial with our scheme because the roles of ciphertexts and secret keys are the same in fully function hiding private-key IPFE. In addition, the fully function hiding private-key IPFE in the (E:con, K:con, D:eq) setting is easily constructible and we describe it in full version. We summarize our result in Table 2.

1.1 Our Techniques

We use bracket notation to denote elements on the exponent of a group element, i.e., for $\iota \in \{1, 2, T\}$, $[x]_\iota$ denotes g_ι^x where g_ι is a generator of a cyclic group G_ι.

private-key scheme				
	E:con, K:con	E:sep, K:con	E:con, K:sep	E:sep, K:sep
D:ct-dom	full	$\perp$	full	selective
D:sk-dom	full	full	$\perp$	selective
D:eq	full	$\perp$	$\perp$	open

public-key scheme				
	E:con, K:con	E:sep, K:con	E:con, K:sep	E:sep, K:sep
D:ct-dom	adaptive	$\perp$	adaptive	semi-adaptive
D:sk-dom	open	open	$\perp$	open
D:eq	open	$\perp$	$\perp$	open

Table 2. Summary of our result. A symbol $\perp$ indicates that the scheme is meaningless.

Private-key UIPFE Our starting point is the fully function hiding unbounded multi-input IPFE (MIPFE) scheme proposed by Datta et al. [14]. In an unbounded MIPFE scheme, an index space for slots are not determined in the setup phase. Then, roughly speaking, an encryption algorithm can generate a ciphertext that corresponds to a vector $\mathbf{x}$ and an arbitrary index $i \in \mathbb{N}$. Also, a key generation algorithm can issue a secret key that is associated with indexed vectors $(S, \{\mathbf{y}_i\}_{i \in S})$ for an arbitrary set $S \subset \mathbb{N}$. Only if a decryptor has all ciphertexts corresponding to elements of the set S, i.e., $\{\mathsf{ct}_i := \mathsf{MIPFE.Enc}(\mathsf{pp}, \mathsf{msk}, i, \mathbf{x}_i)\}_{i \in S}$, the secret key for S can be used for legitimate decryption and reveals $\sum_{i \in S} \langle \mathbf{x}_i, \mathbf{y}_i \rangle$. Their scheme is based on the dual pairing vector spaces (DPVS) framework introduced by Okamoto and Takashima [25] and utilizes a pseudorandom function (PRF) to handle an unbounded index space. Consider the unbounded MIPFE scheme in which the vector length is set to 1 and observe that such a scheme already serves the function of UIPFE in the D:ct-dom setting. More precisely, to encrypt $\mathbf{x} := (x_1, \ldots, x_m) \in \mathbb{Z}^m$, the encryption algorithm computes $\mathsf{ct}_i := \mathsf{MIPFE.Enc}(\mathsf{pp}, \mathsf{msk}, i, x_i)$ for all $i \in [m]$ and set $\mathsf{ct} := (\mathsf{ct}_1, \ldots, \mathsf{ct}_m)$. In key generation for an indexed vector $(S, \mathbf{y} := (y_i)_{i \in S} \in \mathbb{Z}^S)$, the key generation algorithm computes $\mathsf{sk} := \mathsf{MIPFE.KeyGen}(\mathsf{pp}, \mathsf{msk}, S, \mathbf{y})$. Then $\mathsf{MIPFE.Dec}(\mathsf{pp}, \mathsf{ct}, \mathsf{sk})$ outputs $\sum_{i \in S} x_i y_i$. However, this construction allows recomposition of ciphertexts due to the property of MIPFE. That is, for $\mathsf{ct}_1 := (\mathsf{ct}_{1,1}, \ldots, \mathsf{ct}_{1,m})$ and $\mathsf{ct}_2 := (\mathsf{ct}_{2,1}, \ldots, \mathsf{ct}_{2,m})$, we can decrypt a ciphertext like $(\mathsf{ct}_{1,1}, \mathsf{ct}_{2,2}, \ldots, \mathsf{ct}_{2,m})$ correctly whereas UIPFE should not allow such recomposition of ciphertexts.

To prevent such recomposition, each ciphertext of our scheme has a unique randomness that all elements in a ciphertext share. Decryption is possible only if an input ciphertext has a consistent randomness, so this unique randomness prevents recomposed ciphertexts from being decrypted correctly. Essentially, a ciphertext for index i of the MIPFE scheme by Datta et al. has a form like $[\mathbf{c}_i]_1 := [(x_i, 1)\mathbf{B}_i]_1$ and each element of a secret key has a form like $[\mathbf{k}_i]_2 := [(y_i, r_i)\mathbf{B}_i^*]_2$, where $\mathbf{B}_i$ is a 2×2 regular matrix, $\mathbf{B}_i^* := (\mathbf{B}_i^{-1})^\top$, and r_i are random elements in $\mathbb{Z}_p$ s.t. $\sum_{i \in S} r_i = 0$. Bases $\mathbf{B}_i$ are generated unboundedly with a PRF. A decryption algorithm computes $[\sum_{i \in S} \langle \mathbf{c}_i, \mathbf{k}_i \rangle]_T$ and it reveals the inner product $\sum_{i\in S}(x_iy_i + r_i) = \sum_{i\in S} x_iy_i$. In this construction, switching elements of one ciphertext that have the same indices as others does not affect the correct decryption. On the other hand, an element of one ciphertext corresponding to index i of our scheme has a form like $[\mathbf{c}_i]_1 := [(x_i, z)\mathbf{B}_i]_1$ where z is a unique randomness for each ciphertext, whereas each element of a secret key is the same as in the MIPFE scheme. Then it is easy to confirm that unless all $\mathbf{c}_i$ for $i \in S$ have the same randomness, $[\sum_{i\in S}\langle \mathbf{c}_i, \mathbf{k}_i\rangle]_T$ does not reveal the inner product $\sum_{i\in S} x_iy_i$ and this construction prevents recomposition of ciphertexts.

Although the concept of the construction is simple, the security proof of the scheme is rather complicated. The basic proof strategy of our scheme is the same as that by Tomida et al. [28], who proposed a fully function hiding *bounded* IPFE scheme, and this strategy is also employed in [14]. In the case of unbounded MIPFE and UIPFE, however, we encounter a new challenging problem that does not appear in *bounded* IPFE: how to prove collusion resistance against illegitimate secret keys queried by an adversary. More precisely, in the D:ct-dom setting, secret keys whose index sets are not included in the index set of a ciphertext must be useless to decrypt the ciphertext even if their owners collude. For example, an owner of a ciphertext ct_1 for a index set {1,2,3} and two secret keys sk_1 and sk_2 for index sets {1,2,4} and {3,4} respectively must not learn any information about underlying vectors in the ciphertext and secret keys.

In the context of unbounded MIPFE, the problem was solved by cleverly utilizing symmetric key encryption (SKE). Briefly, ciphertexts for index i contain a secret key of SKE that is unique to the index i. On the other hand, a secret key of unbounded MIPFE for an index set S is iteratively encrypted by SKE with all secret keys of SKE in the set S. Then, unless an owner of the secret key for a set S has all ciphertexts in the set S, he or she cannot decrypt the secret key of unbounded MIPFE encrypted by SKE and the secret key is useless to derive information from ciphertexts corresponding to any proper subset of S. Due to UIPFE not allowing the recomposition of ciphertexts, however, we cannot apply a similar technique to UIPFE schemes.

To solve this problem, we introduce a new proof strategy. In fully function hiding scheme, we consider an adversary that can query many ciphertexts and secret keys. First, we generate a situation where it is sufficient to consider only one ciphertext and all secret keys by using hidden spaces of DPVS framework. We can consider that this is a kind of dual system methodology by Waters [29], which allows us to reduce the problem of a security for many keys to that for one key [31]. Then what we need to do next is to ensure that illegitimate keys are useless to decrypt the ciphertext. For the purpose, we randomize all elements in illegitimate secret keys whose indices are out of the index set of the ciphertext by computational argument. That is, the randomization is indistinguishable for all probabilistic polynomial time (PPT) adversaries under the SXDH assumption. In the above simple example, it means that the elements for index 4 in both secret keys are randomized. The intuitive reason to take this step is to ensure that partial decryption does not leak any information on underlying vectors. That is, in the above example, one can correctly compute the term $x_i y_i$ for indices 1 and 2 with sk_1 and 3 with sk_2, which is masked by the term zr_i. What we want to prove here is that the all zr_i terms are indistinguishable from independently random elements in $\mathbb{Z}_p$ and they completely hide the terms $x_i y_i$. Recall that elements in each secret key contain the random numbers r_i such that $\sum_{i \in S} r_i = 0$. Then, if at least one of r_is in each secret key is randomized, entire r_is become completely random elements in $\mathbb{Z}_p$. At this point, partial decryption with illegitimate secret keys reveals no meaningful information and we can complete the proof.

Public-key UIPFE Our public-key UIPFE scheme is technically more intricate than our private-key one. Because we do not need to publish any information for encryption in the private-key UIPFE scheme, we can utilize PRFs to generate dual orthonormal bases unboundedly, which is necessary for encryption. More precisely, an encryption algorithm generates a basis for index i as $F_K(i)$ where F_K is a PRF, and encode the i-th element of the vector using the basis. In the public-key setting, however, a setup algorithm needs to publish information that is needed to encrypt vectors. Thus an encryptor cannot utilize PRFs to generate bases because if a key of a PRF is public, the output is no longer pseudorandom.

Our approach to overcome this problem is an indexing technique [26], which is introduced to construct unbounded inner product *predicate* encryption (IPPE) and attribute based encryption (ABE) schemes. Briefly, we add a two-dimensional prefix that specifies an index to a vector to be encoded, and only if the indices of a ciphertext and a secret key are equal, the correct inner product value is computable. In a ciphertext side, an encoding of the i-th element of a vector $\mathbf{x} := (x_1, \ldots, x_m)$ is the form like $[\mathbf{c}_i]_1 := [(\pi_i(1, i), x_i, z)\mathbf{B}]_1$ and in a secret key side, the index j of an indexed vector $(S, \mathbf{y} := (y_j)_{j \in S})$ is encoded as $[\mathbf{k}_j]_2 := [(\rho_j(-j, 1), y_j, r_j)\mathbf{B}^*]_2$. Then, although all indices share the same dual orthonormal bases, $[\langle \mathbf{c}_i, \mathbf{k}_j \rangle]_T$ reveals the meaningful value only if $i = j$. By this construction, each element in ciphertexts and secret keys is encoded as if dual orthonormal bases that are unique to each index were used.

The basic concept of the security proof of our public-key scheme is also similar to that in [26]. That is, we prove lemmas that say that normal ciphertexts and secret keys are indistinguishable from ones encoded on "somewhat" random dual orthonormal bases for each index by amplifying the entropy of the two-dimensional prefix. More concretely, we use a kind of the following relation in the security proof. Note that it is just a toy example for an intuitive explanation and an informal one. That is, for any polynomial $m := m(\lambda)$, we have the computational indistinguishability:

$$\left\{ \begin{matrix} [(\pi_i(1, i), x_i, z, \ldots)\mathbf{B}]_1 \\ [(\rho_i(-i, 1), y_i, r_i, \ldots)\mathbf{B}^*]_2 \end{matrix} \right\}_{i \in [m]} \approx_c \left\{ \begin{matrix} [(\pi_i(1, i), x_i, z, \ldots)\mathbf{D}_i]_1 \\ [(\rho_i(-i, 1), y_i, r_i, \ldots)\mathbf{D}_i^*]_2 \end{matrix} \right\}_{i \in [m]},$$

where $\{\pi_i\}_{i \in [m]}, \{\rho_i\}_{i \in [m]} \xleftarrow{\mathsf{U}} \mathbb{Z}_p$ and $\mathbf{D}_i := \mathbf{W}_i \mathbf{B}$. LHS represents normal elements of a ciphertext and secret key, and RHS represents elements of ones encoded on "somewhat" random dual orthonormal bases for each index. Here, each $\mathbf{W}_i$ need not be a completely random matrix, and it is sufficient if $\mathbf{W}_i$ is chosen from some specific distribution for our security proof. This is why we call $\mathbf{D}_i$ "somewhat" random. At this point, we can use the proof strategy similar to that of the private-key IPFE scheme because dual orthonormal bases are generated somewhat randomly for each index and we have a similar situation to the private-key IPFE scheme. Although the top-level concept of the techniques are similar to [26], i.e., indexing and entropy amplification, we cannot directly use their techniques because the security proof of our scheme is completely different from that of their scheme. Therefore, we managed to tailor lemmas of entropy amplification suitable for our scheme.

1.2 Discussion

In this work, we cannot achieve the schemes that have the following two features. We quickly discuss the difficulty about them.

Public-key UIPFE scheme without pairing. We briefly explain the reason why constructing unbounded public-key IPFE without pairing is difficult. First, we recall the bounded scheme without pairing by Abdalla et al. in [1] (and the scheme in [5] essentially follows the construction of Abdalla et al.). In their scheme, a master secret key is a randomly chosen vector $\mathbf{s} \in \mathbb{Z}_p^n$ and a public key is a vector of group elements $g^{\mathbf{s}} \in G^n$. To encrypt a vector $\mathbf{x} \in \mathbb{Z}^n$, an encryption algorithm choose a random number $r \in \mathbb{Z}_p$ and compute the ciphertext as $\mathsf{ct} := (g^r, g^{r\mathbf{s}+\mathbf{x}}) \in G^{n+1}$. On the other hand, a secret key for a vector $\mathbf{y} \in \mathbb{Z}^n$ is set as $\mathsf{sk} := (\langle \mathbf{y}, \mathbf{s} \rangle, \mathbf{y}) \in \mathbb{Z}_p^{n+1}$, and a decryption algorithm computes $g^{\langle r\mathbf{s}+\mathbf{x}, \mathbf{y} \rangle} / g^{r \langle \mathbf{y}, \mathbf{s} \rangle} = g^{\langle \mathbf{x}, \mathbf{y} \rangle}$. To handle vectors with unbounded lengths, an encryption algorithm or a key generation algorithm needs to generate an element s.t. $g^{r\mathbf{s}+\mathbf{x}}$ or $\langle \mathbf{y}, \mathbf{s} \rangle$ respectively for a vector $\mathbf{s}$ with an arbitrary length from a fixed public key or master secret key.

As we explained in the technical section, we obtain such a situation by entropy amplification and it requires computational arguments. However, if secret keys consist of elements in $\mathbb{Z}_p$ likely to the scheme by Abdalla et al., we cannot apply computational arguments to secret keys. Therefore, it seems inevitable to encode elements in secret keys on the exponent of group elements to leverage computational arguments, and it incurs the necessity of pairing in decryption.

Adaptively secure (E:sep, K:sep) UIPFE schemes. As we mentioned, our proof strategy needs to guess an index set of a ciphertext and inherently we cannot apply it to (E:sep, K:sep) schemes with adaptive security. We consider that this difficulty is similar to that to prove adaptive security of multi-use KP-ABE from static assumptions (this problem is solved in the semi-adaptive setting [12]). That is, the reduction algorithm needs to embed the instance of an underlying problem into secret keys depending on the instance that the adversary outputs in the challenge phase. Hence, the difficulty disappears in the semi-adaptive setting because the reduction knows the challenge instance before it simulates secret keys. We know that we can obtain adaptively secure multi-use

KP-ABE from so-called q-type assumptions [6,7], then we might be able to obtain adaptively secure (E:sep, K:sep) schemes from q-type assumptions similarly.

1.3 Concurrent Work

Concurrently and independently, Dufour Sans and Pointcheval also presented UIPFE schemes [15]. In our term, they proposed public-key (E:sep, K:sep, D:eq) and (E:sep, K:sep, D:ct-dom) schemes in their paper. Their schemes have short secret keys, meaning that they contain one group element and a corresponding vector. However, their schemes rely on the random oracle model and achieve only the selective security, and their (E:sep, K:sep, D:ct-dom) scheme also relies on a new interactive assumption. More precisely, they assume that a kind of problem is hard for all PPT adversaries even if they are allowed to access some oracles. In addition, their (E:sep, K:sep, D:ct-dom) scheme does not have collusion resistance of illegitimate secret keys, which means that a combination of illegitimate keys can become a legitimate key.

2 Preliminary

2.1 Notations

For a prime p, $\mathbb{Z}_p$ denotes a field $\mathbb{Z}/p\mathbb{Z}$. For natural numbers $n, m \in \mathbb{N}$, $[n]$ denotes a set $\{1, \ldots, n\}$, and $[m, n]$ denotes a set $\{m, \ldots, n\}$ (if $m > n$, $[m, n] := \phi$). For a set S, $s \xleftarrow{\mathsf{U}} S$ denotes that s is uniformly chosen from S. We treat vectors as row vectors. For a vector $\mathbf{x}$, $||\mathbf{x}||_\infty$ denotes its infinity norm. For a field K, $\mathsf{M}_n(K)$ and $\mathsf{GL}_n(K)$ denote a set of all $n \times n$ matrices and all $n \times n$ regular matrices whose elements are in K, respectively. We use a bold upper-case letter to denote a matrix, e.g., $\mathbf{A}$, and a bold lower-case version of the same letter with subscript i to denote the i-th row of the matrix, e.g., $\mathbf{a}_i$. For example, $\mathbf{a}_i$ denotes the i-th row of $\mathbf{A}$. For a regular matrix $\mathbf{A}$, $\mathbf{A}^*$ denotes $(\mathbf{A}^{-1})^\top$. For a generator g_ι of a cyclic group G_ι, a matrix $\mathbf{A}$, and vector $\mathbf{a}$, $[\mathbf{A}]_\iota$ and $[\mathbf{a}]_\iota$ denote the corresponding matrix and vector on the exponent of g_ι, respectively. For vectors $\mathbf{x} := (x_1, \ldots, x_n)$ and $\mathbf{y} := (y_1, \ldots, y_n) \in \mathbb{Z}_p^n$, let $e([\mathbf{x}]_1, [\mathbf{y}]_2) := e(g_1, g_2)^{\langle \mathbf{x}, \mathbf{y} \rangle}$ be a function that computes the inner product on the exponent by $\prod_{i \in [n]} e([x_i]_1, [y_i]_2)$. A function $f : \mathbb{N} \to \mathbb{R}$ is called negligible if $f(\lambda) = \lambda^{-\omega(1)}$ and denotes $f(\lambda) \leq \mathsf{negl}(\lambda)$.

2.2 Basic Notions

Definition 2.1 (Pseudorandom Functions). A pseudorandom function (PRF) family $\mathcal{F} := \{F_K\}_{K \in \mathcal{K}_\lambda}$ with a key space $\mathcal{K}_\lambda$, a domain $\mathcal{X}_\lambda$, and a range $\mathcal{Y}_\lambda$ is a function family that consists of functions $F_K : \mathcal{X}_\lambda \to \mathcal{Y}_\lambda$. Let $\mathcal{R}_\lambda$ be a set of functions consisting of all functions whose domain and range are $\mathcal{X}_\lambda$ and $\mathcal{Y}_\lambda$ respectively. For any PPT adversary $\mathcal{A}$, the following condition holds,

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{PRF}}(\lambda) := \left| \Pr[1 \leftarrow \mathcal{A}^{F_K(\cdot)}] - \Pr[1 \leftarrow \mathcal{A}^{R(\cdot)}] \right| \leq \mathsf{negl}(\lambda),$$

where $K \overset{\mathsf{U}}{\leftarrow} \mathcal{K}_\lambda$ and $R \overset{\mathsf{U}}{\leftarrow} \mathcal{R}_\lambda$.

Definition 2.2 (Bilinear Groups). Bilinear groups $\mathbb{G}:=(p, G_1, G_2, G_T, g_1, g_2, e)$ consist of a prime p, cyclic groups G_1, G_2, G_T of order p, generators g_1 and g_2 of G_1 and G_2 respectively, and a bilinear map $e : G_1 \times G_2 \to G_T$, which has two properties.

– (Bilinearity): $\forall h_1 \in G_1, h_2 \in G_2, a, b \in \mathbb{Z}_p, e(h_1^a, h_2^b) = e(h_1, h_2)^{ab}$.
– (Non-degeneracy): For generators g_1 and g_2, $e(g_1, g_2)$ is a generator of G_T.

A bilinear group generator $\mathcal{G}_{\mathsf{BG}}(1^\lambda)$ takes security parameter 1^λ and outputs bilinear groups $\mathbb{G}$ with a λ-bit prime p.

Definition 2.3 (SXDH Assumption). For $\iota \in \{1, 2\}$, we define the following distribution,

$$\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda), \ \ a, e, f \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p, \ \ D := (\mathbb{G}, [a]_\iota, [e]_\iota)$$
$$[t_\beta]_\iota := [ae + \beta f]_\iota \ \text{ for } \beta \in \{0, 1\}.$$

We say the SXDH assumption holds if for any PPT adversary $\mathcal{A}$ and both $\iota \in \{1, 2\}$,

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{SXDH}}(\lambda) := |\Pr[1 \leftarrow \mathcal{A}(D, [t_0]_\iota)] - \Pr[1 \leftarrow \mathcal{A}(D, [t_1]_\iota)]| \leq \mathsf{negl}(\lambda).$$

2.3 Unbounded Inner Product Functional Encryption for (E:con, K:sep, D:ct-dom)

In this paper, we propose two unbounded inner product functional encryption schemes. The first scheme is private-key unbounded IPFE that is fully function hiding and the second one is public-key unbounded IPFE with adaptive security. Both our schemes can handle (a-priori) unbounded polynomial lengths of vectors for encryption and key generation, and support a function that we call limited-norm inner product. As explained in the introduction, our schemes support inner product in the (E:con, K:sep, D:ct-dom) setting. Informally, for a ciphertext of a vector whose length is m and a secret key with a set S, only if $S \subseteq [m]$, we can decrypt the ciphertext with the secret key and learn the inner product value over the set S. Note that in previous works [3,4], the term *bounded-norm* is used, but in this paper, *bounded* generally refers to vector length. Therefore, we use *limited-norm* for the functionality in this paper.

Definition 2.4 (Limited-Norm Inner Product). This function family $\mathcal{F}$ consists of functions $f_{S,\mathbf{y}}^{X,Y} : \mathbb{Z}^m \to \mathbb{Z}$ where $X, Y \in \mathbb{N}$, $S \subset \mathbb{N}$, $\mathbf{y} := (y_i)_{i \in S} \in \mathbb{Z}^S$ s.t. $||\mathbf{y}||_\infty \leq Y$, and $m \in \mathbb{N}$ s.t. $S \subseteq [m]$. We define the function for every $\mathbf{x} := (x_1, \ldots, x_m) \in \mathbb{Z}^m$ s.t. $||\mathbf{x}||_\infty \leq X$ as

$$f_{S,\mathbf{y}}^{X,Y}(\mathbf{x}) := \sum_{i \in S} x_i y_i.$$

Definition 2.5 (Private-Key Unbounded Inner Product Functional Encryption). Let $\mathcal{X} := \{X_\lambda\}_{\lambda\in\mathbb{N}}, \mathcal{Y} := \{Y_\lambda\}_{\lambda\in\mathbb{N}}$ be ensembles of norm-limit. Private-key unbounded inner product functional encryption (Priv-UIPFE) consists of four algorithms.

$\mathsf{Setup}(1^\lambda)$**:** This algorithm takes a security parameter 1^λ, and outputs a public parameter pp and a master secret key msk.

$\mathsf{Enc}(\mathsf{pp}, \mathsf{msk}, \mathbf{x})$**:** This algorithm takes $\mathsf{pp}, \mathsf{msk}$, and a vector $\mathbf{x} := (x_1, \ldots, x_m) \in \mathbb{Z}^m$ where $m := m(\lambda)$ is any polynomial. It outputs a ciphertext ct_m

$\mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, S, \mathbf{y})$**:** This algorithm takes $\mathsf{pp}, \mathsf{msk}$, a non-empty index set $S \subseteq [s]$ where $s := s(\lambda)$ is any polynomial, and an indexed vector $\mathbf{y} := (y_i)_{i\in S} \in \mathbb{Z}^S$. It outputs a secret key sk_S.

$\mathsf{Dec}(\mathsf{pp}, \mathsf{ct}_m, \mathsf{sk}_S)$**:** This algorithm takes $\mathsf{pp}, \mathsf{ct}_m$ and sk_S and outputs a decrypted value $d \in \mathbb{Z}$ or a symbol $\perp$.

Correctness Priv-UIPFE is *correct* if it satisfies the following condition. For any $\lambda \in \mathbb{N}$, $\mathbf{x} \in \mathbb{Z}^m$ s.t. $m := m(\lambda)$ is any polynomial and $||\mathbf{x}||_\infty \leq X_\lambda$, index set $S \subseteq [s]$ s.t. $s := s(\lambda)$ is any polynomial and $S \subseteq [m]$, and $\mathbf{y} \in \mathbb{Z}^S$ s.t. $||\mathbf{y}||_\infty \leq Y_\lambda$, we have

$$\mathsf{Pr}\left[d = \sum_{i\in S} x_i y_i \;\middle|\; \begin{array}{l} (\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathsf{ct}_m \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathsf{msk}, \mathbf{x}) \\ \mathsf{sk}_S \leftarrow \mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, S, \mathbf{y}) \\ d := \mathsf{Dec}(\mathsf{pp}, \mathsf{ct}_m, \mathsf{sk}_S) \end{array} \right] \geq 1 - \mathsf{negl}(\lambda).$$

Security Priv-UIPFE is *fully function hiding* if it satisfies the following condition. That is, the advantage of $\mathcal{A}$ against Priv-UIPFE defined as follows is negligible in λ for any PPT adversary $\mathcal{A}$,

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Priv\text{-}UIPFE}}(\lambda) := \left| \begin{array}{l} \mathsf{Pr}\left[\begin{array}{l} 1 \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Enc},0}(\mathsf{pp},\mathsf{msk},\cdot),\mathcal{O}_{\mathsf{KG},0}(\mathsf{pp},\mathsf{msk},\cdot,\cdot)}(\mathsf{pp}) : \\ (\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda) \end{array} \right] \\ -\mathsf{Pr}\left[\begin{array}{l} 1 \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Enc},1}(\mathsf{pp},\mathsf{msk},\cdot),\mathcal{O}_{\mathsf{KG},1}(\mathsf{pp},\mathsf{msk},\cdot,\cdot)}(\mathsf{pp}) : \\ (\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda) \end{array} \right] \end{array} \right|.$$

Here, $\mathcal{O}_{\mathsf{Enc},\beta}(\mathsf{pp}, \mathsf{msk}, \cdot)$ with $\beta \in \{0, 1\}$ is an encryption oracle that takes a pair of vectors $(\mathbf{x}^0, \mathbf{x}^1) \in (\mathbb{Z}^m)^2$ with the same polynomial length m, and outputs $\mathsf{Enc}(\mathsf{pp}, \mathsf{msk}, \mathbf{x}^\beta)$. $\mathcal{O}_{\mathsf{KG},\beta}(\mathsf{pp}, \mathsf{msk}, \cdot, \cdot)$ with $\beta \in \{0, 1\}$ is a key generation oracle that takes a set S including polynomial indices and a pair of indexed vectors $(\mathbf{y}^0, \mathbf{y}^1) \in (\mathbb{Z}^S)^2$ associated with the index set S, and outputs $\mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, S, \mathbf{y}^\beta)$. To avoid a trivial attack of $\mathcal{A}$, we have the following condition on $\mathcal{A}$'s queries. Let q_{ct} (resp. q_{sk}) be a total number of ciphertext query (resp. secret key query) of $\mathcal{A}$. For all $j \in [q_{\mathsf{ct}}]$ and $\ell \in [q_{\mathsf{sk}}]$, if $S_\ell \subseteq [m_j]$, then

$$\sum_{i\in S_\ell} x^0_{j,i} y^0_{\ell,i} = \sum_{i\in S_\ell} x^1_{j,i} y^1_{\ell,i}. \tag{1}$$

Consider the modified game where the adversary queries all vectors in one-shot, i.e., $\{(\mathbf{x}_j^0, \mathbf{x}_j^1)\}_{j\in[q_{\mathsf{ct}}]}$ and $\{(\mathbf{y}_\ell^0, \mathbf{y}_\ell^1)\}_{\ell\in[q_{\mathsf{sk}}]}$, right after obtaining a public parameter, and then the adversary receive all ciphertexts and secret keys for queried vectors for β-side. If the advantage of all PPT adversary against the modified game is negligible, we say that Priv-UIPFE is *selectively function hiding.*

Definition 2.6 (Public-key Unbounded Inner Product Functional Encryption). Let $\mathcal{X} := \{X_\lambda\}_{\lambda\in\mathbb{N}}, \mathcal{Y} := \{Y_\lambda\}_{\lambda\in\mathbb{N}}$ be ensembles of norm-limit. Public-key unbounded inner product functional encryption (Pub-UIPFE) consists of four algorithms.

$\mathsf{Setup}(1^\lambda)$**:** This algorithm takes a security parameter 1^λ, and outputs a public key pk and a master secret key msk.

$\mathsf{Enc}(\mathsf{pk}, \mathbf{x})$**:** This algorithm takes pk and a vector $\mathbf{x} := (x_1, \ldots, x_m) \in \mathbb{Z}^m$ where $m := m(\lambda)$ is any polynomial. It outputs a ciphertext ct_m

$\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, S, \mathbf{y})$**:** This algorithm takes $\mathsf{pk}, \mathsf{msk}$, a non-empty index set $S \subseteq [s]$ where $s := s(\lambda)$ is any polynomial, and an indexed vector $\mathbf{y} := (y_i)_{i\in S} \in \mathbb{Z}^S$. It outputs a secret key sk_S.

$\mathsf{Dec}(\mathsf{pk}, \mathsf{ct}_m, \mathsf{sk}_S)$**:** This algorithm takes $\mathsf{pk}, \mathsf{ct}_m$ and sk_S and outputs a decrypted value $d \in \mathbb{Z}$ or a symbol $\perp$.

Correctness Pub-UIPFE is *correct* if it satisfies the following condition. For any $\lambda \in \mathbb{N}$, $\mathbf{x} \in \mathbb{Z}^m$ s.t. $m := m(\lambda)$ is any polynomial and $||\mathbf{x}||_\infty \leq X_\lambda$, index set $S \subseteq [s]$ s.t. $s := s(\lambda)$ is any polynomial and $S \subseteq [m]$, and $\mathbf{y} \in \mathbb{Z}^S$ s.t. $||\mathbf{y}||_\infty \leq Y_\lambda$, we have

$$\Pr\left[d = \sum_{i\in S} x_i y_i \;\middle|\; \begin{array}{l} (\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathsf{ct}_m \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathbf{x}) \\ \mathsf{sk}_S \leftarrow \mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, S, \mathbf{y}) \\ d := \mathsf{Dec}(\mathsf{pk}, \mathsf{ct}_m, \mathsf{sk}_S) \end{array} \right] \geq 1 - \mathsf{negl}(\lambda).$$

Security Pub-UIPFE is *adaptively secure* if it satisfies the following condition. That is, the advantage of $\mathcal{A}$ against Pub-UIPFE defined as follows is negligible in λ for any stateful PPT adversary $\mathcal{A}$,

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Pub\text{-}UIPFE}}(\lambda) := \left| \begin{array}{l} \Pr\left[\beta = 1 \;\middle|\; \begin{array}{l} (\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda) \\ (\mathbf{x}^0, \mathbf{x}^1) \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, \cdot, \cdot)}(\mathsf{pk}) \\ \mathsf{ct}_{m^*} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathbf{x}^0) \\ \beta \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, \cdot, \cdot)}(\mathsf{pk}, \mathsf{ct}_{m^*}) \end{array} \right] \\ -\Pr\left[\beta = 1 \;\middle|\; \begin{array}{l} (\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda) \\ (\mathbf{x}^0, \mathbf{x}^1) \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, \cdot, \cdot)}(\mathsf{pk}) \\ \mathsf{ct}_{m^*} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathbf{x}^1) \\ \beta \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, \cdot, \cdot)}(\mathsf{pk}, \mathsf{ct}_{m^*}) \end{array} \right] \end{array} \right|.$$

Here, the challenge vectors $\mathbf{x}^0$ and $\mathbf{x}^1$ that $\mathcal{A}$ outputs must have the same length m^*. To avoid a trivial attack of $\mathcal{A}$, we have the following condition on $\mathcal{A}$'s queries.

Let q_{sk} be a total number of secret key query of $\mathcal{A}$. For all $\ell \in [q_{\mathsf{sk}}]$, if $S_\ell \subseteq [m^*]$, then

$$\sum_{i \in S_\ell} x_i^0 y_{\ell,i} = \sum_{i \in S_\ell} x_i^1 y_{\ell,i}. \tag{2}$$

Consider the modified game where the adversary is prohibited to make a secret-key query before outputting challenge vectors $(\mathbf{x}^0, \mathbf{x}^1)$. If the advantage of all PPT adversary against the modified game is negligible, we say that Pub-UIPFE is *semi-adaptively secure.*

3 Private-Key Unbounded Inner Product Functional Encryption

Our schemes are based on the DPVS framework introduced by Okamoto and Takashima [25]. We use the following lemma in our Priv-IPFE scheme, which is implicitly shown in [14].

Lemma 3.1. *Let p be a λ-bit prime. For any polynomial $m := m(\lambda)$ and $n := n(\lambda)$, we have*

$$\mathsf{Pr}[\exists i, \det \mathbf{B}_i = 0 | \mathbf{B}_1, \ldots, \mathbf{B}_m \overset{\mathsf{U}}{\leftarrow} \mathsf{M}_n(\mathbb{Z}_p)] = 2^{-\Omega(\lambda)}.$$

3.1 Construction

In the following scheme, norm limits X_λ, Y_λ are some polynomials in λ. Let $\mathcal{F} := \{F_K\}_{K \in \mathcal{K}_\lambda}$ be a PRF family with a key space $\mathcal{K}_\lambda$ consisting of functions $F_K : \{0,1\}^\lambda \to \mathsf{M}_4(\mathbb{Z}_p)$.

$\mathsf{Setup}(1^\lambda)$**:** Takes a security parameter 1^λ and chooses bilinear groups $\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda)$ and a PRF key $K \overset{\mathsf{U}}{\leftarrow} \mathcal{K}_\lambda$. Outputs

$$\mathsf{pp} := \mathbb{G}, \quad \mathsf{msk} := K.$$

$\mathsf{Enc}(\mathsf{pp}, \mathsf{msk}, \mathbf{x})$**:** Takes $\mathsf{pp}, \mathsf{msk}$ and $\mathbf{x} := (x_1, \ldots, x_m) \in \mathbb{Z}^m$ where $m := m(\lambda)$ is any polynomial. Sets $\mathbf{B}_i := F_K(i)$ and $\mathbf{c}_i := (x_i, 0, z, 0)\mathbf{B}_i \in \mathbb{Z}_p^4$ for all $i \in [m]$, where $z \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p$. Outputs

$$\mathsf{ct}_m := ([\mathbf{c}_1]_1, \ldots, [\mathbf{c}_m]_1).$$

If there exists $i \in [m]$ such that $\mathbf{B}_i$ is a singular matrix, outputs $\perp$.

$\mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, S, \mathbf{y})$**:** Takes pp, msk, a non-empty index set $S \subseteq [s]$ where $s := s(\lambda)$ is any polynomial, and an indexed vector $\mathbf{y} := (y_i)_{i \in S} \in \mathbb{Z}^S$. Chooses $\{r_i\}_{i \in S} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p$ s.t. $\sum_{i \in S} r_i = 0$. Sets $\mathbf{B}_i := F_K(i)$ and $\mathbf{k}_i := (y_i, 0, r_i, 0)\mathbf{B}_i^* \in \mathbb{Z}_p^4$ for all $i \in S$. Outputs

$$\mathsf{sk}_S := (S, \{[\mathbf{k}_i]_2\}_{i \in S}).$$

If there exists $i \in S$ such that $\mathbf{B}_i$ is a singular matrix, outputs $\perp$.

$\mathsf{Dec}(\mathsf{pp}, \mathsf{ct}_m, \mathsf{sk}_S)$**:** Takes pp, a ciphertext ct_m for m dimensional vector, and a secret key sk_S for a index set S. If $S \subseteq [m]$, then computes

$$h := \prod_{i \in S} e([\mathbf{c}_i]_1, [\mathbf{k}_i]_2),$$

and searches for d s.t. $e(g_1, g_2)^d = h$ exhaustively in the range of $-|S|X_\lambda Y_\lambda$ to $|S|X_\lambda Y_\lambda$. If such d is found, outputs d. Otherwise, outputs $\perp$.

Correctness Our Priv-UIPFE scheme is correct if $\mathcal{F}$ is a PRF family. We consider the case where for a natural number $m \in \mathbb{N}$, $\mathbf{B}_i := F_K(i)$ for all $i \in [m]$ is invertible. Then, we observe that if $S \subseteq [m]$,

$$h = \prod_{i \in S} e([\mathbf{c}_i]_1, [\mathbf{k}_i]_2) = e(g_1, g_2)^{\sum_{i \in S} \langle \mathbf{c}_i, \mathbf{k}_i \rangle} = e(g_1, g_2)^{\sum_{i \in S}(x_i y_i + z r_i)}.$$

Here we have $\sum_{i \in S} r_i = 0$, then $h = e(g_1, g_2)^{\sum_{i \in S} x_i y_i}$. If $||\mathbf{x}||_\infty \leq X_\lambda$ and $||\mathbf{y}||_\infty \leq Y_\lambda$, $|\sum_{i \in S} x_i y_i| \leq |S| X_\lambda Y_\lambda$ and Dec outputs $\sum_{i \in S} x_i y_i$. Hence, if $\mathbf{B}_i$ for all $i \in [m]$ is invertible without a negligible probability, our scheme is correct. Let $m := m(\lambda)$ be any polynomial. For $i \in [m]$, we have $\mathsf{Pr}[\exists i, \det \mathbf{B}_i = 0 | \mathbf{B}_i \overset{\mathsf{U}}{\leftarrow} \mathsf{M}_4(\mathbb{Z}_p)] = 2^{-\Omega(\lambda)}$ from Lemma 3.1 and $|\mathsf{Pr}[\exists i, \det \mathbf{B}_i = 0 | \mathbf{B}_i \overset{\mathsf{U}}{\leftarrow} \mathsf{M}_4(\mathbb{Z}_p)] - \mathsf{Pr}[\exists i, \det \mathbf{B}_i = 0 | K \overset{\mathsf{U}}{\leftarrow} \mathcal{K}_\lambda, \mathbf{B}_i := F_K(i)]| \leq \mathsf{negl}(\lambda)$ from the definition of PRF. Consequently, $\mathsf{Pr}[\exists i, \det \mathbf{B}_i = 0 | K \overset{\mathsf{U}}{\leftarrow} \mathcal{K}_\lambda, \mathbf{B}_i := F_K(i)] \leq \mathsf{negl}(\lambda)$.

Remark 3.1. Similarly to all previous IPFE schemes based on a cyclic group or bilinear groups, the decryption algorithm of our schemes need to solve the small discrete logarithm problem. As pointed out in [21], however, this step does not affect efficiency so much in many practical applications.

3.2 Security

Theorem 3.1. *Assume that the SXDH assumption holds and $\mathcal{F}$ is a PRF family, then our Priv-UIPFE is fully function hiding. More formally, let m_{max} be the maximum length of vectors with which $\mathcal{A}$ makes a query to the encryption oracle, then for any PPT adversary $\mathcal{A}$ and security parameter λ, there exists a PPT adversary $\mathcal{B}_1$ for the SXDH and $\mathcal{B}_2$ for the PRF family, we have*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Priv\text{-}UIPFE}}(\lambda) \leq \{4q_{\mathsf{sk}} + 2(m_{\mathsf{max}} + 1)q_{\mathsf{ct}} + 2\}\mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{SXDH}}(\lambda) + 2\mathsf{Adv}_{\mathcal{B}_2}^{\mathsf{PRF}}(\lambda) + 2^{-\Omega(\lambda)}.$$

Proof outline The top-level strategy of the proof is similar to that of the proof by Tomida et al. [28], although the order of changing the forms of ciphertexts and secret keys is the opposite. In the security proof, we employ a usual hybrid argument and gradually change the forms of ciphertexts and secret keys queried by an adversary from the case of $\beta = 0$ to $\beta = 1$ defined in Definition 2.5. We use the spaces not used in the actual function, i.e., the second and fourth spaces, for the security proof. Intuitively, the second space is a kind of a working space to handle intermediate states between $\beta = 0$ and $\beta = 1$, and the fourth space is utilized to make a situation where we can focus on only one query even if an adversary makes multiple queries. In other words, we can see the fourth space as a semi-functional space of dual system methodology proposed by Waters [29]. First, the form of secret keys is changed from $[(y^0_{\ell,i}, 0, r_{\ell,i}, 0)\mathbf{B}^*_i]_2$ to $[(y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, 0)\mathbf{B}^*_i]_2$ in the Game 1 sequence. Next, we change the form of ciphertexts from $[(x^0_{j,i}, 0, z_j, 0)\mathbf{B}_i]_1$ to $[(0, x^1_{j,i}, z_j, 0)\mathbf{B}_i]_1$ in the Game 3 sequence, and here we leverage the game condition Eq.(1). Then we switch the first space with the second space as $[(x^1_{j,i}, 0, z_j, 0)\mathbf{B}_i]_1$ and $[(y^1_{\ell,i}, y^0_{\ell,i}, r_{\ell,i}, 0)\mathbf{B}^*_i]_2$. Finally, the form of secret keys is changed from $[(y^1_{\ell,i}, y^0_{\ell,i}, r_{\ell,i}, 0)\mathbf{B}^*_i]_2$ to $[(y^1_{\ell,i}, 0, r_{\ell,i}, 0)\mathbf{B}^*_i]_2$ as the reverse of the Game 1 sequence. The most complicated and important part is the Game 3 sequence, in which we need to deal with the ciphertexts and secret keys that do not satisfy the condition Eq.(1). In Game 3 sequence, we change the ciphertexts from 0-side to 1-side one by one, and in the ν-th iteration of Game 3 sequence, we change the ν-th ciphertexts from 0-side to 1-side. For the ν-th ciphertext, we can classify secret keys queried by an adversary into three types. Let m_ν be the length of the ciphertext.

1. The index set S of the secret key is included in $[m_\nu]$, i,e., $\max S \leq m_\nu$.
2. A part of the index set S is included in $[m_\nu]$, i.e., $(\max S > m_\nu) \wedge (\min S \leq m_\nu)$.
3. The index set S and $[m_\nu]$ are disjoint, i.e., $\min S > m_\nu$.

The cumbersome secret keys are type 2 keys because they can correctly decrypt a part of the ciphertext even though they may not satisfy the condition Eq.(1). We want to change the form of the ν-th ciphertext from 0-side to 1-side by information-theoretical change in Game 3-ν-1-4, but it does not work without any treatment due to the above property of type 2 keys. Therefore, we manage to randomize or "sanitize" type 2 keys from Game 3-ν-1-1 to Game 3-ν-1-3.

Proof. We prove Theorem 3.1 by a series of games. For each game transition, we prove that the difference between probabilities that the adversary $\mathcal{A}$ outputs 1 in both games is negligible.

Game 0: This game is the same as the real security game when $\beta = 0$ in Definition 2.5. That is, the j-th ciphertext query with a pair of vectors $(\mathbf{x}^0_j, \mathbf{x}^1_j) \in (\mathbb{Z}^{m_j})^2$ is replied as

$$\begin{aligned} \mathbf{c}_{j,i} &:= (x^0_{j,i}, 0, z_j, 0)\mathbf{B}_i \text{ for all } i \in [m_j] \\ \mathsf{ct}_{j,m_j} &:= ([\mathbf{c}_{j,1}]_1, \ldots, [\mathbf{c}_{j,m_j}]_1). \end{aligned}$$

The ℓ-th secret key query with an index set S_ℓ and a pair of vectors $(\mathbf{y}^0_\ell, \mathbf{y}^1_\ell) \in (\mathbb{Z}^{S_\ell})^2$ is replied as

$$\begin{aligned} \mathbf{k}_{\ell,i} &:= (y^0_{\ell,i}, 0, r_{\ell,i}, 0)\mathbf{B}^*_i \text{ for all } i \in S_\ell \\ \mathsf{sk}_{\ell,S_\ell} &:= (S_\ell, \{[\mathbf{k}_{\ell,i}]_2\}_{i \in S_\ell}). \end{aligned}$$

Game 0': This game is the same as Game 0 except for the way of making dual orthonormal bases. In Game 0, the dual orthonormal bases for the i-th element are made as $(\mathbf{B}_i, \mathbf{B}_i^*)$ where $\mathbf{B}_i := F_K(i)$, but in Game 0', they are made as $\mathbf{B}_i \xleftarrow{\mathsf{U}} \mathsf{GL}_4(\mathbb{Z}_p)$. More precisely, the cipertext oracle and secret key oracle have the same list $\mathcal{L}$ for bases. When the oracle needs a basis for the i-th element, it searches for $(i, \mathbf{B}_i)$ from $\mathcal{L}$. If the oracle find it, the oracle uses the bases, and if not, it generates $\mathbf{B}_i \xleftarrow{\mathsf{U}} \mathsf{GL}_4(\mathbb{Z}_p)$ and records them as $(i, \mathbf{B}_i)$ into $\mathcal{L}$.

Game 1-μ-1 ($\mu \in [q_{\mathsf{sk}}]$)**:** We define Game 1-0-3 as equivalent to Game 0'. This game is the same as Game 1-($\mu - 1$)-3 except that in the μ-th secret key query, $\mathbf{k}_{\mu,i}$ is set as

$$w \xleftarrow{\mathsf{U}} \mathbb{Z}_p, \ \ \mathbf{k}_{\mu,i} := (y^0_{\mu,i}, 0, r_{\mu,i}, \boxed{wr_{\mu,i}})\mathbf{B}_i^* \ \text{ for all } i \in S_\mu.$$

Game 1-μ-2 ($\mu \in [q_{\mathsf{sk}}]$)**:** This game is the same as Game 1-μ-1 except that in the μ-th secret key query, $\mathbf{k}_{\mu,i}$ is set as

$$w \xleftarrow{\mathsf{U}} \mathbb{Z}_p, \ \ \mathbf{k}_{\mu,i} := (y^0_{\mu,i}, \boxed{y^1_{\mu,i}}, r_{\mu,i}, wr_{\mu,i})\mathbf{B}_i^* \ \text{ for all } i \in S_\mu.$$

Game 1-μ-3 ($\mu \in [q_{\mathsf{sk}}]$)**:** This game is the same as Game 1-μ-2 except that in the μ-th secret key query, $\mathbf{k}_{\mu,i}$ is set as

$$\mathbf{k}_{\mu,i} := (y^0_{\mu,i}, y^1_{\mu,i}, r_{\mu,i}, \boxed{0})\mathbf{B}_i^* \ \text{ for all } i \in S_\mu.$$

Game 2: This game is the same as Game 1-q_{sk}-3 except that in all secret key queries, $\mathbf{k}_{\ell,i}$ for all $\ell \in [q_{sk}]$ is set as

$$\mathbf{k}_{\ell,i} := (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, \boxed{\tilde{r}_{\ell,i}})\mathbf{B}_i^* \ \text{ for all } i \in S_\ell,$$

where $\tilde{r}_{\ell,i} \xleftarrow{\mathsf{U}} \mathbb{Z}_p$ s.t. $\sum_{i \in S_\ell} \tilde{r}_{\ell,i} = 0$.

Game 3-ν-1 ($\nu \in [q_{\mathsf{ct}}]$)**:** Game 2 is equivalent to Game 3-0-3. This game is the same as Game 3-($\nu - 1$)-3 except that in the ν-th ciphertext query, $\mathbf{c}_{\nu,i}$ is set as

$$\tilde{z}_\nu \xleftarrow{\mathsf{U}} \mathbb{Z}_p, \ \ \mathbf{c}_{\nu,i} := (x^0_{\nu,i}, 0, z_\nu, \boxed{\tilde{z}_\nu})\mathbf{B}_i \ \text{ for all } i \in [m_\nu].$$

Game 3-ν-2 ($\nu \in [q_{\mathsf{ct}}]$)**:** This game is the same as Game 3-ν-1 except that in the ν-th ciphertext query, $\mathbf{c}_{\nu,i}$ is set as

$$\tilde{z}_\nu \xleftarrow{\mathsf{U}} \mathbb{Z}_p, \ \ \mathbf{c}_{\nu,i} := (\boxed{0, x^1_{\nu,i}}, z_\nu, \tilde{z}_\nu)\mathbf{B}_i \ \text{ for all } i \in [m_\nu].$$

Game 3-ν-3 ($\nu \in [q_{\mathsf{ct}}]$)**:** This game is the same as Game 3-ν-2 except that in the ν-th ciphertext query, $\mathbf{c}_{\nu,i}$ is set as

$$\mathbf{c}_{\nu,i} := (0, x^1_{\nu,i}, z_\nu, \boxed{0})\mathbf{B}_i \ \text{ for all } i \in [m_\nu].$$

Game 4: This game is the same as Game 3-q_{ct}-5 except that in all ciphertext and secret key queries, $\mathbf{c}_{j,i}$ and $\mathbf{k}_{\ell,i}$ are set as

$$\mathbf{c}_{j,i} := (\boxed{x^1_{j,i}, 0}, z_j, 0)\mathbf{B}_i \ \text{ for all } i \in [m_j],$$
$$\mathbf{k}_{\ell,i} := (\boxed{y^1_{\ell,i}, y^0_{\ell,i}}, r_{\ell,i}, \tilde{r}_{\ell,i})\mathbf{B}^*_i, \ \text{ for all } i \in S_\ell.$$

Game 5: This game is the same as the real security game when $\beta = 1$ in Definition 2.5. That is, the j-th ciphertext query with a pair of vectors $(\mathbf{x}^0_j, \mathbf{x}^1_j) \in (\mathbb{Z}^{m_j})^2$ is replied as

$$\mathbf{c}_{j,i} := (x^1_{j,i}, 0, z_j, 0)\boxed{\mathbf{B}_i} \ \text{ for all } i \in [m_j]$$
$$\mathsf{ct}_{j,m_j} := ([\mathbf{c}_{j,1}]_1, \ldots, [\mathbf{c}_{j,m_j}]_1).$$

The ℓ-th secret key query with an index set S_ℓ and a pair of vectors $(\mathbf{y}^0_\ell, \mathbf{y}^1_\ell) \in (\mathbb{Z}^{S_\ell})^2$ is replied as

$$\mathbf{k}_{\ell,i} := (y^1_{\ell,i}, \boxed{0}, r_{\ell,i}, \boxed{0})\boxed{\mathbf{B}^*_i} \ \text{ for all } i \in S_\ell$$
$$\mathsf{sk}_{\ell,S_\ell} := (S_\ell, \{[\mathbf{k}_{\ell,i}]_2\}_{i \in S_\ell}).$$

Note that $\mathbf{B}_i$ is generated as $\mathbf{B}_i := F_K(i)$ in Game 5.

Thanks to Lemma 3.2 to Lemma 3.11, we can conclude the proof of Theorem 3.1. □

In the following, we denote the event that $\mathcal{A}$ outputs 1 in Game ι by E_ι.

Lemma 3.2. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for PRFs s.t.*

$$|\Pr[\mathsf{E}_0] - \Pr[\mathsf{E}_{0'}]| \leq \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

Proof. First, we consider Game 0_{M}, which is the same as Game 0 except that $\mathbf{B}_i$ is generated as $\mathbf{B}_i \xleftarrow{\mathsf{U}} \mathsf{M}_4(\mathbb{Z}_p)$ for each i. The following inequality directly follows from the property of PRF s.t. $|\Pr[\mathsf{E}_0] - \Pr[\mathsf{E}_{0_\mathsf{M}}]| \leq \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{B}}(\lambda)$. Next, we have $|\Pr[\mathsf{E}_{0_\mathsf{M}}] - \Pr[\mathsf{E}_{0'}]| \leq 2^{-\Omega(\lambda)}$ from Lemma 3.1. Then Lemma 3.2 holds. □

Lemma 3.3. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$|\Pr[\mathsf{E}_{1\text{-}(\mu\,-\,1)\text{-}3}] - \Pr[\mathsf{E}_{1\text{-}\mu\text{-}1}]| \leq \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

Proof. We show that we can make a reduction algorithm $\mathcal{B}$ for the SXDH using $\mathcal{A}$. $\mathcal{B}$ obtains an instance of SXDH with $\iota := 2$, i.e., $(\mathbb{G}, [a]_2, [e]_2, [t_\beta]_2)$, and sets $\mathsf{pp} := \mathbb{G}$. $\mathcal{B}$ defines random dual orthonormal bases $\mathbf{B}_i, \mathbf{B}^*_i$ as follows,

$$\mathbf{W}_i \xleftarrow{\mathsf{U}} \mathsf{GL}_4(\mathbb{Z}_p), \ \ \mathbf{B}_i := \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 0 & 1 \\ & & 1 & -a \end{pmatrix} \mathbf{W}_i, \ \ \mathbf{B}^*_i := \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & a & 1 \\ & & 1 & 0 \end{pmatrix} \mathbf{W}^*_i \in \mathsf{GL}_4(\mathbb{Z}_p).$$

Then $\mathcal{B}$ simulates all ciphertext queries and all secret key queries except the μ-th one as follows.

$$[\mathbf{c}_{j,i}]_1 := [(x^0_{j,i}, 0, z_{j,i}, 0)\mathbf{B}_i]_1 \text{ for all } i \in [m_j],$$
$$[\mathbf{k}_{\ell,i}]_2 := \begin{cases} [(y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, 0)\mathbf{B}^*_i]_2 \text{ for all } i \in S_\ell & (\ell < \mu) \\ [(y^0_{\ell,i}, 0, r_{\ell,i}, 0)\mathbf{B}^*_i]_2 \text{ for all } i \in S_\ell & (\ell > \mu). \end{cases}$$

Note that $\mathcal{B}$ cannot compute $[\mathbf{b}_{i,4}]_1$ because it does not know $[a]_1$, but the above instances are computable without $[\mathbf{b}_{i,4}]_1$. For the μ-th secret key query, $\mathcal{B}$ replies to $\mathcal{A}$ for all $i \in S_\mu$ as

$$r'_i \xleftarrow{\mathsf{U}} \mathbb{Z}_p \text{ s.t. } \sum_{i \in S_\mu} r'_i = 0,$$
$$[\mathbf{k}_{\mu,i}]_2 := [(y^0_{\mu,i}, 0, 0, 0)\mathbf{B}^*_i + r'_i(0, 0, t_\beta, e)\mathbf{W}^*_i]_2 = [(y^0_{\mu,i}, 0, er'_i, \beta f r'_i)\mathbf{B}^*_i]_2.$$

Observe that we can implicitly set $r_{\mu,i} := er'_i$ and $w := f/e$ unless $e = 0$, then $\mathcal{A}$'s view is the same as in Game 1-$(\mu - 1)$-3 (resp. Game 1-μ-1) if $\beta = 0$ (resp. $\beta = 1$). □

Lemma 3.4. *For any PPT adversary $\mathcal{A}$, we have*

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

Proof. We define $(\mathbf{D}_i, \mathbf{D}^*_i)$ as

$$\mathbf{D}_i := \begin{pmatrix} 1 & & & 0 \\ & 1 & & \frac{y^1_{\mu,i}}{wr_{\mu,i}} \\ & & 1 & 0 \\ & & & 1 \end{pmatrix} \mathbf{B}_i, \quad \mathbf{D}^*_i := \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ 0 & -\frac{y^1_{\mu,i}}{wr_{\mu,i}} & 0 & 1 \end{pmatrix} \mathbf{B}^*_i \in \mathsf{GL}_4(\mathbb{Z}_p).$$

Observe that $(\mathbf{D}_i, \mathbf{D}^*_i)$ are random dual orthonormal bases. Then, for all $j \in [q_{\mathsf{ct}}]$ and $\ell \in [q_{\mathsf{sk}}]$, we have

$$\mathbf{c}_{j,i} = (x^0_{j,i}, 0, z_{j,i}, 0)\mathbf{B}_i = (x^0_{j,i}, 0, z_{j,i}, 0) \begin{pmatrix} 1 & & & 0 \\ & 1 & & -\frac{y^1_{\mu,i}}{wr_{\mu,i}} \\ & & 1 & 0 \\ & & & 1 \end{pmatrix} \mathbf{D}_i = (x^0_{j,i}, 0, z_{j,i}, 0)\mathbf{D}_i,$$

$$\begin{aligned}\mathbf{k}_{\ell,i} &= (y^0_{\ell,i}, \beta_\ell y^1_{\ell,i}, r_{\ell,i}, \hat{\beta}_\ell wr_{\mu,i})\mathbf{B}^*_i = (y^0_{\ell,i}, \beta_\ell y^1_{\ell,i}, r_{\ell,i}, \hat{\beta}_\ell wr_{\mu,i}) \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ 0 & \frac{y^1_{\mu,i}}{wr_{\mu,i}} & 0 & 1 \end{pmatrix} \mathbf{D}^*_i \\ &= (y^0_{\ell,i}, (\beta_\ell + \hat{\beta}_\ell)y^1_{\ell,i}, r_{\ell,i}, \hat{\beta}_\ell wr_{\mu,i})\mathbf{D}^*_i,\end{aligned}$$

where $\beta_\ell = 0$ if $\ell \geq \mu$ and $\beta_\ell = 1$ if $\ell < \mu$, and $\hat{\beta}_\ell = 0$ if $\ell \neq \mu$ and $\hat{\beta}_\ell = 1$ if $\ell = \mu$. Then if $w \neq 0$ and $r_{\mu,i} \neq 0$, $\mathcal{A}$'s view is identically distributed in Game 1-μ-2 and Game 1-μ-3. □

Lemma 3.5. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$|\Pr[\mathsf{E}_{1\text{-}\mu\text{-}2}] - \Pr[\mathsf{E}_{1\text{-}\mu\text{-}3}]| \leq \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

This lemma can be proven almost the same as Lemma 3.3, so we omit the proof.

Lemma 3.6. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$|\Pr[\mathsf{E}_{1\text{-}q_{\mathsf{sk}}\text{-}3}] - \Pr[\mathsf{E}_{2}]| \leq \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}$$

Proof. We show that we can make a reduction algorithm $\mathcal{B}$ for the SXDH using $\mathcal{A}$. $\mathcal{B}$ obtains an instance of SXDH with $\iota := 2$, i.e., $(\mathbb{G}, [a]_2, [e]_2, [t_\beta]_2)$, and sets $\mathsf{pp} := \mathbb{G}$. $\mathcal{B}$ defines random dual orthonormal bases $\mathbf{B}_i, \mathbf{B}^*_i$ as follows,

$$\mathbf{W}_i \xleftarrow{\mathsf{U}} \mathsf{GL}_4(\mathbb{Z}_p), \ \ \mathbf{B}_i := \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 0 & 1 \\ & & 1 & -a \end{pmatrix} \mathbf{W}_i, \ \ \mathbf{B}^*_i := \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & a & 1 \\ & & 1 & 0 \end{pmatrix} \mathbf{W}^*_i \in \mathsf{GL}_4(\mathbb{Z}_p).$$

Then $\mathcal{B}$ simulates all ciphertext queries and all secret key queries as follows.

$$\begin{aligned} &[\mathbf{c}_{j,i}]_1 := [(x^0_{j,i}, 0, z_{j,i}, 0)\mathbf{B}_i]_1 \ \text{ for all } i \in [m_j], \\ &r'_{\ell,i}, r''_{\ell,i} \xleftarrow{\mathsf{U}} \mathbb{Z}_p \ \text{ s.t. } \sum_{i \in S_\ell} r'_{\ell,i} = \sum_{i \in S_\ell} r''_{\ell,i} = 0, \\ &[\mathbf{k}_{\ell,i}]_2 := [(y^0_{\ell,i}, y^1_{\ell,i}, r'_{\ell,i}, 0)\mathbf{B}^*_i + r''_{\ell,i}(0,0,t_\beta,e)\mathbf{W}^*_i]_2 \\ &\quad = [(y^0_{\ell,i}, y^1_{\ell,i}, r'_{\ell,i} + er''_{\ell,i}, \beta f r''_{\ell,i})\mathbf{B}^*_i]_2 \ \text{ for all } i \in S_\ell. \end{aligned}$$

Note that $\mathcal{B}$ cannot compute $[\mathbf{b}_{i,4}]_1$ because it does not know $[a]_1$, but the above instances are computable without $[\mathbf{b}_{i,4}]_1$. Observe that we can implicitly set $r_{\ell,i} := r'_{\ell,i} + er''_{\ell,i}$ and $\tilde{r}_{\ell,i} := fr''_{\ell,i}$ unless $f = 0$, then $\mathcal{A}$'s view is the same as in Game 1-q_{sk}-3 (resp. Game 2) if $\beta = 0$ (resp. $\beta = 1$). □

Lemma 3.7. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$|\Pr[\mathsf{E}_{3\text{-}(\nu\,-\,1)\text{-}3}] - \Pr[\mathsf{E}_{3\text{-}\nu\text{-}1}]| \leq \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda).$$

Proof. We show that we can make a reduction algorithm $\mathcal{B}$ for the SXDH using $\mathcal{A}$. $\mathcal{B}$ obtains an instance of SXDH with $\iota := 1$, i.e., $(\mathbb{G}, [a]_1, [e]_1, [t_\beta]_1)$, and sets $\mathsf{pp} := \mathbb{G}$. $\mathcal{B}$ defines random dual orthonormal bases $\mathbf{B}_i, \mathbf{B}^*_i$ as follows,

$$\mathbf{W}_i \xleftarrow{\mathsf{U}} \mathsf{GL}_4(\mathbb{Z}_p), \ \ \mathbf{B}_i := \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & a & 1 \\ & & 1 & 0 \end{pmatrix} \mathbf{W}_i, \ \ \mathbf{B}^*_i := \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 0 & 1 \\ & & 1 & -a \end{pmatrix} \mathbf{W}^*_i \in \mathsf{GL}_4(\mathbb{Z}_p).$$

Then $\mathcal{B}$ simulates all ciphertext queries except the ν-th one and all secret key queries as follows,

$$[\mathbf{c}_{j,i}]_1 := \begin{cases} [(0, x^1_{j,i}, z_{j,i}, 0)\mathbf{B}_i]_1 & \text{for all } i \in [m_j] \quad (j < \nu) \\ [(x^0_{j,i}, 0, z_{j,i}, 0)\mathbf{B}_i]_1 & \text{for all } i \in [m_j] \quad (j > \nu), \end{cases}$$

$$r'_{\ell,i}, r''_{\ell,i} \xleftarrow{\mathsf{U}} \mathbb{Z}_p \ \text{ s.t. } \sum_{i \in S_\ell} r'_{\ell,i} = \sum_{i \in S_\ell} r''_{\ell,i} = 0,$$

$$\begin{aligned}[\mathbf{k}_{\ell,i}]_2 &:= [(y^0_{\ell,i}, y^1_{\ell,i}, r'_{\ell,i}, 0)\mathbf{B}^*_i + (0, 0, r''_{\ell,i}, 0)\mathbf{W}^*_i]_2 \\ &= [(y^0_{\ell,i}, y^1_{\ell,i}, r'_{\ell,i} + ar''_{\ell,i}, r''_{\ell,i})\mathbf{B}^*_i]_2 \ \text{ for all } i \in S_\ell.\end{aligned}$$

Note that $\mathcal{B}$ cannot compute $[\mathbf{b}^*_{i,4}]_2$ because it does not know $[a]_2$, but the above instances are computable without $[\mathbf{b}^*_{i,4}]_2$. Observe that we can implicitly set $r_{\ell,i} := r'_{\ell,i} + ar''_{\ell,i}$ and $\tilde{r}_{\ell,i} := r''_{\ell,i}$, so $\mathcal{B}$ correctly simulates the answer for queries. For the ν-th ciphertext query, $\mathcal{B}$ replies to $\mathcal{A}$ for all $i \in [m_\nu]$ as

$$[\mathbf{c}_{\nu,i}]_1 := [(x^0_{\nu,i}, 0, 0, 0)\mathbf{B}_i + (0, 0, t_\beta, e)\mathbf{W}_i]_1 = [(x^0_{\nu,i}, 0, e, \beta f)\mathbf{B}_i]_1.$$

Observe that we can implicitly set $z_\nu := e$ and $\tilde{z}_\nu := f$, then $\mathcal{A}$'s view is the same as in Game 3-$(\nu-1)$-3 (resp. Game 3-ν-1) if $\beta = 0$ (resp. $\beta = 1$). □

Lemma 3.8. *Let m_{max} be the maximum length of vectors with which $\mathcal{A}$ makes a query to the encryption oracle. For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$|\Pr[\mathsf{E}_{3\text{-}\nu\text{-}1}] - \Pr[\mathsf{E}_{3\text{-}\nu\text{-}2}]| \le 2m_{\mathsf{max}}\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

Proof. To prove Lemma 3.8, we consider the following intermediate games between Game 3-ν-1 and 3-ν-2. In each intermediate game, the challenger chooses a random element $m'_\nu \xleftarrow{\mathsf{U}} [m_{\mathsf{max}}]$ as a guess of m_ν at the beginning of the games.

Game 3-ν-1-1 ($\nu \in [q_{\mathsf{ct}}]$)**:** This game is the same as Game 3-ν-1 except that the challenger aborts the game immediately if the vector length of the ν-th ciphertext query is not m'_ν i.e., $m'_\nu \neq m_\nu$. We define that $\mathcal{A}$'s output is $\perp$ when the game is aborted.

Game 3-ν-1-2 ($\nu \in [q_{\mathsf{ct}}]$)**:** This game is the same as Game 3-ν-1-1 except the following. In the ℓ-th secret key query for all ℓ s.t. whose index set S_ℓ contains both elements that are greater than m'_ν and not greater than m'_ν, i.e., $(\max S_\ell > m'_\nu) \wedge (\min S_\ell \le m'_\nu)$, $\mathbf{k}_{\ell,i}$ is set as

$$\mathbf{k}_{\ell,i} := \begin{cases} (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, \tilde{r}_{\ell,i})\mathbf{B}^*_i & (i \in S_\ell, i \le m'_\nu) \\ (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, \boxed{a\tilde{r}_{\ell,i}})\mathbf{B}^*_i & (i \in S_\ell, i > m'_\nu) \end{cases}$$

where $a \xleftarrow{\mathsf{U}} \mathbb{Z}_p, \tilde{r}_{\ell,i} \xleftarrow{\mathsf{U}} \mathbb{Z}_p$ s.t. $\sum_{i \in S_\ell} \tilde{r}_{\ell,i} = 0$.

Game 3-ν-1-3 ($\nu \in [q_{\mathsf{ct}}]$)**:** This game is the same as Game 3-ν-1-2 except that in the ℓ-th secret key query for all ℓ s.t. $(\max S_\ell > m'_\nu) \wedge (\min S_\ell \le m'_\nu)$, $\mathbf{k}_{\ell,i}$ is set as

$$\bar{r}_{\ell,i} \xleftarrow{\mathsf{U}} \mathbb{Z}_p, \ \ \mathbf{k}_{\ell,i} := (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, \boxed{\bar{r}_{\ell,i}})\mathbf{B}^*_i \ \text{ for all } i \in S_\ell.$$

Game 3-ν-1-4 ($\nu \in [q_{\mathsf{ct}}]$)**:** This game is the same as Game 3-ν-1-3 except that in the ν-th ciphertext query, $\mathbf{c}_{\nu,i}$ is set as

$$\tilde{z}_\nu \xleftarrow{\mathsf{U}} \mathbb{Z}_p, \quad \mathbf{c}_{\nu,i} := (\boxed{0, x^1_{\nu,i}}, z_\nu, \tilde{z}_\nu)\mathbf{B}_i \quad \text{for all } i \in [m'_\nu].$$

Game 3-ν-1-5 ($\nu \in [q_{\mathsf{ct}}]$)**:** This game is the same as Game 3-ν-1-4 except that in all secret key queries, $\mathbf{k}_{\ell,i}$ are set as

$$\mathbf{k}_{\ell,i} := (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, \boxed{\tilde{r}_{\ell,i}})\mathbf{B}^*_i \quad \text{for all } i \in S_\ell,$$

where $\tilde{r}_{\ell,i} \xleftarrow{\mathsf{U}} \mathbb{Z}_p$ s.t. $\sum_{i \in S_\ell} \tilde{r}_{\ell,i} = 0$.

Next we consider the probability that $\mathcal{A}$ outputs 1 in each game. Thanks to Claim 1 to Claim 6, we have

$$\begin{aligned} |\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1}] - \mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}2}]| &= m_{\mathsf{max}} |\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}1}] - \mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}5}]| \\ &\leq 2m_{\mathsf{max}} \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)} \end{aligned}$$

This concludes the proof of Lemma 3.8. □

Claim 1. For any PPT adversary $\mathcal{A}$, we have

$$\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}1}] = \frac{1}{m_{\mathsf{max}}} \mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1}]$$

Proof. First, we consider the game (denoted by Game X) that is the same as Game 3-ν-1 except that $\mathcal{A}$'s output is defined as $\perp$ when $m'_\nu \neq m_\nu$. Note that the challenger does not abort the game in Game X in contrast to Game 3-ν-1-1. It is obvious that the probabilities that $\mathcal{A}$ outputs 1 are equal in Game X and Game 3-ν-1-1 respectively. Then, we have

$$\begin{aligned} \mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}1}] = \mathsf{Pr}[\mathsf{E}_X] &= \sum_{i \in [m_{\mathsf{max}}]} \mathsf{Pr}[m'_\nu = i] \mathsf{Pr}[m_\nu = i \wedge \mathsf{E}_{3\text{-}\nu\text{-}1} | m'_\nu = i] \\ &= \frac{1}{m_{\mathsf{max}}} \sum_{i \in [m_{\mathsf{max}}]} \mathsf{Pr}[m_\nu = i \wedge \mathsf{E}_{3\text{-}\nu\text{-}1}] \\ &= \frac{1}{m_{\mathsf{max}}} \mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1}]. \end{aligned}$$

The second line follows from the fact that m'_ν is chosen independently from $\mathcal{A}$'s view in Game X and its value does not affect $\mathcal{A}$'s behavior. □

Claim 2. For any PPT adversary $\mathcal{A}$, we have

$$|\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}1}] - \mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}2}]| \leq 2^{-\Omega(\lambda)}.$$

Proof. For $i > m'_\nu$, we define $(\mathbf{D}_i, \mathbf{D}^*_i)$ as

$$\mathbf{D}_i := \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & a \end{pmatrix} \mathbf{B}_i, \;\; \mathbf{D}^*_i := \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1/a \end{pmatrix} \mathbf{B}^*_i \in \mathsf{GL}_4(\mathbb{Z}_p).$$

Ciphertexts except the ν-th one and secret keys that have indices greater than m'_ν are changed as

$$\begin{aligned} \mathbf{c}_{j,i} &= (\beta_j x^0_{j,i}, (1-\beta_j)x^1_{j,i}, z_j, 0)\mathbf{B}_i = (\beta_j x^0_{j,i}, (1-\beta_j)x^1_{j,i}, z_j, 0) \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1/a \end{pmatrix} \mathbf{D}_i \\ &= (\beta_j x^0_{j,i}, (1-\beta_j)x^1_{j,i}, z_j, 0)\mathbf{D}_i \;\text{ for all } i > m'_\nu, \\ \mathbf{k}_{\ell,i} &= (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, \tilde{r}_{\ell,i})\mathbf{B}^*_i = (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, \tilde{r}_{\ell,i}) \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & a \end{pmatrix} \mathbf{D}^*_i \\ &= (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, a\tilde{r}_{\ell,i})\mathbf{D}^*_i \;\text{ for all } i > m'_\nu, \end{aligned}$$

where $\beta_j = 0$ if $j < \nu$ and $\beta_j = 1$ if $j \geq \nu$. Note that secret keys whose all indices are greater than m'_ν are not affected by the basis change because $\{\tilde{r}_{\ell,i}\}_{i \in S_\ell}$ s.t. $\sum_{i \in S_\ell} \tilde{r}_{\ell,i} = 0$ and $\{a\tilde{r}_{\ell,i}\}_{i \in S_\ell}$ s.t. $\sum_{i \in S_\ell} \tilde{r}_{\ell,i} = 0$ are identically distributed. Finally, when $m'_\nu = m_\nu$, this basis change does not affect ct_{ν,m_ν} because it is applied only for the bases with indices $i > m_\nu$. Hence, in Game 3-ν-1-1 and Game 3-ν-1-2, $\mathcal{A}$'s view is identically distributed unless $a = 0$. □

Claim 3. For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.

$$|\Pr[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}2}] - \Pr[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}3}]| \leq \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

Proof. We show that we can make a reduction algorithm $\mathcal{B}$ for the SXDH using $\mathcal{A}$. In the beginning of the simulation, $\mathcal{B}$ chooses a $m'_\nu \overset{\mathsf{U}}{\leftarrow} [m_{\mathsf{max}}]$ as a guess of m_ν. If the guess is incorrect, $\mathcal{B}$ aborts and outputs 0. Otherwise, $\mathcal{B}$ outputs $\mathcal{A}$'s output as it is. $\mathcal{B}$ obtains an SXDH instance with $\iota := 2$, i.e., $(\mathbb{G}, [a]_2, [e]_2, [t_\beta]_2)$ and gives $\mathsf{pp} := \mathbb{G}$ to $\mathcal{A}$. $\mathcal{B}$ defines dual orthonormal bases as $\mathbf{B}_i \overset{\mathsf{U}}{\leftarrow} \mathsf{GL}_4(\mathbb{Z}_p)$ for each index i. Then, all ciphertexts and the ℓ-th secret key s.t. $(\max S_\ell \leq m'_\nu) \vee (\min S_\ell > m'_\nu)$ can be generated by using $\mathbf{B}_i$ and $\mathbf{B}^*_i$. For the ℓ-th secret key s.t. $(\max S_\ell > m'_\nu) \wedge (\min S_\ell \leq m'_\nu)$, $\mathcal{B}$ computes secret keys as follows.

$$u_{\ell,i}, u'_{\ell,i} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p \;\text{ s.t. } \sum_{i \in S_\ell} u_{\ell,i} = \sum_{i \in S_\ell} u'_{\ell,i} = 0,$$

$$[\mathbf{k}_{\ell,i}]_2 := \begin{cases} [(y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, eu_{\ell,i} + u'_{\ell,i})\mathbf{B}^*]_2 & (i \in S_\ell, i \leq m'_\nu) \\ [(y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, t_\beta u_{\ell,i} + au'_{\ell,i})\mathbf{B}^*]_2 & \\ = [(y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, a(eu_{\ell,i} + u'_{\ell,i}) + \beta f u_{\ell,i})\mathbf{B}^*]_2 & (i \in S_\ell, i > m'_\nu) \end{cases}$$

Then, we can define $\tilde{r}_{\ell,i} := eu_{\ell,i} + u'_{\ell,i}$. In the case of $\beta = 0$, $[\mathbf{k}_{\ell,i}]_2$ is distributed identically to Game 3-ν-1-2. Next, we consider the case $\beta = 1$. First, $\{\tilde{r}_{\ell,i}\}_{i \in S_\ell}$ and $\{u_{\ell,i}\}_{i \in S_\ell}$ are independently distributed because the information of $\{u_{\ell,i}\}_{i \in S_\ell}$ in $\{\tilde{r}_{\ell,i}\}_{i \in S_\ell}$ is completely hidden by $\{u'_{\ell,i}\}_{i \in S_\ell}$. Therefore, we can set $\bar{r}_{\ell,i} := \begin{cases} \tilde{r}_{\ell,i} & (i \in S_\ell, i \le m'_\nu) \\ a\tilde{r}_{\ell,i} + fu_{\ell,i} & (i \in S_\ell, i > m'_\nu) \end{cases}$, unless $f = 0$. Hence, $[\mathbf{k}_{\ell,i}]_2$ is distributed identically to Game 3-ν-1-3 if $\beta = 1$. □

Claim 4. For any PPT adversary $\mathcal{A}$, we have

$$|\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}3}] - \mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}4}]| \le 2^{-\Omega(\lambda)}.$$

Proof. Here, we denote the event such that $m'_\nu = m_\nu$ in Game ι by X_ι. By the game definition, we have

$$\begin{aligned}
&|\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}3}] - \mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}4}]| \\
=&|\mathsf{Pr}[\mathsf{X}_{3\text{-}\nu\text{-}1\text{-}3}]\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}3}|\mathsf{X}_{3\text{-}\nu\text{-}1\text{-}3}] - \mathsf{Pr}[\mathsf{X}_{3\text{-}\nu\text{-}1\text{-}4}]\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}4}|\mathsf{X}_{3\text{-}\nu\text{-}1\text{-}4}]| \\
=&|\mathsf{Pr}[\mathsf{X}_{3\text{-}\nu\text{-}1\text{-}3}](\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}3}|\mathsf{X}_{3\text{-}\nu\text{-}1\text{-}3}] - \mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}4}|\mathsf{X}_{3\text{-}\nu\text{-}1\text{-}4}])|.
\end{aligned}$$

In the third line, we use the fact that $\mathcal{A}$'s view is identical before the ν-th ciphertext query and then we have $\mathsf{Pr}[\mathsf{X}_{3\text{-}\nu\text{-}1\text{-}3}] = \mathsf{Pr}[\mathsf{X}_{3\text{-}\nu\text{-}1\text{-}4}]$. Therefore, it is sufficient to prove that $|\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}3}|\mathsf{X}_{3\text{-}\nu\text{-}1\text{-}3}] - \mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}4}|\mathsf{X}_{3\text{-}\nu\text{-}1\text{-}4}])| \le 2^{-\Omega(\lambda)}$. For the purpose, we analyze $\mathcal{A}$'s view under the condition such that $m'_\nu = m_\nu$.

We define $(\mathbf{D}_i, \mathbf{D}_i^*)$ for all $i \in [m_\nu]$ as

$$\mathbf{D}_i := \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ \frac{x^0_{\nu,i}}{\tilde{z}_\nu} & -\frac{x^1_{\nu,i}}{\tilde{z}_\nu} & 0 & 1 \end{pmatrix} \mathbf{B}_i, \;\; \mathbf{D}_i^* := \begin{pmatrix} 1 & & & -\frac{x^0_{\nu,i}}{\tilde{z}_\nu} \\ & 1 & & \frac{x^1_{\nu,i}}{\tilde{z}_\nu} \\ & & 1 & 0 \\ & & & 1 \end{pmatrix} \mathbf{B}_i^* \in \mathsf{GL}_4(\mathbb{Z}_p).$$

Observe that $(\mathbf{D}_i, \mathbf{D}_i^*)$ are random dual orthonormal bases. Then, for all $j \in [q_{\mathsf{ct}}]$, we have

$$\begin{aligned}
\mathbf{c}_{j,i} &= (\beta_j x^0_{j,i}, (1-\beta_j)x^1_{j,i}, z_j, \hat{\beta}_j \tilde{z}_\nu)\mathbf{B}_i \\
&= (\beta_j x^0_{j,i}, (1-\beta_j)x^1_{j,i}, z_j, \hat{\beta}_j \tilde{z}_\nu) \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ -\frac{x^0_{\nu,i}}{\tilde{z}_\nu} & \frac{x^1_{\nu,i}}{\tilde{z}_\nu} & 0 & 1 \end{pmatrix} \mathbf{D}_i \\
&= ((\beta_j - \hat{\beta}_j)x^0_{j,i}, (1-\beta_j+\hat{\beta}_j)x^1_{j,i}, z_j, \hat{\beta}_j \tilde{z}_\nu)\mathbf{D}_i,
\end{aligned}$$

where $\beta_j = 0$ if $j < \nu$ and $\beta_j = 1$ if $j \geq \nu$, and $\hat{\beta}_j = 0$ if $j \neq \nu$ and $\hat{\beta}_j = 1$ if $j = \nu$. On the other hand, for all ℓ s.t. $\max S_\ell \leq m_\nu$, we have

$$\mathbf{k}_{\ell,i} = (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, \tilde{r}_{\ell,i})\mathbf{B}^*_i = (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, \tilde{r}_{\ell,i}) \begin{pmatrix} 1 & & & \frac{x^0_{\nu,i}}{\tilde{z}_\nu} \\ & 1 & & -\frac{x^1_{\nu,i}}{\tilde{z}_\nu} \\ & & 1 & 0 \\ & & & 1 \end{pmatrix} \mathbf{D}^*_i$$

$$= (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, \tilde{r}_{\ell,i} + \frac{1}{\tilde{z}_\nu}(x^0_{\nu,i}y^0_{\ell,i} - x^1_{\nu,i}y^1_{\ell,i}))\mathbf{D}^*_i.$$

Here, we have the condition Eq.(1) s.t. $\sum_{i \in S_\ell}(x^0_{\nu,i}y^0_{\ell,i} - x^1_{\nu,i}y^1_{\ell,i}) = 0$, because $S_\ell \subseteq [m_\nu]$. Hence, we can set $\tilde{r}'_{\ell,i} := \tilde{r}_{\ell,i} + \frac{1}{\tilde{z}_\nu}(x^0_{\nu,i}y^0_{\ell,i} - x^1_{\nu,i}y^1_{\ell,i})$. Observe that $\tilde{r}'_{\ell,i}$ is randomly distributed s.t. $\sum_{i \in S_\ell} \tilde{r}'_{\ell,i} = 0$. In the same way, for all ℓ s.t. $(\max S_\ell > m_\nu) \wedge (\min S_\ell \leq m_\nu)$, we have

$$\mathbf{k}_{\ell,i} = \begin{cases} (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, \bar{r}_{\ell,i} + \frac{1}{\tilde{z}_\nu}(x^0_{\nu,i}y^0_{\ell,i} - x^1_{\nu,i}y^1_{\ell,i}))\mathbf{D}^*_i & (i \leq m_\nu) \\ (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, \bar{r}_{\ell,i})\mathbf{B}^*_i & (i > m_\nu) \end{cases}$$

In this case, there is no condition on $(x^0_{\nu,i}y^0_{\ell,i} - x^1_{\nu,i}y^1_{\ell,i})$. However, because $\bar{r}_{\ell,i}$ are chosen randomly from $\mathbb{Z}_p$, then $\bar{r}'_{\ell,i} := \bar{r}_{\ell,i} + \frac{1}{\tilde{z}_\nu}(x^0_{\nu,i}y^0_{\ell,i} - x^1_{\nu,i}y^1_{\ell,i})$ are also random elements in $\mathbb{Z}_p$. Note that for all ℓ s.t. $\min S_\ell > m_\nu$, this basis change does not affect $\mathsf{sk}_{\ell,S_\ell}$ because we only change the bases for $i \leq m_\nu$. Then, in Game 3-ν-1-3 and Game 3-ν-1-4, $\mathcal{A}$'s view is identically distributed unless $\tilde{z}_\nu = 0$ under the condition such that $m'_\nu = m_\nu$. □

Claim 5. For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.

$$|\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}4}] - \mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}5}]| \leq \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

Claim 5 can be proven by just the reverse of Game 3-($\nu - 1$)-1-1 to Game 3-ν-1-3, so we omit the proof.

Claim 6. For any PPT adversary $\mathcal{A}$, we have

$$\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}5}] = \frac{1}{m_{\mathsf{max}}}\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}2}]$$

The difference between Game 3-ν-1-5 and 3-ν-2 is just the existence of the abort condition introduced in Game 3-ν-1-1. Then, we can prove Claim 6 similarly to Claim 1.

Lemma 3.9. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$|\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}2}] - \mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}3}]| \leq \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda).$$

This lemma can be proven by just the reverse of Game 3-($\nu - 1$)-3 to Game 3-ν-1, so we omit the proof.

Lemma 3.10. *For any PPT adversary $\mathcal{A}$, we have*

$$\mathsf{Pr}[\mathsf{E}_{3\text{-}q_{\mathsf{ct}}\text{-}3}] = \mathsf{Pr}[\mathsf{E}_4].$$

Proof. We define $(\mathbf{D}_i, \mathbf{D}_i^*)$ as

$$\mathbf{D}_i := \begin{pmatrix} & 1 & & \\ 1 & & & \\ & & 1 & \\ & & & 1 \end{pmatrix} \mathbf{B}_i, \;\; \mathbf{D}_i^* := \begin{pmatrix} & 1 & & \\ 1 & & & \\ & & 1 & \\ & & & 1 \end{pmatrix} \mathbf{B}_i^* \in \mathsf{GL}_4(\mathbb{Z}_p).$$

Observe that $(\mathbf{D}_i, \mathbf{D}_i^*)$ are random dual orthonormal bases. Then, for all $j \in [q_{\mathsf{ct}}]$ and $\ell \in [q_{\mathsf{sk}}]$, we have

$$\mathbf{c}_{j,i} = (0, x_{j,i}^1, z_j, 0)\mathbf{B}_i = (x_{j,i}^1, 0, z_j, 0)\mathbf{D}_i \;\text{ for all } i \in [m_j],$$
$$\mathbf{k}_{\ell,i} = (y_{\ell,i}^0, y_{\ell,i}^1, r_{\ell,i}, \tilde{r}_{\ell,i})\mathbf{B}_i^* = (y_{\ell,i}^1, y_{\ell,i}^0, r_{\ell,i}, \tilde{r}_{\ell,i})\mathbf{D}_i^* \;\text{ for all } i \in S_\ell.$$

Then, in Game 3-q_{ct}-3 and Game 4, $\mathcal{A}$'s view is identically distributed. □

Lemma 3.11. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}_1$ for the SXDH and $\mathcal{B}_2$ for PRF s.t.*

$$|\mathsf{Pr}[\mathsf{E}_4] - \mathsf{Pr}[\mathsf{E}_5]| \leq (2q_{\mathsf{sk}} + 1)\mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{SXDH}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_2}^{\mathsf{PRF}}(\lambda) + 2^{-\Omega(\lambda)}.$$

This lemma can be proven by just the reverse of Games 0 to 2, so we omit the proof.

4 Public-Key Unbounded Inner Product Functional Encryption

In the following scheme, norm limits X_λ, Y_λ are some polynomials in λ.

4.1 Construction

$\mathsf{Setup}(1^\lambda)$**:** Takes a security parameter 1^λ and generates $\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda)$ and $\mathbf{B} \xleftarrow{\mathsf{U}} \mathsf{GL}_7(\mathbb{Z}_p)$. Outputs

$$\mathsf{pk} := (\mathbb{G}, [\mathbf{b}_1]_1, \ldots, [\mathbf{b}_4]_1), \;\; \mathsf{msk} := (\mathbf{b}_1^*, \ldots, \mathbf{b}_4^*),$$

where $\mathbf{b}_i$ (resp. $\mathbf{b}_j^*$) denotes the i-th row of $\mathbf{B}$ (resp. j-th row of $\mathbf{B}^*$).

$\mathsf{Enc}(\mathsf{pk}, \mathbf{x})$**:** Takes pk and $\mathbf{x} := (x_1, \ldots, x_m) \in \mathbb{Z}^m$ where $m = m(\lambda)$ is any polynomial. Defines $\mathbf{c}_i := (\pi_i(1, i), x_i, z, 0, 0, 0)\mathbf{B} \in \mathbb{Z}_p^7$ for all $i \in [m]$, where $\pi_i, z \xleftarrow{\mathsf{U}} \mathbb{Z}_p$. Outputs

$$\mathsf{ct}_m := ([\mathbf{c}_1]_1, \ldots, [\mathbf{c}_m]_1).$$

$\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, S, \mathbf{y})$**:** Takes pk, msk, a non-empty index set $S \subseteq [s]$ where $s = s(\lambda)$ is any polynomial, and an indexed vector $\mathbf{y} := (y_i)_{i \in S} \in \mathbb{Z}^S$. Chooses $\{r_i\}_{i \in S} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p$ s.t. $\sum_{i \in S} r_i = 0$ and $\rho_i \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p$, and defines $\mathbf{k}_i := (\rho_i(-i, 1), y_i, r_i, 0, 0, 0)\mathbf{B}^* \in \mathbb{Z}_p^7$ for all $i \in S$. Outputs

$$\mathsf{sk}_S := (S, \{[\mathbf{k}_i]_2\}_{i \in S}).$$

$\mathsf{Dec}(\mathsf{pk}, \mathsf{ct}_m, \mathsf{sk}_S)$**:** Takes pk, a ciphertext ct_m for m dimensional vector, and a secret key sk_S for a index set S. If $S \subseteq [m]$, then computes

$$h := \prod_{i \in S} e([\mathbf{c}_i]_1, [\mathbf{k}_i]_2),$$

and searches for d s.t. $e(g_1, g_2)^d = h$ exhaustively in the range of $-|S|X_\lambda Y_\lambda$ to $|S|X_\lambda Y_\lambda$. If such d is found, outputs d. Otherwise, outputs $\perp$.

Correctness Observe that if $S \subseteq [m]$,

$$h = \prod_{i \in S} e([\mathbf{c}_i]_1, [\mathbf{k}_i]_2) = e(g_1, g_2)^{\sum_{i \in S} \langle \mathbf{c}_i, \mathbf{k}_i \rangle} = e(g_1, g_2)^{\sum_{i \in S}(x_i y_i + z r_i)}.$$

Here we have $\sum_{i \in S} r_i = 0$, then $h = e(g_1, g_2)^{\sum_{i \in S} x_i y_i}$. If $||\mathbf{x}||_\infty \le X_\lambda$ and $||\mathbf{y}||_\infty \le Y_\lambda$, then $|\sum_{i \in S} x_i y_i| \le |S| X_\lambda Y_\lambda$ and Dec outputs $\sum_{i \in S} x_i y_i$.

4.2 Security

Theorem 4.1. *Assume that the SXDH assumption holds, then our Pub-UIPFE is adaptively secure. More formally, let m_{max} be the maximum length of the challenge vector that $\mathcal{A}$ outputs and s_{max} be the maximum index with which $\mathcal{A}$ queries the key generation oracle, then for any PPT adversary $\mathcal{A}$ and security parameter λ, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Pub\text{-}UIPFE}}(\lambda) \le \{16 m_{\mathsf{max}}^2 + 8 m_{\mathsf{max}}(s_{\mathsf{max}} - 1) + 4\} \mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda) + 2^{-\Omega(\lambda)}.$$

Proof outline The top-level strategy of the security proof is simple. Consider a world where an encryption algorithm could magically generate unbounded random dual orthonormal bases for each index. Then we observe that only one loop of the Game 3 sequence in the Priv-UIPFE scheme suffices for the Pub-UIPFE scheme because there is one challenge ciphertext query and no challenge secret key query. To generate such a situation, we utilize an entropy-amplification technique like [26] and show that PPT adversaries cannot distinguish the real world from the "magical" world under the SXDH assumption. In the following, we provide a more concrete overview of the proof. Similarly to the Game 3 sequence in the Priv-UIPFE scheme, we first change the challenge ciphertext and all secret keys into the following form,

$$\tilde{z}, \{\tilde{r}_{\ell,i}\}_{i \in S_\ell} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p \text{ s.t. } \sum_{i \in S_\ell} \tilde{r}_{\ell,i} = 0,$$

$$\mathbf{c}_i := (\pi_i(1, i), x_i^0, z, \boxed{\tilde{z}}, 0, 0)\mathbf{B}, \;\; \mathbf{k}_{\ell,i} := (\rho_{\ell,i}(-i, 1), y_{\ell,i}, r_{\ell,i}, \boxed{\tilde{r}_{\ell,i}}, 0, 0)\mathbf{B}^*.$$

Next, we change $\mathbf{k}_{\ell,i}$ for all ℓ s.t. $(\max S_\ell > m') \wedge (\min S_\ell \leq m')$, where m' is the guess of the vector length for the challenge ciphertext, as

$$\{\bar{r}_{\ell,i}\}_{i \in S_\ell} \xleftarrow{\mathsf{U}} \mathbb{Z}_p, \ \ \mathbf{k}_{\ell,i} := (\rho_{\ell,i}(-i,1), y_{\ell,i}, r_{\ell,i}, \boxed{\bar{r}_{\ell,i}}, 0, 0)\mathbf{B}^*. \tag{3}$$

Then, we change $\mathbf{c}_i$ as

$$\mathbf{c}_i := (\pi_i(1,i), \boxed{x_i^1}, z, \tilde{z}, 0, 0)\mathbf{B}, \tag{4}$$

similar to Priv-UIPFE. The remaining sequence is just the reverse. In the case of the Priv-UIPFE scheme, recall that we perform distinct basis changes for each index in the steps of Eq.(3) and Eq.(4). However, we cannot perform such basis changes in Pub-UIPFE, because all indices share the same dual orthonormal bases. To overcome this difficulty, we conduct this step by computational change on the basis of the SXDH assumption. Specifically, we introduce the following two lemmas and use them in the proof as a kind of basis change in Priv-UIPFE. Especially, it is relatively easy to see that Lemma 4.2 can be used for showing that PPT adversaries cannot distinguish the real world, i.e., $\beta = 0$, from the "magical" world, i.e., $\beta = 1$, where dual orthonormal bases for each index are "somewhat" random. In other words, in the case of $\beta = 1$, dual orthonormal bases for index i is generated as

$$\mathbf{D}_i := \begin{pmatrix} \mathbf{I}_2 & & & & \\ & 1 & & & \\ & & 1 & & \\ & w_i & & 1 & \\ & & & & \mathbf{I}_2 \end{pmatrix} \mathbf{B}, \ \ \mathbf{D}_i^* := \begin{pmatrix} \mathbf{I}_2 & & & & \\ & 1 & & -w_i & \\ & & 1 & & \\ & & & 1 & \\ & & & & \mathbf{I}_2 \end{pmatrix} \mathbf{B}^*. \tag{5}$$

Lemma 4.1 is used for the step of Eq.(3), which corresponds to Games 3-ν-2 and 3-ν-3 in the proof of Priv-UIPFE, and Lemma 4.2 is used for the step of Eq.(4), which corresponds to Game 3-ν-4 in the proof of Priv-UIPFE. In our Pub-UIPFE scheme, there are three-dimensional subspaces that are not used in the actual function: the 5-7th spaces. The fifth space is a kind of a semi-functional space that is similar to the fourth space of our Priv-UIPFE scheme. The sixth and seventh spaces are necessary to amplify the entropy of the two dimensional prefix for the proof of the lemmas. Similar to here, adding extra spaces other than the semi-functional space and amplifying the entropy in the space are also done in [11, 23, 26].

Lemma 4.1. *For any polynomial* $m := m(\lambda)$ *and* $n := n(\lambda)$*, we define the following distribution,*

$$\begin{aligned}
&\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda),\ \ \mathbf{B} \xleftarrow{\mathsf{U}} \mathsf{GL}_7(\mathbb{Z}_p),\ \ \{\pi_i\}_{i\in[m]}, \tilde{z} \xleftarrow{\mathsf{U}} \mathbb{Z}_p,\\
&\mathbf{u}_i := (\pi_i(1,i), 0, 0, \tilde{z}, 0, 0)\mathbf{B} \ \text{ for all } i \in [m],\\
&D := (\mathbb{G}, [\mathbf{b}_1]_1, \ldots, [\mathbf{b}_4]_1, [\mathbf{b}_1^*]_2, \ldots, [\mathbf{b}_5^*]_2, [\mathbf{u}_1]_1, \ldots, [\mathbf{u}_m]_1),\\
&\{\rho'_i\}_{i\in[m+1,n]}, \{r'_i\}_{i\in[m+1,n]} \xleftarrow{\mathsf{U}} \mathbb{Z}_p,\\
&\mathbf{u}^*_{i,\beta} := (\rho'_i(-i,1), 0, 0, \beta r'_i, 0, 0)\mathbf{B}^* \ \text{ for all } i \in [m+1,n],\\
&U_\beta := \{[\mathbf{u}^*_{i,\beta}]_2\}_{i\in[m+1,n]}.
\end{aligned}$$

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

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{P1}}_{\mathcal{A}}(\lambda) :=& |\mathsf{Pr}[1 \leftarrow \mathcal{A}(D, U_0)] - \mathsf{Pr}[1 \leftarrow \mathcal{A}(D, U_1)]|\\
\leq& 4(n-m)\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.
\end{aligned}$$

Lemma 4.2. *For any polynomial* $m := m(\lambda)$ *and* $n := n(\lambda)$*, we define the following distribution,*

$$\begin{aligned}
&\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda),\ \ \mathbf{B} \xleftarrow{\mathsf{U}} \mathsf{GL}_7(\mathbb{Z}_p),\ \ \{\rho'_i\}_{i\in[m+1,n]} \xleftarrow{\mathsf{U}} \mathbb{Z}_p,\\
&\mathbf{u}^*_i := (\rho'_i(-i,1), 1, 0, 0, 0, 0)\mathbf{B}^* \ \text{ for all } i \in [m+1,n],\\
&D := (\mathbb{G}, [\mathbf{b}_1]_1, \ldots, [\mathbf{b}_4]_1, [\mathbf{b}_1^*]_2, [\mathbf{b}_2^*]_2, [\mathbf{b}_4^*]_2, [\mathbf{b}_5^*]_2, \{[\mathbf{u}^*_i]_2\}_{i\in[m+1,n]}),\\
&\{\pi'_i\}_{i\in[m]}, \{\rho'_i\}_{i\in[m]}, \{w_i\}_{i\in[m]} \xleftarrow{\mathsf{U}} \mathbb{Z}_p,\\
&\mathbf{u}_{i,\beta} := (\pi'_i(1,i), \beta w_i, 0, 1, 0, 0)\mathbf{B} \ \text{ for all } i \in [m],\\
&\mathbf{u}^*_{i,\beta} := (\rho'_i(-i,1), 1, 0, -\beta w_i, 0, 0)\mathbf{B}^* \ \text{ for all } i \in [m],\\
&U_\beta := \{[\mathbf{u}_{i,\beta}]_1, [\mathbf{u}^*_{i,\beta}]_2\}_{i\in[m]}.
\end{aligned}$$

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

$$\mathsf{Adv}^{\mathsf{P2}}_{\mathcal{A}}(\lambda) := |\mathsf{Pr}[1 \leftarrow \mathcal{A}(D, U_0)] - \mathsf{Pr}[1 \leftarrow \mathcal{A}(D, U_1)]| \leq 8m\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

The formal proofs of Theorem 4.1, Lemma 4.1, and Lemma 4.2 are presented in the full version of this paper.

Acknowledgments

We are very grateful to Pratish Datta and Tatsuaki Okamoto for giving us a chance to start this work. We also would like to thank anonymous reviewers for their helpful comments.

References

1. Abdalla, M., Bourse, F., De Caro, A., Pointcheval, D.: Simple functional encryption schemes for inner products. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 733–751. Springer, Heidelberg (Mar / Apr 2015)
2. Abdalla, M., Bourse, F., De Caro, A., Pointcheval, D.: Better security for functional encryption for inner product evaluations. Cryptology ePrint Archive, Report 2016/011 (2016), `http://eprint.iacr.org/2016/011`
3. Abdalla, M., Catalano, D., Fiore, D., Gay, R., Ursu, B.: Multi-input functional encryption for inner products: Function-hiding realizations and constructions without pairings. Cryptology ePrint Archive, Report 2017/972 (2017), `http://eprint.iacr.org/2017/972`
4. Abdalla, M., Gay, R., Raykova, M., Wee, H.: Multi-input inner-product functional encryption from pairings. In: Coron, J., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part I. LNCS, vol. 10210, pp. 601–626. Springer, Heidelberg (Apr / May 2017)
5. Agrawal, S., Libert, B., Stehlé, D.: Fully secure functional encryption for inner products, from standard assumptions. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part III. LNCS, vol. 9816, pp. 333–362. Springer, Heidelberg (Aug 2016)
6. Attrapadung, N.: Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 557–577. Springer, Heidelberg (May 2014)
7. Attrapadung, N.: Dual system encryption framework in prime-order groups via computational pair encodings. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part II. LNCS, vol. 10032, pp. 591–623. Springer, Heidelberg (Dec 2016)
8. Bishop, A., Jain, A., Kowalczyk, L.: Function-hiding inner product encryption. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015, Part I. LNCS, vol. 9452, pp. 470–491. Springer, Heidelberg (Nov / Dec 2015)
9. Boneh, D., Sahai, A., Waters, B.: Functional encryption: Definitions and challenges. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 253–273. Springer, Heidelberg (Mar 2011)
10. Brakerski, Z., Vaikuntanathan, V.: Circuit-ABE from LWE: Unbounded attributes and semi-adaptive security. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part III. LNCS, vol. 9816, pp. 363–384. Springer, Heidelberg (Aug 2016)
11. Chen, J., Gong, J., Kowalczyk, L., Wee, H.: Unbounded ABE via bilinear entropy expansion, revisited. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part I. LNCS, vol. 10820, pp. 503–534. Springer, Heidelberg (Apr / May 2018)
12. Chen, J., Wee, H.: Semi-adaptive attribute-based encryption and improved delegation for Boolean formula. In: Abdalla, M., Prisco, R.D. (eds.) SCN 14. LNCS, vol. 8642, pp. 277–297. Springer, Heidelberg (Sep 2014)
13. Datta, P., Dutta, R., Mukhopadhyay, S.: Functional encryption for inner product with full function privacy. In: Cheng, C.M., Chung, K.M., Persiano, G., Yang, B.Y. (eds.) PKC 2016, Part I. LNCS, vol. 9614, pp. 164–195. Springer, Heidelberg (Mar 2016)
14. Datta, P., Okamoto, T., Tomida, J.: Full-hiding (unbounded) multi-input inner product functional encryption from the k-linear assumption. In: Abdalla, M., Dahab, R. (eds.) PKC 2018, Part II. LNCS, vol. 10770, pp. 245–277. Springer, Heidelberg (Mar 2018)

15. Dufour Sans, E., Pointcheval, D.: Unbounded inner product functional encryption, with succinct keys. Cryptology ePrint Archive, Report 2018/487 (2018), https://eprint.iacr.org/2018/487
16. Garg, S., Gentry, C., Halevi, S.: Candidate multilinear maps from ideal lattices. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 1–17. Springer, Heidelberg (May 2013)
17. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. In: 54th FOCS. pp. 40–49. IEEE Computer Society Press (Oct 2013)
18. Garg, S., Gentry, C., Halevi, S., Zhandry, M.: Functional encryption without obfuscation. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016-A, Part II. LNCS, vol. 9563, pp. 480–511. Springer, Heidelberg (Jan 2016)
19. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Juels, A., Wright, R.N., Vimercati, S. (eds.) ACM CCS 06. pp. 89–98. ACM Press (Oct / Nov 2006), available as Cryptology ePrint Archive Report 2006/309
20. Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 146–162. Springer, Heidelberg (Apr 2008)
21. Kim, S., Lewi, K., Mandal, A., Montgomery, H., Roy, A., Wu, D.J.: Function-hiding inner product encryption is practical. Cryptology ePrint Archive, Report 2016/440 (2016), http://eprint.iacr.org/2016/440
22. Kim, S., Kim, J., Seo, J.H.: A new approach for practical function-private inner product encryption. Cryptology ePrint Archive, Report 2017/004 (2017), http://eprint.iacr.org/2017/004
23. Lewko, A.B., Waters, B.: Unbounded HIBE and attribute-based encryption. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 547–567. Springer, Heidelberg (May 2011)
24. Lin, H.: Indistinguishability obfuscation from SXDH on 5-linear maps and locality-5 PRGs. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part I. LNCS, vol. 10401, pp. 599–629. Springer, Heidelberg (Aug 2017)
25. Okamoto, T., Takashima, K.: Fully secure functional encryption with general relations from the decisional linear assumption. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 191–208. Springer, Heidelberg (Aug 2010)
26. Okamoto, T., Takashima, K.: Fully secure unbounded inner-product and attribute-based encryption. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 349–366. Springer, Heidelberg (Dec 2012)
27. O'Neill, A.: Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556 (2010), http://eprint.iacr.org/2010/556
28. Tomida, J., Abe, M., Okamoto, T.: Efficient functional encryption for inner-product values with full-hiding security. In: Bishop, M., Nascimento, A.C.A. (eds.) ISC 2016. LNCS, vol. 9866, pp. 408–425. Springer, Heidelberg (Sep 2016)
29. Waters, B.: Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (Aug 2009)
30. Waters, B.: A punctured programming approach to adaptively secure functional encryption. In: Gennaro, R., Robshaw, M.J.B. (eds.) CRYPTO 2015, Part II. LNCS, vol. 9216, pp. 678–697. Springer, Heidelberg (Aug 2015)
31. Wee, H.: Dual system encryption via predicate encodings. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 616–637. Springer, Heidelberg (Feb 2014)