A Simple Inner-Product Functional Encryption Scheme from the Inverse-DDH Assumption

Jim Barthel[1], Răzvan Roşie[1], and Rajeev Anand Sahu[1]

University of Luxembourg, Luxembourg
first_name.last_name@uni.lu

Abstract. We propose a new and straightforward functional encryption scheme for the bounded-norm inner-product functionality (IPFE) in the public-key settings. We prove the security of the proposed scheme with respect to standard assumptions. Specifically, our construction is secure under the Inverse Decisional Diffie-Hellman computational hardness assumption (DDHI), which is not known to imply or be implied by DDH. The proof technique chiefly exploits the algebraic properties of matrices, viewed as linear maps between vector spaces.

Keywords: functional encryption, Inverse-DDH, inner-product encryption

1 Introduction

Functional encryption (FE), formalized through the works of O'Neill[O'N10] and Boneh, Sahai and Waters [BSW11], came up as an ambitious cryptographic paradigm allowing for surgical access over encrypted data: given a ciphertext CT encrypting some input message (i.e. plaintext) m, the holder of a token (or functional key) sk_f can determine the value of the function f evaluated on the plaintext m, and nothing else about m (except for the maybe its size). Due to the wide range of potential applications, functional encryption has been regarded as a powerful tool when compared to the existing, classical, well-established cryptographic primitives, such as public-key encryption or identity-based encryption. Goldwasser et al. [GKP+13] mention data-mining techniques such as searching for keywords over encrypted data as a potential utilization of FE in practice. Another major workstream study the relationship between functional encryption and indistinguishability obfuscation [GGH+13,AJ15,BV15,AFH+16,LPST16,JLS20].

With respect to its derivatives, the original definitional landscape for FE has been extended to functional signatures (FS) and functional pseudorandom functions, which were introduced with the work of Boyle, Goldwasser and Ivan [BGI14]. The aim of their work is to capture the counterpart of functional encryption with reference to signing capabilities (one signs the value of a function f applied on a message m, the resulting signature being verifiable under the master public-key).

With respect to the prior and actual research status on functional encryption, we identify two major spheres: (1) a more general setting trying to achieve FE

schemes that support an *unbounded* number of functional keys; such works investigate generic theoretical implications/separations between different kinds of primitives — such as fully-homomorphic encryption (FHE), attribute-based encryption (ABE) [BGG+14] and FE [GKP+13] — or between different "flavours" of functional encryption: [LPST16,KS17,JLS20] study the relation between private-key and public-key functional encryption, or between single-input and multi-input schemes [BKS16]; (2) a second group of works concentrates on constructions targeting simple functionalities (i.e. inner-product or quadratic FE), such as the ones presented in [ABDP15,BBL17,BJK15,DDM16,Lin16,BCFG17,Tom], with their security relying on well-established assumption.

Inner-Product Functional Encryption. In terms of its semantics, an FE scheme for the simple inner-product functionality (IPFE) with a bounded norm can be *informally* described as follows: a plaintext is represented through a vector $\mathbf{x}$ and its corresponding ciphertext is $\mathsf{CT}_{\mathbf{x}}$. A functional key $\mathrm{sk}_{\mathbf{y}}$ is issued for each corresponding vector $\mathbf{y}$. Decryption takes as input the pair $(\mathsf{CT}_{\mathbf{x}}, \mathrm{sk}_{\mathbf{y}})$, the output being $\mathbf{x}^{\top} \cdot \mathbf{y}$. Despite its simplicity, an inner product functionality may found attractive applications: as a **motivational example**, assume a national fiscal agency stores the monthly incomes of the citizens of a country in an encrypted format (in order to prevent potential leaks from the highly sensitive data). Suppose that at some point, the government would like to introduce a differential tax scale taking into consideration the cumulative income (e.g 1% for January, 2% for February, ..., 30% for December)[1]. In order to simulate the impact of these measures to the population and to the budget execution for a year, an audit company would like to query the database without knowing the actual financial assets of citizens. Turning the layman description into one closer to a program specification, we would represent the encrypted incomes as a 12-dimensional vector $\mathbf{x}$, and the tax percents as a vector $\mathbf{y} \leftarrow (1, 2, \ldots, 30)$. Assuming that no annual global income is greater than B, one can compute the required queries through the means of an inner-product functional encryption scheme. Thus, the actual data in the newly planned-taxes may be revealed or not to an audit company. If there is no concern in revealing such data, an IPFE scheme would suffice. However, if $\mathbf{y}$ needs to be protected, a function-hiding search (one that hides $\mathbf{y}$) must be performed.

Depending on the application, one may also demand that an FE scheme satisfies *function privacy*: a guarantee that the functional key sk_f does not reveal the function f that is considered, in addition to the original s-IND-FE-CPA security notion [DDM16]. For the case of IPFE, this translates into preventing an adversary from distinguishing between the values of $\mathbf{y}$ into $\mathrm{sk}_{\mathbf{y}}$ as well as the value of the plaintext. As stated in [BJK15], function-hiding FE cannot be achieved in the public-key setting: an adversary can obtain the keys for two different functions f_1 and f_2, then using the public mpk it can obtain a distinguishing CT_{m} such that $f_1(\mathsf{m}) \neq f_2(\mathsf{m})$.

[1] For instance, such a government would like to gain more from the interests obtained through bank deposits, which accumulates over the year.

1.1 Our contributions

Despite the simplicity of the construction we propose, we believe the main contribution of this work relies on the proof techniques. We reuse an *agile algebraic* technicality showed by Hofheinz and Jager in [HJ16], and we give a brief overview in what follows. Assume a vector $\mathbf{u}$ lies in a subspace $\mathcal{U}$. One can implement a linear map $\mathbf{M} : \mathcal{U} \to \mathcal{S}$, as follows: (1) build $\mathbf{C}$, a matrix mapping $\mathcal{W} \to \mathcal{U}$, where $\mathcal{W}$ is the subspace spanned by the first $n-1$ canonical vectors (having the element in position n set to 0); (2) build $\mathbf{C}'$ as a matrix that maps $\mathcal{W} \to \mathcal{S}$. Then set $\mathbf{M} \leftarrow \mathbf{C}^{-1} \cdot \mathbf{C}'$, which will map $\mathcal{U} \xrightarrow{\mathbf{C}^{-1}} \mathcal{W} \xrightarrow{\mathbf{C}'} \mathcal{S}$, as depicted in Figure 1.

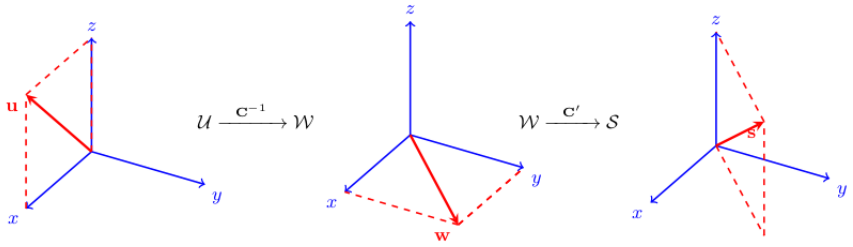

Fig. 1. Assuming a vector space of dimension 3 over $\mathbb{Z}_p$, the matrix $\mathbf{C}'$ will map entries in $\mathcal{W}$ (having $z = 0$) to the vector space defined by the first 2 rows in $\mathbf{C}'$. Similarly one can imagine $\mathbf{C} : \mathcal{W} \to \mathcal{U}$ (thus $\mathbf{C}^{-1} : \mathcal{U} \to \mathcal{W}$) and compute its inverse, in order to set $\mathbf{M} \leftarrow \mathbf{C}^{-1} \cdot \mathbf{C}'$.

Our contribution (Section 3) is a (plain) selectively-secure IPFE scheme. We use the notations introduced in [EHK+13], and define $[a] := g^a$, where g is a cyclic group generator and a is an element over $\mathbb{Z}_p$. The bulk of the construction can be characterized as follows: (1) the msk is set to a uniformly sampled matrix $\mathbf{T}$, while the mpk is the group encoded version of its inverse, namely $[\mathbf{T}^{-1}]$; (2) encryption is done by sampling a random vector $\mathbf{t}$ over $\mathbb{Z}_p^n$, "masking" $\mathbf{x}$ as $\mathbf{x} + \mathbf{t}$ and encoding in the group as: $[\mathbf{x} + \mathbf{t}]$. Additionally, the encoding of $\mathbf{t}^\top \cdot \mathbf{T}^{-1}$ is released; (3) to decrypt, one needs to know $\mathbf{y}$, together with $\mathbf{y}^\top \cdot \mathbf{T}$. Then the decryption computes $[\mathbf{y}^\top \cdot (\mathbf{t} + \mathbf{x}) - \mathbf{y}^\top \cdot \mathbf{t}]$ in order to finally get the DLOG($[\mathbf{x}^\top \cdot \mathbf{y}]$)[2]. The very interesting bit comes with the s-IND-FE-CPA proof: we instantiate an *auxiliary* matrix $\mathbf{T}'$ via the **DDHI** assumption (Definition 5) by implicitly setting the last row in $\mathbf{T}'$ to be of the form: $(0, \ldots, 0, a)$, where $[a]$ is taken from the **DDHI**-tuple. Essentially, this prevents an adversary deriving $\mathsf{sk}_\mathbf{y}$ where $\mathbf{y} \perp (\mathbf{x}^L - \mathbf{x}^R)$ — we achieve this by left-multiplying $\mathbf{T}'$ with a linear map $\mathbf{L} : (\mathbf{x}^L - \mathbf{x}^R) \to \mathcal{W}$ using a technique developed in [HJ16]. Most importantly, given

[2] Solving DLOG to recover inner-products, while assuming they have bounded norm is a technique used in the existing works on IPFE.

the form of $\mathbf{T}'$, we can always set its inverse (which will contain the challenge $[a^{-1}$ or $\$]$) only in its last column.

2 Definitions

Conventions. We denote by $s \leftarrow\!\!\$\ S$ the fact that s is picked uniformly at random from a finite set S. Variables in bold capital letters stand for matrices (e.g. $\mathbf{M}$) while bold lowercase letters represent vectors (e.g. $\mathbf{u}$). A subscript i on a vector $\mathbf{u}$ (e.g. $\mathbf{u}_i$) stands for the i-th component of the vector. An analogue convention is used for matrices. By $[a] \coloneqq g^a$ we denote the "encoding of an element" w.r.t. a group generator $g \in \mathbb{G}$, while through $[\mathbf{M}]$ and $[\mathbf{u}]$, we denote the encodings of a matrix, respectively vector. $\overline{\mathbf{W}}$ denotes the matrix formed by the top $n-1$ *rows* of a matrix $\mathbf{W}$ of size $n \times n$. When working with a family of vectors $\mathbf{v}$, we use the upper script to distinguish between them: $\mathbf{v}^{(0)}, \mathbf{v}^{(1)}, \ldots$. We abuse notation and extend it to bilinear maps by writing $e([\mathbf{A}], [\mathbf{B}]) = e(g,g)^{\mathbf{A}\cdot\mathbf{B}} = [\mathbf{A}\cdot\mathbf{B}]$ to denote the matrix obtained after multiplying the exponents and getting as a result the pairing of entries. By $C(\mathbf{A})$, we denote the columnspace of a matrix $\mathbf{A}$, and by $C(\mathbf{A}^\top)$, we denote its rowspace. We denote the security parameter by $\lambda \in \mathbb{N}$ and we assume it is given to all algorithms in the unary representation 1^λ. We regard an algorithm as being randomized (unless stated) and being modeled by a Turing machine. PPT as usual stands for "probabilistic polynomial-time." Given a randomized algorithm $\mathcal{A}$ we denote the action of running $\mathcal{A}$ on input(s) $(1^\lambda, x_1, \ldots)$ with uniform random coins r and assigning the output(s) to $(y_1, \ldots)$ by $(y_1, \ldots) \leftarrow\!\!\$\ \mathcal{A}(1^\lambda, x_1, \ldots; r)$. We denote the set of all negligible functions by Negl. With $\bar{x} \prec x$, we denote a bitstring prefix.

2.1 Functional Encryption - Definitions

When considering the encryption setting, one should distinguish between two kinds of functional encryption schemes: (1) on the one hand, in public-key FE schemes, encryption happens through the means of a master public-key, while in (2) the private-key paradigm, access to a master private-key must be offered in order to compute a ciphertext. Such a difference is significant from multiple points of view: efficiency, malleability, use cases etc., but in our work, we focus exclusively on certain security notions (i.e. on indistinguishability). Since FE is a central notion of this work, we define here the two concepts. In terms of the security experiments, we provide the games for selective and adaptive functional versions of "chosen-plaintext attack" security. We also extend and cover function-hiding, with the remarkable implication that a function-hiding FE scheme achieves s-IND-FE-CPA-security [DDM16].

Definition 1 (Functional Encryption Scheme - Public-Key Setting). *A functional encryption scheme* **FE** *in the public-key setting consists of a tuple of algorithms* (**Gen**, **KDer**, **Enc**, **Dec**) *such that:*

1. $(\mathrm{msk}, \mathrm{mpk}) \leftarrow_{\$} \mathbf{FE.Gen}(1^{\lambda})$: *guven the unary representation of the security parameters* λ*, it outputs a pair of master secret/public keys.*
2. $\mathsf{CT} \leftarrow_{\$} \mathbf{FE.Enc}(\mathrm{mpk}, \mathsf{m})$*: the randomized encryption procedure encrypts the plaintext* m *with respect to the master key* mpk*.*
3. $\mathrm{sk}_f \leftarrow_{\$} \mathbf{FE.KDer}(\mathrm{msk}, f)$*: using the master secret key and the given functionality* f*, the (possibly randomized) key-derivation procedure outputs a functioal key* sk_f*.*
4. $\mathbf{FE.Dec}(\mathrm{sk}_f, \mathsf{CT})$ *decrypts the ciphertext* CT *using the functional key* sk_f *in order to learn a valid message* $f(\mathsf{m})$ *or a special symbol* $\perp$*, in case the decryption procedure fails.*

– *We say that a public-key* **FE** *scheme is correct if the following holds:*

$$\Pr\left[\mathbf{Dec}(\mathrm{sk}_f, \mathsf{CT}) = f(\mathsf{m}) \;\middle|\; \begin{array}{l}(\mathrm{msk}, \mathrm{mpk}) \leftarrow_{\$} \mathbf{Gen}(1^{\lambda}) \wedge \\ \mathsf{CT} \leftarrow_{\$} \mathbf{Enc}(\mathrm{mpk}, \mathsf{m}) \wedge \\ \mathrm{sk}_f \leftarrow_{\$} \mathbf{KDer}(\mathrm{msk}, f)\end{array}\right] \in 1 - \textsc{Negl}(\lambda) \ .$$

– *We say that a public-key* **FE** *scheme is (selective)* s-IND-FE-CPA*-secure if the advantage of any* PPT *adversary* $\mathcal{A}$ *against the* s-IND-FE-CPA*-game defined in Figure 2 is negligible:*

$$\mathbf{Adv}^{\text{s-IND-FE-CPA}}_{\mathcal{A},\mathbf{FE}}(\lambda) \coloneqq \left|\Pr\left[\text{s-IND-FE-CPA}^{\mathcal{A}}_{\mathbf{FE}}(\lambda) = 1\right] - \frac{1}{2}\right| \in \textsc{Negl}(\lambda) \ .$$

Similarly, we say that it is (adaptive) IND-FE-CPA*-secure if:*

$$\mathbf{Adv}^{\text{IND-FE-CPA}}_{\mathcal{A},\mathbf{FE}}(\lambda) \coloneqq \left|\Pr\left[\text{IND-FE-CPA}^{\mathcal{A}}_{\mathbf{FE}}(\lambda) = 1\right] - \frac{1}{2}\right| \in \textsc{Negl}(\lambda) \ .$$

Private-Key Setting. Originally, FE has been introduced in the public-key setting, and was regarded as the most general extension of the PKE paradigm. However, the private-key counterpart was recently related to the public-key framework via a series of results [BKS16,KS17], which increases the interest in the primitive[3]. For completeness, we provide the definition of the scheme below:

Definition 2 (Functional Encryption Scheme - Private-Key Setting). *A functional encryption scheme* **FE** *in the private-key setting consists of a tuple of* PPT *algorithms* (**Gen**, **KDer**, **Enc**, **Dec**) *such that:*

1. $\mathrm{msk} \leftarrow_{\$} \mathbf{FE.Gen}(1^{\lambda})$: *takes as input the unary representation of the security parameters and outputs* msk*.*
2. $\mathrm{sk}_f \leftarrow_{\$} \mathbf{FE.KDer}(\mathrm{msk}, f)$*: given the master secret key and a function* f*, the (randomized) key-derivation procedure outputs a corresponding* sk_f*.*
3. $\mathsf{CT} \leftarrow_{\$} \mathbf{FE.Enc}(\mathrm{msk}, \mathsf{m})$*: the randomized encryption procedure encrypts the plaintext* m *with respect to* msk*.*

[3] Although this primitive may found concrete practical applications depending on the use cases.

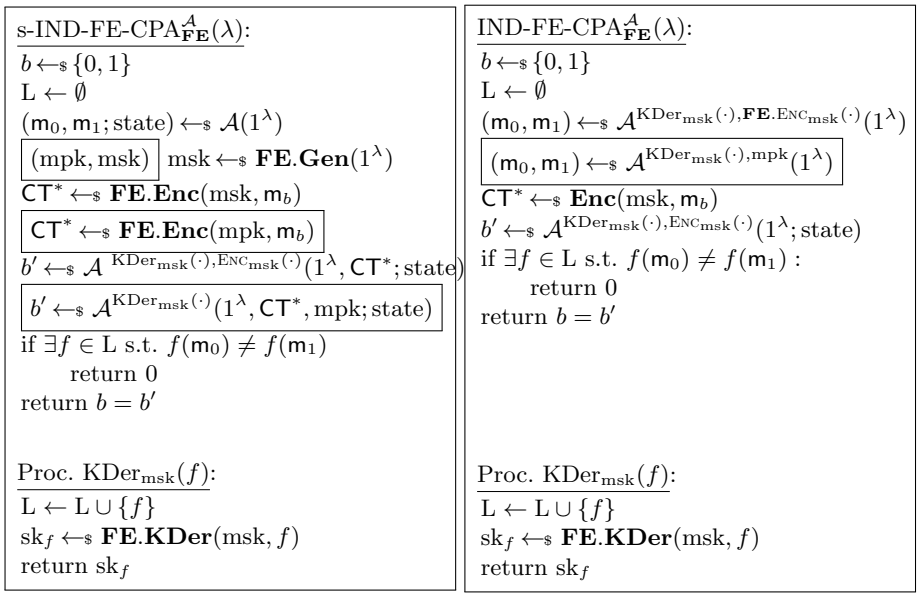

Fig. 2. The selective and adaptive indistinguishability experiment defined for a functional encryption scheme. The difference between the private-key and the public-key settings are marked in boxed lines of codes, corresponding to the latter notion.

4. $\mathbf{FE.Dec}(\mathrm{sk}_f, \mathsf{CT})$ *decrypts the* CT *using the functional key* sk_f *in order to learn a valid message* $f(\mathsf{m})$ *or a special symbol* $\perp$*, in case the decryption procedure fails.*

– *We say that a private-key* **FE** *scheme is correct if:*

$$\Pr\left[\mathbf{Dec}(\mathrm{sk}_f, \mathsf{CT}) = f(\mathsf{m}) \;\middle|\; \begin{array}{l} \mathrm{msk} \leftarrow\!\!\$ \mathbf{Gen}(1^\lambda) \wedge \\ \mathsf{CT} \leftarrow\!\!\$ \mathbf{Enc}(\mathrm{msk}, \mathsf{m}) \wedge \\ \mathrm{sk}_f \leftarrow\!\!\$ \mathbf{KDer}(\mathrm{msk}, f) \end{array}\right] = 1 - \textsc{Negl}(\lambda) \ .$$

– *A private-key* **FE** *scheme is (selective)* s-IND-FE-CPA*-secure if the advantage of any* PPT *adversary* $\mathcal{A}$ *against the* s-IND-FE-CPA*-game defined in Figure 2 is negligible:*

$$\mathbf{Adv}^{\text{IND-FE-CPA}}_{\mathcal{A},\mathbf{FE}}(\lambda) := \left|\Pr\left[\text{IND-FE-CPA}^{\mathcal{A}}_{\mathbf{FE}}(\lambda) = 1\right] - \frac{1}{2}\right| \in \textsc{Negl}(\lambda) \ .$$

Similarly, we say that it is (adaptive) IND-FE-CPA*-secure if:*

$$\mathbf{Adv}^{\text{IND-FE-CPA}}_{\mathcal{A},\mathbf{FE}}(\lambda) := \left|\Pr\left[\text{IND-FE-CPA}^{\mathcal{A}}_{\mathbf{FE}}(\lambda) = 1\right] - \frac{1}{2}\right| \in \textsc{Negl}(\lambda) \ .$$

Inner-Product Functional Encryption. As mentioned in the introduction, constructing functional encryption for general-purpose circuits is an open

problem. We formally define the bounded-norm inner-product functionality in what follows:

Definition 3 (Bounded-Norm Inner-Product Functionality). *Let* $(\mathbf{x}, \mathbf{y}) \in \mathbb{Z}_p^{2n}$ *and* $B < p$. *We define* $IP_B(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \cdot \mathbf{y}$, *when* $\mathbf{x}^\top \cdot \mathbf{y} \leq B$ *and* $IP_B(\mathbf{x}, \mathbf{y}) = \perp$ *otherwise.*

Function-Hiding IPFE Schemes The function hiding property for IPFE has been considered in the work of [BJK15]. The goal is to hide the function that is being computed from the view of a computationally bounded adversary. The security games are defined in Figure 3.

Definition 4 (Function Hiding - Private-Key Setting). *A private-key* **FE** *scheme is* FHIDE*-secure if the advantage of any* PPT *adversary* $\mathcal{A}$ *against the* FHIDE*-game defined in Figure 3 is negligible:*

$$\mathbf{Adv}^{\text{FHIDE}}_{\mathcal{A},\mathbf{FE}}(\lambda) \coloneqq \left| \Pr\left[\text{FHIDE}^{\mathcal{A}}_{\mathbf{FE}}(\lambda) = 1\right] - \frac{1}{2} \right| \in \text{NEGL}(\lambda) \ .$$

$\underline{\text{FHIDE}^{\mathcal{A}}_{\mathbf{FE}}(\lambda)}$:
$b \leftarrow\!\!\$ \{L, R\}$
$\text{L}_{\mathbf{Enc}} \leftarrow \emptyset$
$\text{L}_{\mathbf{KDer}} \leftarrow \emptyset$
$\text{msk} \leftarrow \mathbf{FE.Gen}(1^\lambda)$
$b' \leftarrow\!\!\$\ \mathcal{A}^{\text{ENC}_{\text{msk}}(\cdot), \text{KDer}_{\text{msk}}(\cdot)}(1^\lambda)$
return $b = b' \wedge \text{Valid}(\text{L}_{\mathbf{Enc}}, \text{L}_{\mathbf{KDer}})$

$\underline{\text{Valid}(\text{L}_{\mathbf{Enc}}, \text{L}_{\mathbf{KDer}})}$:
$\forall (\mathsf{m}_i^L, \mathsf{m}_i^R) \times (\text{sk}_{f_j}{}^L, \text{sk}_{f_j}{}^R) \in \text{L}_{\mathbf{Enc}} \times \text{L}_{\mathbf{KDer}}$:
  if $f_j^L(\mathsf{m}_i^L) \neq f_j^R(\mathsf{m}_i^R)$:
    return 0
return 1

$\underline{\text{ENC}_{\text{msk}}(\mathsf{m}^L, \mathsf{m}^R)}$:
$\text{L}_{\mathbf{Enc}} \leftarrow \text{L}_{\mathbf{Enc}} \cup \{(\mathsf{m}^L, \mathsf{m}^R)\}$
if $b = L$:
  $\mathsf{CT} \leftarrow\!\!\$\ \mathbf{Enc}(\text{msk}, \mathsf{m}^L)$
else:
  $\mathsf{CT} \leftarrow\!\!\$\ \mathbf{Enc}(\text{msk}, \mathsf{m}^R)$
return CT

$\underline{\text{KDer}_{\text{msk}}(f^L, f^R)}$:
$\text{L}_{\mathbf{KDer}} \leftarrow \text{L}_{\mathbf{KDer}} \cup \{(f^L, f^R)\}$
if $b = L$:
  $\text{sk}_f \leftarrow\!\!\$\ \mathbf{KDer}(\text{msk}, f^L)$
else:
  $\text{sk}_f \leftarrow\!\!\$\ \mathbf{KDer}(\text{msk}, f^R)$
return sk_f

Fig. 3. Function hiding in the private key setting.

Note that the function-hiding property implies indistinguishability (s-IND-FE-CPA) for free [DDM16].

2.2 Computational Hardness Assumptions

We define the computational hypothesis to be used throughout the next sections. The Inverse-DDH asks an adversary to distinguish between the inverse of an element and a randomly sampled one, in the presence of a computationally bounded adversary. This assumption is a restriction of the ℓ-**DDHI** assumption, introduced in [BDZ03].

Definition 5 (Decisional-Diffie Hellman Inversion Assumption). *Let $\mathbb{G}$ be a group of prime order p, $[1]$ a generator and $r \leftarrow_\$ \mathbb{Z}_p$. The following advantage of any* PPT *adversary $\mathcal{A}$ is negligible:*

$$\mathbf{Adv}^{\mathbf{DDHI}}_{\mathcal{A}}(\lambda) \coloneqq \Pr\left[\mathcal{A}([1],[a],[r]) = 1\right] - \Pr\left[\mathcal{A}([1],[a],[a^{-1}]) = 1\right] \in \text{NEGL}(\lambda) \ . \tag{1}$$

More recently, a framework of assumptions [EHK+13] were derived from the matrix-form of the **DDH** assumption. We define two of them.

Definition 6 ($\mathbf{U_k} - \mathbf{MDDH}$ [EHK+13]). *Let U_k stand for the uniform distribution defined over the sets of matrices of size $(k+1) \times k$ with elements over a cyclic group $\mathbb{G}$ of prime order p. Let $\mathbf{A}$ be a matrix sampled according to distribution U_k and $\mathbf{z} \leftarrow_\$ \mathbb{Z}_p^k$ and $\mathbf{r} \leftarrow_\$ \mathbb{Z}_p^{k+1}$. The following advantage of any* PPT *adversary $\mathcal{A}$ is negligible:*

$$\begin{aligned}\mathbf{Adv}^{\mathbf{U_k-MDDH}}_{\mathcal{A}}(\lambda) \coloneqq &\Pr\left[\mathcal{A}([\mathbf{z}],[\mathbf{A}],[\mathbf{A}\cdot\mathbf{z}]) = 1\right] - \\ &\Pr\left[\mathcal{A}([\mathbf{z}],[\mathbf{A}],[\mathbf{r}]) = 1\right] \in \text{NEGL}(\lambda) \ .\end{aligned} \tag{2}$$

Definition 7 ($\mathbf{n} - \mathbf{Rank}n$, [HJ16]). *Let $\mathbf{M}_i$ denote a matrix of rank i. Let $\mathbf{M}_n \leftarrow_\$ \mathbb{Z}_p^{n\times n}$ and $\mathbf{M}_{n-1} \leftarrow_\$ \left\{\mathbf{M} \in \mathbb{Z}_p^{n\times n} \mid rank(\mathbf{M}) = n-1\right\}$. Then, for any* PPT *adversary $\mathcal{A}$ the following holds:*

$$\mathbf{Adv}^{\mathcal{A}}_{\mathbf{n-Rank}}(\lambda) \coloneqq \Pr\left[1 \leftarrow_\$ \mathcal{A}([\mathbf{M}_n])\right] - \Pr\left[1 \leftarrow_\$ \mathcal{A}([\mathbf{M}_{n-1}])\right] \in \text{NEGL}(\lambda)$$

3 A Simple IPFE Scheme From DDH-Inversion Problem

In this section we introduce a simple inner-product functional encryption scheme in the public-key setting. We recall the ElGamal instantiation of the first inner-product FE scheme proposed in the literature [ABDP15], since its simplicity provides an illuminating example:

1. The master secret key consists of an n-dimensional vector $\mathbf{s}$ taken over $\mathbb{Z}_p$, while the master public-key is set to be the encoding of $\mathbf{s}$.
2. Encrypting the vector $\mathbf{x}$ is done by sampling a uniform r over $\mathbb{Z}_p$ and setting $\mathsf{CT} \leftarrow ([r], [r\mathbf{s} + \mathbf{x}])$.
3. The functional key $\mathrm{sk}_{\mathbf{y}}$ corresponding to $\mathbf{y}$ is deterministically generated and obtained as $\mathbf{s}^\top \cdot \mathbf{y}$;
4. The decryption steps through (a) getting $[r\mathbf{s}^\top \cdot \mathbf{y} + \mathbf{x}^\top \cdot \mathbf{y}]$, (b) getting $[r\mathbf{s}^\top \cdot \mathbf{y}]$, (c) subtracting the exponents in (b) from the ones in (a) and computing the discrete log — $\mathbf{DLP}\left(\left[\mathbf{x}^\top \cdot \mathbf{y}\right]\right)$.

As regards storage efficiency, the size of the mpk consists in n elements, while the sizes of $\mathsf{CT}_{\mathbf{x}}$ and sk_f are $n+1$ group elements, respectively 1 element. As can be observed in Section 3.1, the original proposal outperforms our candidate construction in both memory and time efficiency. Remarkably, the framework developed in [ABDP15] depends *generically* on the IND-CPA of the underlying public-key encryption scheme (if the scheme posses certain structural and homomorphic properties). Thus, for the case of ElGamal, the s-IND-FE-CPA follows from **DDH**. We emphasize that the proof we give is based on a different, *unrelated* assumption: **DDHI**, which neither implies or is implied by **DDH**.

3.1 Construction

This section introduces a simple IPFE scheme. Concretely, the master secret-key consists of a square matrix $\mathbf{T}$ sampled uniformly at random from $\mathbb{Z}_p^{n\times n}$. The key-derivation procedure corresponding to vector $\mathbf{y}$ makes use of $\mathbf{T}$, in order to derive $\mathbf{y}^\top \cdot \mathbf{T}$. Encrypting $\mathbf{x}$ makes use of the encoded inverse of $\mathbf{T}$ as mpk and is not as simple as obtaining $\left[\mathbf{T}^{-1} \cdot \mathbf{x}\right]$, as this operation gives rise to a trivially deterministic scheme. We face this issue by introducing additional randomness through vector $\mathbf{t}$ while releasing $[\mathbf{tt}] \leftarrow \left[\mathbf{T}^{-1} \cdot \mathbf{t}\right]$. Decryption works by computing $\left[\mathbf{t}^\top \cdot \mathbf{y}\right]$ and then "subtracting" this quantity from $\left[\sum_i^k (\mathbf{x}_i + \mathbf{t}_i) \cdot \mathbf{y}_i\right]$. Finally, the inner-product of $\mathbf{x}^\top \cdot \mathbf{y}$ is encapsulated in the exponent. To recover it, we make use of the assumed bounded-norm property of our scheme, extracting the discrete-log (the same technique as in [ABDP15]).

$\underline{\mathbf{Gen}(1^\lambda)}$:	$\underline{\mathbf{Enc}(\mathrm{mpk}, \mathbf{x} \leftarrow (\mathbf{x}_1, \ldots, \mathbf{x}_n))}$:
$\mathbf{T} \leftarrow_\$ \mathbb{Z}_p^{n\times n}$	$\left[\mathbf{T}^{-1}\right] \leftarrow \mathrm{mpk}$
$\mathrm{msk} \leftarrow \mathbf{T}$	$\mathbf{t} \leftarrow_\$ \mathbb{Z}_p^n$
$\mathrm{mpk} \leftarrow \left[\mathbf{T}^{-1}\right]$	$[\mathbf{tt}] \leftarrow \left[\mathbf{T}^{-1} \cdot \mathbf{t}\right]$
return $(\mathrm{msk}, \mathrm{mpk})$	for $i \leftarrow 1, n$ do:
	$\quad [\mathsf{CT}_i] \leftarrow [\mathbf{x}_i + \mathbf{t}_i]$
	$[\mathsf{CT}_\mathbf{x}] \leftarrow \left([\mathsf{CT}_1], \ldots, [\mathsf{CT}_n], [\mathbf{tt}]\right)$
	return $\mathsf{CT}_\mathbf{x}$
$\underline{\mathbf{KDer}(\mathrm{msk}, \mathbf{y} \leftarrow (\mathbf{y}_1, \ldots, \mathbf{y}_n))}$:	$\underline{\mathbf{Dec}(\mathsf{CT}_\mathbf{x}, \mathrm{sk}_\mathbf{y})}$:
$\mathbf{T} \leftarrow \mathrm{msk}$	$\gamma \leftarrow \left[\mathbf{y}\mathbf{y}^\top \cdot \mathbf{tt}\right]$
$\mathbf{y}\mathbf{y}^\top \leftarrow \mathbf{y}^\top \cdot \mathbf{T}$	$s \leftarrow \prod_{i=1}^n \mathsf{CT}_i^{\mathbf{y}_i}$
$\mathrm{sk}_\mathbf{y} \leftarrow (\mathbf{y}\mathbf{y}^\top, \mathbf{y})$	$R \leftarrow s/\gamma$
	return $\mathbf{DLP}(R)$

Fig. 4. A simple, bounded-norm inner-product functional encryption scheme, with the security proof relying on **DDHI** assumption.

Lemma 1 (Correctness). *The* **IPFE** *construction in Figure 4 enjoys correctness according to Definition 1.*

Proof.

$$
\begin{aligned}
&\gamma \leftarrow \left[\mathbf{t}\mathbf{t}^\top \cdot \mathbf{y}\mathbf{y}\right] \iff \\
&\gamma \leftarrow \left[\mathbf{t}^\top \cdot \mathbf{y}\right] \ . \\
&s \leftarrow \prod_{i=1}^{n} \mathsf{CT}_i^{\mathbf{y}_i} \iff \\
&s \leftarrow \left[\sum_{i=1}^{k} (x_i \cdot y_i + t_i \cdot y_i)\right] \iff \\
&s \leftarrow \left[\mathbf{x}^\top \cdot \mathbf{y} + \mathbf{t}^\top \cdot \mathbf{y})\right] \ . \\
&R \leftarrow s/\gamma \iff \\
&R \leftarrow \left[\mathbf{x}^\top \cdot \mathbf{y}\right]
\end{aligned}
\tag{3}
$$

Given that the $\|\mathbf{x}^\top \cdot \mathbf{y}\|$ is bounded by B, one can store in a table the encodings of the values $1 \to B$. This allows to query the DLOG through lookups in the table during the decryption procedure.

3.2 Security

Theorem 1. *The bounded-norm* **IPFE** *construction introduced in Figure 4 enjoys* s-IND-FE-CPA*-security (Definition 1) under the* **DDHI** *(Definition 5). The advantage of any* PPT *adversary* $\mathcal{A}$ *against the* s-IND-FE-CPA *security experiment is bounded as follows:*

$$\mathbf{Adv}^{\text{s-IND-FE-CPA}}_{\mathcal{A}',\mathbf{FE}}(\lambda) \leq \mathbf{Adv}^{\mathbf{DDHI}}_{\mathcal{A}}(\lambda) \ .$$

Proof. We show via a proof by contradiction that **DDHI**-hard implies the scheme in *Figure 4* is s-IND-FE-CPA-secure. Let $\mathcal{A}$ be an adversary against the s-IND-FE-CPA-security of our construction. We construct a PPT algorithm $\mathcal{A}'$ that makes use of $\mathcal{A}$ as an oracle in order to break the **DDHI** assumption. The **DDHI** game with respect to a group $\mathbb{G}$ of prime order p commences by sampling uniformly at random $a \leftarrow_{\$} \mathbb{Z}_p$ and providing to $\mathcal{A}'$ the tuple $([1], [a], [z])$, where z is either sampled uniformly at random over $\mathbb{G}$ or $z \leftarrow a^{-1}$. In the **DDHI**-security experiment, $\mathcal{A}'$ will simulate the view of an adversary, while in the view of $\mathcal{A}$, $\mathcal{A}'$ will "implement" the s-IND-FE-CPA security experiment.

Key-Derivation Queries. The first observation one can make comes by restricting the s-IND-FE-CPA experiment to inner-product functionalities. Specifically, the IPFE instantiation of the s-IND-FE-CPA experiment enables key-queries corresponding to vectors $\mathbf{y}$ such that:

$$\forall \mathbf{y} \in \mathbb{Z}_p^n : \mathbf{y}^\top \cdot \mathbf{x}^L = \mathbf{y}^\top \cdot \mathbf{x}^R \tag{4}$$

which can be equivalently stated that $\mathbf{y} \perp \left(\mathbf{x}^L - \mathbf{x}^R\right)$. This suggest that the challenge tuple $(\mathbf{x}^L, \mathbf{x}^R)$ implicitly defines an $(n-1)$-dimensional subspace:

$$\mathcal{S} \coloneqq \left\{\mathbf{y} \in \mathbb{Z}_p^n : \mathbf{y} \perp (\mathbf{x}^L - \mathbf{x}^R)\right\} \tag{5}$$

Prohibiting key-derivations for vectors outside $\mathcal{S}$ is one of the milestones in our security proof. The core idea is to implement a linear map L, using an astute algebraic trick presented in [HJ16], and "embed" this linear map in the matrix $\mathbf{T}$. Concretely, given $\mathbf{y} \in \mathcal{S}$, we can define a matrix $\mathbf{L}^{-1}$ that will map vectors from $\mathcal{S}$ to $\mathcal{W}$, where $\mathcal{W}$ stands for the $(n-1)$-dimensional vector space spanned by the first $n-1$ vectors in the canonical basis over $\mathbb{Z}_p$ (or equivalently, the subspace containing all vectors over $\mathbb{Z}_p^n$ with the last component set to 0):

$$\mathcal{W} \coloneqq \left\{ \mathbf{v} \in \mathbb{Z}_p^n : v_n = 0 \right\} \tag{6}$$

Obtaining $\mathbf{L}^{-1}$ is done by getting the inverse of $\mathbf{L}$, where $\mathbf{L} : \mathcal{W} \to \mathcal{S}$. The way to implement the matrix $\mathbf{L}$ is as follows: (1) sample uniformly at random a basis $\mathbf{B}_{\mathcal{S}}$ for $\mathcal{S}$ and a random vector $\mathbf{u} \leftarrow_{\$} \mathbb{Z}_p^n$; (2) set $\mathbf{L} \leftarrow \begin{pmatrix} \mathbf{B}_{\mathcal{S}} \\ \mathbf{u} \end{pmatrix}$.

Finally, the matrix $\mathbf{T}$ can be determined by sampling uniformly at random a set of $n-1$ *row* vectors $\mathbf{r}^{(1)}, \ldots, \mathbf{r}^{(n-1)}$, and then setting $\mathbf{T} \leftarrow \mathbf{L}^{-1} \cdot \mathbf{T}'$ where:

$$\mathbf{T}' \leftarrow \begin{pmatrix} \mathbf{r}_1^{(1)} & \ldots & \mathbf{r}_{n-1}^{(1)} & \mathbf{r}_n^{(1)} \\ \vdots & & \vdots & \vdots \\ \mathbf{r}_1^{(n-1)} & \ldots & \mathbf{r}_{n-1}^{(n-1)} & \mathbf{r}_n^{(n-1)} \\ 0 & \ldots & 0 & a \end{pmatrix} \tag{7}$$

Note that the construction of $\mathbf{T}'$ implicitly sets to a the value in position (n, n).

Remark. If $\mathbf{y} \in \mathcal{S}$, then the **KDer** procedure introduced in Figure 4 halts (i.e. $\mathcal{A}'$ can simulate **KDer**).

The remark follows from the fact that if $\mathbf{y} \in \mathcal{S}$, then $\mathbf{y}^\top \cdot \mathbf{T} = \mathbf{y}^\top \cdot \left(\mathbf{L}^{-1} \cdot \mathbf{T}' \right)$ $= \left(\mathbf{y}^\top \cdot \mathbf{L}^{-1} \right) \cdot \mathbf{T}'$. From $\mathbf{y} \in \mathcal{S}$ and $\mathbf{L}^{-1} : \mathcal{S} \to \mathcal{W}$, it results that $\mathbf{y} \cdot \mathbf{L}^{-1} \in \mathcal{W}$ (thus it has the last component 0). When performing the multiplication with $\mathbf{T}'$, the last row in $\mathbf{T}'$ is ignored due to the 0 component occurring in $\mathbf{y} \cdot \mathbf{L}^{-1}$, and thus $\mathcal{A}'$ is agnostic of the value of a. On the other hand, if $\mathbf{y} \notin \mathcal{S}$, then $\mathbf{y} \cdot \mathbf{L}^{-1} \notin \mathcal{W}$, meaning that the last row in $\mathbf{T}$ is used, (impossibility of simulation without knowledge of a).

The Public-Key. Once the matrix $\mathbf{T}$ has been obtained, one can get the algebraic form of $\left[\mathbf{T}^{-1}\right]$, by making use of the challenge value $[z]$. Concretely, we set:

$$\left[\mathbf{T}^{-1}\right] \leftarrow \left[\mathbf{T}'^{-1} \cdot \mathbf{L}\right] \tag{8}$$

where

$$\left[\mathbf{T}'^{-1}\right] \leftarrow \begin{pmatrix} \left[\mathbf{s}_1^{(1)}\right] & \ldots & \left[\mathbf{s}_{n-1}^{(1)}\right] & \left[z \cdot \mathbf{s}_n^{(1)}\right] \\ \vdots & & \vdots & \vdots \\ \left[\mathbf{s}_1^{(n-1)}\right] & \ldots & \left[\mathbf{s}_{n-1}^{(n-1)}\right] & \left[z \cdot \mathbf{s}_n^{(n-1)}\right] \\ [0] & \ldots & [0] & [z] \end{pmatrix} \tag{9}$$

and

$$\mathbf{L} : \mathcal{W} \to \mathcal{S} \ .$$

Note the special form of $\mathbf{T}'^{(-1)}$. The vectors $\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(n-1)}$ are obtained from the inverse of $\mathbf{T}'$ and depend on $\mathbf{r}^{(1)}, \ldots, \mathbf{r}^{(n-1)}$. As a general remark, note that $\left[\mathbf{T}^{-1} \cdot \mathbf{L}\right]$ can be computed, since $\mathbf{L}$ is known "in plain".

The Challenge Ciphertext. $\mathcal{A}'$ embeds the challenge value, namely $[z] \leftarrow \left[a^{-1}\right]$ or $[z] \leftarrow [r]$ in the challenge ciphertext. $\mathcal{A}'$ samples uniformly at random $b \in \{L, R\}$ and $\mathcal{A}'$ encrypts $\mathbf{x}^{(b)}$ with respect to the mpk. Then, $\mathcal{A}'$ samples $\mathbf{t}$ from $\mathbb{Z}_p^n$ such that $\mathbf{t}^\top \cdot \mathbf{u} = 0$. Then, $[\mathbf{tt}] \leftarrow [\mathbf{T} \cdot \mathbf{t}]$ is also given to the adversary.

Below, we show that in both cases, the adversary is able decrypt with respect to a key $\mathsf{sk}_{\mathbf{y}}$, independently of the challenge value $[z]$, the ciphertext being information theoretically hidden.

– **Case 1**: when $[z] \leftarrow \left[a^{-1}\right]$, we argue the chosen message, say $\mathbf{x}^b \leftarrow_\$ \left\{\mathbf{x}^L, \mathbf{x}^R\right\}$ is information-theoretically hidden from the view of the adversary $\mathcal{A}$. For this logical branch of the proof, observe that symbolically, $\left[\mathbf{y}\mathbf{y}^\top \cdot \mathbf{tt}\right] = \left[\mathbf{y}^\top \cdot \mathbf{t}\right]$ since $\mathbf{yy} \leftarrow \mathbf{y}^\top \mathbf{L}^{-1} \cdot \mathbf{T}'$ and $[\mathbf{tt}] \leftarrow \left[\mathbf{T}'^{-1} \cdot \mathbf{L} \cdot \mathbf{t}\right]$, since for this case $\mathbf{L}^{-1} \cdot \mathbf{T}' \cdot \mathbf{T}'^{-1} \cdot \mathbf{L}$ is symbolically equivalent to the identity matrix. Since $\mathbf{t}$ is a randomly sampled vector, it follows that:

$$\begin{aligned} \Pr\left[\mathsf{CT}_{\mathbf{x}} \leftarrow_\$ \mathbf{Enc}(\mathrm{mpk}, \mathbf{x}^L) | \, [z] \leftarrow \left[a^{-1}\right]\right] = \\ \Pr\left[\mathsf{CT}_{\mathbf{x}} \leftarrow_\$ \mathbf{Enc}(\mathrm{mpk}, \mathbf{x}^R) | \, [z] \leftarrow \left[a^{-1}\right]\right] \end{aligned} \tag{10}$$

– **Case 2**: when $[z] \leftarrow [r]$, the argument becomes more convoluted, and we detail it in what follows. When $[z] \leftarrow [r]$, we break the matrix $\mathbf{T}^{-1} = \mathbf{T}'^{-1} \cdot \mathbf{L}$ as $\mathbf{T}^{-1} = \left(\mathbf{T}_a'^{-1} + \mathbf{T}_r'^{-1}\right) \cdot \mathbf{L}$. An explicit form of $\mathbf{T}'^{-1} = \mathbf{T}_a'^{-1} + \mathbf{T}_r'^{-1}$ is below:

$$\begin{aligned} \mathbf{T}'^{-1} = & \begin{pmatrix} \mathbf{s}_1^{(1)} & \ldots & \mathbf{s}_{n-1}^{(1)} & a^{-1} \cdot \mathbf{s}_n^{(1)} \\ \vdots & & \vdots & \vdots \\ \mathbf{s}_1^{(n-1)} & \ldots & \mathbf{s}_{n-1}^{(n-1)} & a^{-1} \cdot \mathbf{s}_n^{(n-1)} \\ 0 & \ldots & 0 & a^{-1} \end{pmatrix} + \\ & \begin{pmatrix} 0 \ldots 0 & (r - a^{-1}) \cdot \mathbf{s}_n^{(1)} \\ \vdots \quad \vdots & \vdots \\ 0 \ldots 0 & (r - a^{-1}) \cdot \mathbf{s}_n^{(n-1)} \\ 0 \ldots 0 & (r - a^{-1}) \end{pmatrix} . \end{aligned} \tag{11}$$

Encrypting $\mathbf{x}^b$ for this case is done as well by sampling a $\mathbf{t}$ such that $\mathbf{t}^\top \cdot \mathbf{u} = 0$. Then $\mathcal{A}'$ issues $\left[\mathbf{x}^b + \mathbf{t}\right]$ as part of the challenge ciphertext. The only difference is that in this case we must prove decryption is correct. $\mathcal{A}'$ releases $\left[\left(\mathbf{T}'^{-1} \cdot \mathbf{L}\right) \cdot \mathbf{t}\right]$, which can be written as:

$$\left[\mathbf{T}'^{-1} \cdot \mathbf{L} \cdot \mathbf{t}\right] = \left[\mathbf{T}_{a^{-1}}'^{-1} \cdot \mathbf{L} \cdot \mathbf{t} + \mathbf{T}_r'^{-1} \cdot \mathbf{L} \cdot \mathbf{t}\right] \tag{12}$$

Given the adversary $\mathcal{A}$ can query $\mathbf{y}^\top \cdot \mathbf{L}^{-1} \cdot \mathbf{T}'_a$, when it computes $\left[\mathbf{y}\mathbf{y}^\top \cdot \mathbf{t}\mathbf{t}\right]$, it obtains:

$$\left[\mathbf{y}\mathbf{y}^\top \cdot \mathbf{t}\mathbf{t}\right] = \left[\mathbf{y}^\top \cdot \mathbf{t} + \mathbf{y}^\top \cdot \mathbf{L}^{-1} \cdot \mathbf{T}'_a \cdot \mathbf{T}'^{-1}_r \cdot \mathbf{L} \cdot \mathbf{t}\right] \tag{13}$$

We develop on the second term of the sum, showing it is actually 0.

$$\begin{aligned}
\mathbf{y}^\top \cdot \mathbf{L}^{-1} \cdot \mathbf{T}'_a \cdot \mathbf{T}'^{-1}_r \cdot \mathbf{L} \cdot \mathbf{t} &= \mathbf{y}^\top \cdot \mathbf{L}^{-1} \cdot \begin{pmatrix} 0 \dots 0 & e \\ \vdots & \vdots \\ 0 \dots 0 & f \\ 0 \dots 0 & g \end{pmatrix} \cdot \mathbf{L} \cdot \mathbf{t} \\
&= (\mathbf{y}_1 \ \dots \ \mathbf{y}_{n-1} \ 0) \cdot \begin{pmatrix} 0 \dots 0 & e \\ \vdots & \vdots \\ 0 \dots 0 & f \\ 0 \dots 0 & g \end{pmatrix} \cdot \mathbf{L} \cdot \mathbf{t} \\
&= (0 \ \dots \ 0 \ \alpha) \cdot \mathbf{L} \cdot \mathbf{t} \\
&= \alpha \cdot \mathbf{u}^\top \cdot \mathbf{t}
\end{aligned} \tag{14}$$

Since $\mathbf{t}^\top \cdot \mathbf{u} = 0$, then the last part of the sum is 0. This proves that decryption works, and $\mathbf{x}^b$ is information theoretically hidden.

4 A Word on Our Assumption

The relations between decisional Diffie Hellman, the Square DDH and the Inverse DDH, have been studied in the work of [BDZ03]. Their work, does not show that **DDH** is equivalent to **DDHI**.

However, the paper shows a) **DDHI** $\iff$ SquareDDH, and b) SquareDDH $\implies$ **DDH**. By transitivity, it follows that **DDHI** is a stronger assumption than **DDH**.

We provide a second look over the first result that is described in Section 3.2 of [BDZ03], namely **DDHI** $\implies$ SquareDDH. In their reduction "**DDHI** $\implies$ SquareDDH", the authors assume a SquareDDH adversary. The reduction receives as input

$$(g, g^x, g^r) \ ,$$

where r is either random or $1/x$.

Then, in order to simulate a SquareDDH tuple, one should set it as $(g^{r*s}, g^s, g^{x*s^2})$. However, the simulation does not hold as claimed, because if $r \leftarrow 1/x$, then the result becomes

$$((g^{s/x}), (g^{s/x})^x, (g^{s/x})^{s*x^2})$$

which is different from the expected

$$((g^{s/x}), (g^{s/x})^x, (g^{s/x})^{x^2})$$

Thus, the claim that **DDHI** $\implies$ **DDH** may not be true over the algebraic structure we consider herein.

References

ABDP15. Michel Abdalla, Florian Bourse, Angelo De Caro, and David Pointcheval. Simple functional encryption schemes for inner products. In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 733–751. Springer, Heidelberg, March / April 2015.

AFH+16. Martin R. Albrecht, Pooya Farshim, Dennis Hofheinz, Enrique Larraia, and Kenneth G. Paterson. Multilinear maps from obfuscation. In Kushilevitz and Malkin [KM16], pages 446–473.

AJ15. Prabhanjan Ananth and Abhishek Jain. Indistinguishability obfuscation from compact functional encryption. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 308–326. Springer, Heidelberg, August 2015.

BBL17. Fabrice Benhamouda, Florian Bourse, and Helger Lipmaa. CCA-secure inner-product functional encryption from projective hash functions. In Serge Fehr, editor, *PKC 2017, Part II*, volume 10175 of *LNCS*, pages 36–66. Springer, Heidelberg, March 2017.

BCFG17. Carmen Elisabetta Zaira Baltico, Dario Catalano, Dario Fiore, and Romain Gay. Practical functional encryption for quadratic functions with applications to predicate encryption. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 67–98. Springer, Heidelberg, August 2017.

BDZ03. Feng Bao, Robert H. Deng, and Huafei Zhu. Variations of Diffie-Hellman problem. In Sihan Qing, Dieter Gollmann, and Jianying Zhou, editors, *ICICS 03*, volume 2836 of *LNCS*, pages 301–312. Springer, Heidelberg, October 2003.

BGG+14. Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 533–556. Springer, Heidelberg, May 2014.

BGI14. Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional signatures and pseudorandom functions. In Hugo Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 501–519. Springer, Heidelberg, March 2014.

BJK15. Allison Bishop, Abhishek Jain, and Lucas Kowalczyk. Function-hiding inner product encryption. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 470–491. Springer, Heidelberg, November / December 2015.

BKS16. Zvika Brakerski, Ilan Komargodski, and Gil Segev. Multi-input functional encryption in the private-key setting: Stronger security from weaker assumptions. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 852–880. Springer, Heidelberg, May 2016.

BSW11. Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In Yuval Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 253–273. Springer, Heidelberg, March 2011.

BV15. Nir Bitansky and Vinod Vaikuntanathan. Indistinguishability obfuscation from functional encryption. In Venkatesan Guruswami, editor, *56th FOCS*, pages 171–190. IEEE Computer Society Press, October 2015.

CG13. Ran Canetti and Juan A. Garay, editors. *CRYPTO 2013, Part II*, volume 8043 of *LNCS*. Springer, Heidelberg, August 2013.

DDM16. Pratish Datta, Ratna Dutta, and Sourav Mukhopadhyay. Functional encryption for inner product with full function privacy. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *PKC 2016, Part I*, volume 9614 of *LNCS*, pages 164–195. Springer, Heidelberg, March 2016.

EHK+13. Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. An algebraic framework for Diffie-Hellman assumptions. In Canetti and Garay [CG13], pages 129–147.

GGH+13. Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, October 2013.

GKP+13. Shafi Goldwasser, Yael Tauman Kalai, Raluca A. Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. How to run turing machines on encrypted data. In Canetti and Garay [CG13], pages 536–553.

HJ16. Dennis Hofheinz and Tibor Jager. Verifiable random functions from standard assumptions. In Kushilevitz and Malkin [KM16], pages 336–362.

JLS20. Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. Cryptology ePrint Archive, Report 2020/1003, 2020. https://eprint.iacr.org/2020/1003.

KM16. Eyal Kushilevitz and Tal Malkin, editors. *TCC 2016-A, Part I*, volume 9562 of *LNCS*. Springer, Heidelberg, January 2016.

KS17. Ilan Komargodski and Gil Segev. From minicrypt to obfustopia via private-key functional encryption. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 122–151. Springer, Heidelberg, April / May 2017.

Lin16. Huijia Lin. Indistinguishability obfuscation from constant-degree graded encoding schemes. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 28–57. Springer, Heidelberg, May 2016.

LPST16. Huijia Lin, Rafael Pass, Karn Seth, and Sidharth Telang. Indistinguishability obfuscation with non-trivial efficiency. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *PKC 2016, Part II*, volume 9615 of *LNCS*, pages 447–462. Springer, Heidelberg, March 2016.

O'N10. Adam O'Neill. Definitional issues in functional encryption. 2010.

Tom. Junichi Tomida. Tightly secure inner product functional encryption: Multi-input and function-hiding constructions. pages 459–488.