A New Paradigm for Public-Key Functional Encryption for Degree-2 Polynomials

Romain Gay

Cornell Tech, NY
romain.gay@cornell.edu

Abstract. We give the first public-key functional encryption that supports the generation of functional decryption keys for degree-2 polynomials, with succinct ciphertexts, whose semi-adaptive simulation-based security is proven under standard assumptions. At the heart of our new paradigm lies a so-called partially function-hiding functional encryption scheme for inner products, which admits public-key instances, and that is sufficient to build functional encryption for degree-2 polynomials. Doing so, we improve upon prior works, such as the constructions from Lin (CRYPTO 17) or Ananth Sahai (EUROCRYPT 17), both of which rely on function-hiding inner product FE, that can only exist in the private-key setting. The simplicity of our construction yields the most efficient FE for quadratic functions from standard assumptions (even those satisfying a weaker security notion). The interest of our methodology is that the FE for quadratic functions that builds upon any partially function-hiding FE for inner products inherits the security properties of the latter. In particular, we build a partially function-hiding FE for inner products that enjoys simulation security, in the semi-adaptive setting, where the challenge sent from the adversary can be chosen adaptively after seeing the public key (but before corrupting functional decryption keys). This is in contrast from prior public-key FE for quadratic functions from Baltico et al. (CRYPTO 17), which only achieved an indistinguishability-based, selective security. As a bonus, we show that we can obtain security against Chosen-Ciphertext Attacks straightforwardly. Even though this is the de facto security notion for encryption, this was not achieved by prior functional encryption schemes for quadratic functions, where the generic Fujisaki Otamoto transformation (CRYPTO 99) does not apply.

1 Introduction

Functional Encryption [O'N10,BSW11](in short: FE) is a general paradigm where restricted decryption keys are generated, that let users learn specific functions of the encrypted data. Namely, each decryption key sk_f is associated with a function f, and the decryption of an encrypted message x with sk_f recovers $f(x)$, and nothing else. The scheme must be resistant to any collusion of decryption keys sk_f for different functions f: such group of keys should not learn anything more than the information leaked by each key sk_f, individually. This security property makes FE schemes both hard to build and extremely useful, provided the class of function they handle is large. In fact, it has been shown [BV15,AJ15] that general purpose functional encryption gives a construction of Indinstiguishability Obfuscation [BGI+01,GGG+14](in short: iO) for all circuits, a powerful object that has been remarkably successful at providing an all-purpose tool for solving cryptographic problems [SW14]. Surprisingly, even FE for smaller classes of functions are powerful. Recently, [LT17] has shown that succinct FE supporting degree-3 functions is sufficient to build iO, together with additional assumptions on the existence of special kind of pseudo-random generators [1]. However, there is no construction of such FE schemes from standard, well understood assumptions. All known constructions rely on either multilinear maps, or iO itself. Can we build FE for rich classes of functions from standard assumptions?

Beyond the case of predicate encryption [BW07,KSW08,GVW15], little is known about standard-based FE constructions. [ABDP15] gave the first construction of FE for inner products, where the encryption of a vector $\mathbf{x} \in \mathbb{Z}^n$, together with a decryption key associated with vector $\mathbf{y} \in \mathbb{Z}^n$, yields the inner product of $\mathbf{x}$

[1] Namely, the existence of pseudo-random generators of block-wise locality 3.

and $\mathbf{y}$. That is, their scheme can generate decryption keys that compute a weighted sum on encrypted data. They prove selective security, a useful but artificial security notion where the adversary has to commit to its challenge ciphertext beforehand. Later, [ALS16] gave constructions with full security (aka adaptive security, where the adversary can request decryption keys and the challenge ciphertext adaptively). Both constructions use standard assumptions (DDH, LWE, DCR). Note that inner products already capture constant depth circuits, by simply expressing circuits as polynomials, and encrypting all the monomials (of constant degree). However, for most applications, and in particular to obtain iO, one needs to recursively apply the FE scheme to itself. This bootstrapping requires the ciphertexts to be succinct, that is, their size should only depend on the underlying message, and not on the function to be evaluated. Following this quest for succinct FE for richer classes of functions, [BCFG17] (concurrently [AS17,Lin17] in the private-key setting), gave the first construction of succinct FE that supports the evaluation of quadratic functions on ciphertexts. All of these constructions are proven secure under an indistinguishability-based security definition, which is cumbersome to use, and is too weak to meaningful security for some functionality. Moreover, all these schemes either achieve only selective security, or assume the generic group model.

Our contributions. We build the first simulation-secure FE in the semi-adaptive setting, whose security relies on a standard assumption, that supports a functionality beyond inner products, or predicate encryption. In our scheme, ciphertexts are associated with two vectors $\mathbf{x} \in \mathbb{Z}^n$ and $\mathbf{y} \in \mathbb{Z}^m$, and decryption keys are associated with a matrix $\mathbf{F} \in \mathbb{Z}^{n \times m}$. The decryption of a ciphertext $\mathsf{ct}_{\mathbf{x},\mathbf{y}}$ with a decryption key $\mathsf{sk}_{\mathbf{F}}$ recovers $\mathbf{x}^\top \mathbf{F} \mathbf{y} \in \mathbb{Z}$. The ciphertext size is $O(n+m)$ group elements, and security relies on pairings (DLIN).

Scheme:	public-key	security	assumption
[AS17]	✗	SEL-IND	GGM
[Lin17]	✗	SEL-IND	SXDH
[BCFG17]	✓	SEL-IND	SXDH & 3-PDDH
[BCFG17,DGP18]	✓	AD-IND	GGM
ours	✓	SAD-SIM	DLIN

Fig. 1. Quadratic FE. Here, {AD,SAD,SEL}-{IND,SIM} stands for {adaptive,semi-adaptive,selective}-{indistinguishability,simulation} security. GGM stands for Generic Group Model.

To build our quadratic FE, we deploy a new paradigm that uses at its core a so-called partially function-hiding inner-product FE, where decryption keys partially hide their underlying function (in the case of inner product, their underlying vector). This approach allows us to obtain public-key FE, as opposed to prior work [AS17,Lin17] relying on full-fledged function-hiding inner-product FE, which is inherently private-key.

We then build a partially function-hiding inner-product FE with simulation security. This security notion implies its indistinguishability-based counterpart, and drastically simplifies the proof compared to previous works relying on indistinguishability-secure inner-product FE (for instance [Lin17]). This simplicity is illustrated by short ciphertexts and keys (see Fig.2). We obtain simulation security in the semi-adaptive setting, where an adversary is restricted to choose its challenge before querying any secret keys. This is the best we can hope for: a simple extension of [BSW11,AGVW13] shows that adaptively simulation secure partially function-hiding inner-product FE are impossible to achieve from standard assumptions (note this impossibility result doesn't apply to schemes proved in the generic group model, such as the inner-product FE from [KLM+18]). As shown in [BSW11], indistiguishability-based security is inadequate for some functionality. For instance, if a ciphertext encrypts the seed of a PRG, and each functional decryption key is associated with one position of the output of the PRG, simulation-based security ensure that only the output of the PRG is revealed, whereas indistinguishability-based security is essentially useless, since it only proves that an encryp-

tion of a seed is computationally indistinguishable from an encryption of seed' if PRG(seed)=PRG(seed'). This example is relevant in the context of quadratic FE, since our construction is expressive enough to evaluate the output of a PRG (see Remark 1). This indicates that simulation security is qualitatively stronger than its indistinguisahbility-based counterpart.

Scheme:	$\|\mathsf{ct}\|$	$\|\mathsf{sk}\|$	sec., assump.
[BCFG17]	$2n\|\mathbb{G}_1\| + (2m+2)\|\mathbb{G}_2\|$	$2\|\mathbb{G}_1\| + \|\mathbb{G}_2\|$	AD-IND, GGM
[BCFG17]	$(6n+1)\|\mathbb{G}_1\| + (6m+1)\|\mathbb{G}_2\|$	$\|\mathbb{G}_1\| + [\mathbb{G}_2\|$	SEL-IND, SDXH & 3-PDDH
[DGP18]	$(2n+1)\|\mathbb{G}_1\| + 2m\|\mathbb{G}_2\|$	$\mathbb{G}_2$	AD-IND, GGM
ours	$(4n+2m+2)\|\mathbb{G}_1\| + m\|\mathbb{G}_2\|$	$(3n+2m+2)\|\mathbb{G}_2\|$	SAD-SIM, DLIN

Fig. 2. Efficiency comparison between public-key quadratic FE, where ciphertext encrypt $(\mathbf{x}, \mathbf{y}) \in \mathbb{Z}^n \times \mathbb{Z}^m$ and decryption keys are associated with $\mathbf{F} \in \mathbb{Z}^{n \times m}$. {AD,SAD,SEL}-{IND,SIM} stands for {adaptive,semi-adaptive,selective}-{indistinguishability,simulation} security. GGM stands for Generic Group Model. SXDH stands for Symmetric eXternal Diffie Hellman, 3-PDDH stands for 3-Party Decisional Diffie Hellman, DLIN stands for Decisional LINear, both of which are standards assumptions in pairing groups.

Another benefit of our new approach is that many properties of the underlying partially function-hiding inner-product FE can be lifted to the overall quadratic FE. This is case of the semi-adaptive simulation-based security, but we also show that if the partially function-hiding inner-product FE is secure against Chosen-Ciphertext Attacks (CCA-security), then so is the resulting quadratic FE. CCA-security is the de facto security notion for encryption, as it captures active or man-in-the-middle attacks, as opposed to CPA security. However, previous quadratic FE only prove CPA security. Note that generic transformation, such as Fujisaki Okamoto transform [FO99], cannot be applied here, since it relies on hybrid encryption, which is incompatible with functional encryption, which permits selective computation on encrypted data, as opposed to the all-or-nothing access provided by typical encryption. The CHK transform [CHK04] has been extended in [GPSW06] to obtain CCA-security for Attribute-Based Encryption (ABE) with some delegatability property. This property has been relaxed in [YAHK11]. However, these techniques only apply to ABE, where a decryption secret key recovers the encrypted plaintext entirely, or not at all, which is different in nature from the Functional Encryption we are studying here, where only partial information about the plaintext is recovered. The only generic transformation that seems to apply in our case is the dual encryption methodology from [NY90], which has the disadvantage of doubling the size of ciphertexts, and relying on (simulation-sound) non-interactive zero knowledge proofs. [BBL17] avoids using the Naor Yung paradigm, and builds the first CCA-secure FE (beyond the case of ABE), which handles inner product, and is based on efficient hash-proof systems. Their security proof crucially relies on structural (linearly homomorphic) properties of hash proofs system, which is tailored to FE for inner products. Indeed, none of these techniques seem to be applicable to existing quadratic FE, such as [BCFG17]. Our construction strikes by its simplicity: it suffices to build a CCA-secure partially function-hiding inner-product FE, which can be simply obtained by adding an Quasi-Adaptive Non-Interactive Zero Knowledge argument for the simple language of DDH tuples, without doubling the size of the ciphertext as required by Naor Yung dual encryption methodology. Instantiating these with arguments from [KW15] only adds 2 group elements in the ciphertexts, and requires no extra assumption. Surely, this is made easy by the use of pairings, which are not used by [BBL17]. In fact, we do not consider CCA-security to be the main technical contribution of this paper, but rather an illustration of the interest of building quadratic FE from inner-product FE, as is done in our new paradigm.

Technical overview.

Quadratic FE. Our quadratic FE uses a pairing group $\mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, where the encryption of $\mathbf{x}, \mathbf{y}$ contains an encryption $\mathsf{Enc}_1(\mathbf{x}; r)$ of $\mathbf{x}$ under randomness r, that consists of elements in $\mathbb{G}_1$, and an encryption $\mathsf{Enc}_2(\mathbf{y}; s)$ of $\mathbf{y}$ under randomness s, which consists of elements in $\mathbb{G}_2$. Thanks to the pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, we can compute the product of $\mathsf{Enc}_1(\mathbf{x}, r)$ and $\mathsf{Enc}_2(\mathbf{y}, s)$ to obtain the output $\mathbf{x}^\top \mathbf{F} \mathbf{y}$ in the group $\mathbb{G}_T$, masked by some extra terms, that can be expressed as the inner product of a vector that only depends on the input $\mathbf{x}, \mathbf{y}$, and the randomness r, s used by the encryption, together with another vector which only depends on the secret key of these encryptions, and the matrix $\mathbf{F}$. Both vectors have a dimension that is *linear* in the dimension of the vectors $\mathbf{x}$ and $\mathbf{y}$. Thus, as in [AS17,Lin17], we can use an inner-product FE to compute the masking term. Such inner-product FE needs to be function-hiding, since revealing the secret key would compromise the security of the encryptions Enc_1 and Enc_2. However, function-hiding FE is an inherently private-key primitive, since a public encryption would allow to recover the function underlying each decryption key, simply by encrypting well-chosen vectors and decrypting them using the decryption key. To obtain a public key quadratic FE, we make the crucial observation that the underlying function-hiding FE for inner products is only used for vectors that lie in some specific subspace. Thus, we create, and make public, a restricted encryption key that can only generate ciphertexts for these vectors, while still providing some meaningful function-hiding. In particular, we obtain a public-key inner-product FE where decryption keys *partially* hide their underlying vector, which turns out to be sufficient for the security proof of the quadratic FE. Roughly speaking, the security of the inner-product FE proves that only the masking terms are revealed, along with some partial information on the secret keys that do not compromise security of the encryptions Enc_1, Enc_2. Thus, we obtain security of the quadratic FE using the security of the latter encryptions.

Partially function-hiding inner-product FE. We now highlight the construction of our new public-key, partially function-hiding inner-product FE. Our starting point is the FE for inner products from [ALS16], where decrypting an encryption $\mathsf{Enc}(\mathbf{x})$ of a vector $\mathbf{x}$ with a decryption key $\mathsf{KeyGen}(\mathbf{y})$ associated with vector $\mathbf{y}$ yields the inner product $\mathbf{x}^\top \mathbf{y}$. Their scheme is not function-hiding since $\mathbf{y}$ is part of the decryption key generated by $\mathsf{KeyGen}(\mathbf{y})$. As in [Lin17, Section 6.3], we use the fact that the decryption computes the inner product of $\mathsf{Enc}(\mathbf{x})$ and $\mathsf{KeyGen}(\mathbf{y})$ to obtain $\mathbf{x}^\top \mathbf{y}$. Namely, we replace the vector $\mathbf{y}$ in each decryption key by an ALS encryption of $\mathbf{y}$, and $\mathbf{x}$ in each ciphertext is replaced by an ALS decryption key for $\mathbf{x}$ (see Fig.3). Function-Hiding (hiding $\mathbf{y}$) follows from the security of the inner ALS FE, whereas security (hiding $\mathbf{x}$) follows from the security of the outter ALS FE. [Lin17] uses a similar approach, where the entire decryption key is encrypted using an outter inner-product FE (see Fig.3), and the underlying inner-product FE [ABDP15] are only selective indistinguishability secure.

ours	[Lin17]
$\mathsf{ct}_\mathbf{x} = \mathsf{Enc}^{\mathsf{out}}\left(\mathsf{KeyGen}^{\mathsf{in}}(\mathbf{x})\right)$	$\mathsf{ct}_\mathbf{x} = \mathsf{KeyGen}^{\mathsf{out}}\left(\mathsf{Enc}^{\mathsf{in}}(\mathbf{x})\right)$
$\mathsf{sk}_\mathbf{y} = \mathsf{KeyGen}^{\mathsf{out}}\left(\mathsf{Enc}^{\mathsf{in}}(\mathbf{y})\right)$	$\mathsf{sk}_\mathbf{y} = \mathsf{Enc}^{\mathsf{out}}\left(\mathsf{KeyGen}^{\mathsf{in}}(\mathbf{y})\right)$

Fig. 3. Function-Hiding FE for inner products. In the leftmost column (resp. rightmost column) $(\mathsf{Enc}^{\mathsf{out}}, \mathsf{KeyGen}^{\mathsf{out}})$ and $(\mathsf{Enc}^{\mathsf{in}}, \mathsf{KeyGen}^{\mathsf{in}})$ are two independent instances of [ALS16] (resp. [ABDP15]) FE for inner products.

To make our scheme public-key, we publish a restricted secret key for the inner layer FE that lets $\mathsf{KeyGen}^{\mathsf{in}}(\mathbf{x})$ run on vectors $\mathbf{x}$ that lie in some subspace. To be of use in our quadratic FE scheme, our function-hiding FE needs to be simulation secure (this is stronger than the classical indistinguishability based security for FE). We prove simulation security using the simulation security of [ALS16], which was proved in [AGRW17,Wee17] in the selective setting.

CCA-security. As a bonus, we show that we can easily obtain CCA-security for our partially function-hiding inner-product FE, and that security property is transferred to the overall quadratic FE. We use Quasi-Adaptive Non-Interactive arguments for the simple language of DDH tuples, which must fulfill one-time simulation-soundness, in order to boost the security of our partially function-hiding FE for inner products to handle Chosen Ciphertext Attacks. This QANIZK argument can be instantiated with [KW15, Section 3.3], which only adds two group elements in the ciphertexts (this is the case $k = 1$ in their paper) and rely on the Kernel assumption, implied by SXDH (this is competitive with Fiat Shamir NIZKs, and does not rely on the random oracle model). Recall that prior constructions fail to obtain CCA-security even in the random oracle model, since the Fujisaki Okamoto transform, which relies on hybrid encryption (that is incompatible with functional encryption, where only a partial information on the plaintext is recovered during decryption), is of no help here.

Conclusion, and perspective. Summarizing, we exhibit a new paradigm to build quadratic FE from partially function-hiding FE for inner products, a newly introduced primitive that bypasses impossibility results of public-key function-hiding FE. This gives stronger, desirable security guarantees that were previously not achieved. Moreover, its simplicity is appealing, not only because it gives constructions that outperform previous standard-based schemes in terms of ciphertext size, but also because it transfers properties from inner-product FE to quadratic FE. An important exception is adaptive security. Even though there are adaptively-secure inner-product FE (in fact we claim, without proof, that our semi-adaptive partially function-hiding FE for inner products can be extended to the adaptive, indistinguishability-based setting, up to doubling the size of the ciphertexts, as done in [LV16]), our quadratic FE fails at achieving adaptive security. Despite this shortcomings, we are optimistic this new approach will shed light on the largely unexplored domain of building functional encryption for richer functionalities from standard assumptions.

Road-map. The rest of this paper is organized as follows. After giving some relevant technical preliminaries in Section 2, we define partially function-hiding public-key FE for inner products, and generically use it to build a quadratic FE, in Section 3. Then, in Section 4, we give concrete instances of such partially function-hiding FE for inner products, using standard assumptions on pairing groups.

2 Preliminaries

2.1 Notations

For any set S, we denote by $a \leftarrow_{\mathrm{R}} S$ a uniformly random element a in S. PPT stands for Probabilistic Polynomial Time. For any PPT algorithm $\mathcal{A}$, we denote by $x \leftarrow \mathcal{A}$ a random output from $\mathcal{A}$. We use $\approx_c$ to denote computational indistinguishability, and $\equiv$ to denote equality between distributions.

2.2 Pairing groups.

Let PGGen be a PPT algorithm that on input the security parameter 1^λ, returns a description $\mathcal{PG} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, P_1, P_2, e)$ where for all $s \in \{1, 2, T\}$, $\mathbb{G}_s$ is an additive cyclic group of order p for a 2λ-bit prime p. $\mathbb{G}_1$ and $\mathbb{G}_2$ are generated by P_1 and P_2 respectively, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is an efficiently computable (non-degenerate) bilinear map. Define $P_T := e(P_1, P_2)$, which is a generator of $\mathbb{G}_T$, of order p. We use implicit representation of group elements. For $s \in \{1, 2, T\}$ and $a \in \mathbb{Z}_p$, define $[a]_s = a \cdot P_s \in \mathbb{G}_s$ as the implicit representation of a in G_s. More generally, for a matrix $\mathbf{A} = (a_{ij}) \in \mathbb{Z}_p^{n \times m}$ we define $[\mathbf{A}]_s$ as the implicit representation of $\mathbf{A}$ in $\mathbb{G}_s$:

$$[\mathbf{A}]_s := \begin{pmatrix} a_{11} \cdot P_s & \dots & a_{1m} \cdot P_s \\ & & \\ a_{n1} \cdot P_s & \dots & a_{nm} \cdot P_s \end{pmatrix} \in \mathbb{G}_s^{n \times m}.$$

Given $[a]_1$ and $[b]_2$, one can efficiently compute $[a \cdot b]_T$ using the pairing e. For matrices $\mathbf{A}$ and $\mathbf{B}$ of matching dimensions, define $e([\mathbf{A}]_1, [\mathbf{B}]_2) := [\mathbf{AB}]_T$. For any matrix $\mathbf{A}, \mathbf{B} \in \mathbb{Z}_p^{n \times m}$, any group $s \in \{1, 2, T\}$, we denote by $[\mathbf{A}]_s + [\mathbf{B}]_s = [\mathbf{A} + \mathbf{B}]_s$.

For any prime p, we define the following distributions. The DDH distribution over $\mathbb{Z}_p^2$: $a \leftarrow_{\mathrm{R}} \mathbb{Z}_p$, outputs $\mathbf{a} := \binom{1}{a}$. The DLIN distribution over $\mathbb{Z}_p^{3\times 2}$: $a, b \leftarrow_{\mathrm{R}} \mathbb{Z}_p$, outputs $\mathbf{A} := \begin{pmatrix} a & 0 \\ 0 & b \\ 1 & 1 \end{pmatrix}$.

Definition 1 (DDH assumption). *For any adversary $\mathcal{A}$, any group $s \in \{1, 2, T\}$ and any security parameter λ, let*

$$\mathrm{Adv}^{\mathsf{DDH}}_{\mathbb{G}_s,\mathcal{A}}(\lambda) := |\Pr[1 \leftarrow \mathcal{A}(\mathcal{PG}, [\mathbf{a}]_s, [\mathbf{a}r]_s)] - \Pr[1 \leftarrow \mathcal{A}(\mathcal{PG}, [\mathbf{a}]_s, [\mathbf{u}]_s)]|,$$

where the probabilities are taken over $\mathcal{PG} \leftarrow_R \mathsf{GGen}(1^\lambda, d)$, $\boldsymbol{a} \leftarrow_R \mathsf{DDH}$, $r \leftarrow_R \mathbb{Z}_p$, $\mathbf{u} \leftarrow_R \mathbb{Z}_p^2$, and the random coins of $\mathcal{A}$. We say DDH holds in $\mathbb{G}_s$ if for all PPT adversaries $\mathcal{A}$, $\mathrm{Adv}^{\mathsf{DDH}}_{\mathbb{G}_s,\mathcal{A}}(\lambda)$ is a negligible function of λ.

Definition 2 (SXDH assumption). *For any security parameter λ and any pairing group $\mathcal{PG} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, P_1, P_2, e) \leftarrow_R \mathsf{PGGen}(1^\lambda)$, we say SXDH holds in $\mathcal{PG}$ if DDH holds in $\mathbb{G}_1$ and $\mathbb{G}_2$.*

Definition 3 (bilateral DLIN). *For any adversary $\mathcal{A}$, any security parameter λ, let*

$$\mathrm{Adv}^{\mathsf{DLIN}}_{\mathcal{A}}(\lambda) := |\Pr[1 \leftarrow \mathcal{A}\left(\mathcal{PG}, \{[\mathbf{A}]_s, [\mathbf{Ar}]_s, \}_{s\in\{1,2\}}\right) - \Pr[1 \leftarrow \mathcal{A}\left(\mathcal{PG}, \{[\mathbf{A}]_s, [\mathbf{u}]_s\}_{s\in\{1,2\}}\right)]|,$$

with where the probabilities are taken over $\mathcal{PG} \leftarrow_R \mathsf{GGen}(1^\lambda, d)$, $\mathbf{A} \leftarrow_R \mathsf{DLIN}$, $\mathbf{r} \leftarrow_R \mathbb{Z}_p^2$, $\mathbf{u} \leftarrow_R \mathbb{Z}_p^3$, and the random coins of $\mathcal{A}$. We say bilateral DLIN holds relative to $\mathcal{PG}$ if for all PPT adversaries $\mathcal{A}$, $\mathrm{Adv}^{\mathsf{DLIN}}_{\mathcal{A}}(\lambda)$ is a negligible function of λ.

2.3 Functional Encryption.

A functional encryption scheme for a functionality $\mathcal{F} : \mathcal{X} \to \mathcal{Z}$ is a tuple of PPT algorithms:

- $\mathsf{Setup}(1^\lambda, \mathcal{F})$: on input the security paramter λ, the functionality $\mathcal{F}$, returns a public key pk (which is implicitly an input of all other algorithms), and a master secret key msk.
- $\mathsf{Enc}(x \in \mathcal{X})$: returns ct_x, an encryption of x.
- $\mathsf{KeyGen}(\mathsf{msk}, f \in \mathcal{F})$: returns sk_f, a decryption key for f.
- $\mathsf{Dec}(\mathsf{ct}_x, \mathsf{sk}_f)$: deterministic algorithm that returns a value in $\mathcal{Z}$, or $\perp$ if it fails.

An FE scheme is said to be private-key if Enc requires msk as additional input, otherwise, it is public-key.

Correctness. For any security paramter λ, any functionality $\mathcal{F} : \mathcal{X} \to \mathcal{Z}$, any $x \in \mathcal{X}$, and $f \in \mathcal{F}$, $\Pr[\mathsf{Dec}(\mathsf{ct}_x, \mathsf{sk}_f) = f(x)] = 1$, where the probability is taken over $(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{F})$, $\mathsf{ct}_x \leftarrow \mathsf{Enc}(x)$, $\mathsf{sk}_f \leftarrow \mathsf{KeyGen}(\mathsf{msk}, f)$.

Security. We recall the notion of simulation security, which implies its indistinguishability counterpart. Both notions were originally introduced in [BSW11,O'N10]. We work in the semi-adaptive setting, where the adversary sends its challenge x before querying any secret keys, but after receiving the public key. Semi-adaptive security has been introduced in [CW14] in the context of Attribute-Based Encryption, and subsequently studied in [GKW16]. It implies traditional selective security (where the adversary sends x before seeing the public key and querying secret keys), and is implied by the full-fledged adaptive security (where the adversary can query secret keys before sending its challenge x). We give both Chosen-Plaintext Attack (CPA) and Chosen-Ciphertext Attack variants of simulation security.

Definition 4 (Simulation CPA security). *For any FE scheme* FE *for functionality* $\mathcal{F}$*, any security parameter* λ*, any PPT simulator* $\mathcal{S} := (\widetilde{\mathsf{Setup}}, \widetilde{\mathsf{Enc}}, \widetilde{\mathsf{KeyGen}})$*, and any PPT stateful adversary* $\mathcal{A}$*, we define the following two experiments.*

$\underline{\mathsf{Real}^{\mathsf{CPA\text{-}FE}}_{\mathcal{A}}(1^\lambda)}$:
$(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{F})$
$x^\star \leftarrow \mathcal{A}(1^\lambda, \mathsf{pk})$
$\mathsf{ct}^\star \leftarrow \mathsf{Enc}(x^\star)$
$\alpha \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KeyGen}}(\cdot)}(\mathsf{ct}^\star)$

$\underline{\mathsf{Ideal}^{\mathsf{CPA\text{-}FE}}_{\mathcal{A},\mathcal{S}}(1^\lambda)}$:
$(\widetilde{\mathsf{pk}}, \widetilde{\mathsf{msk}}) \leftarrow \widetilde{\mathsf{Setup}}(1^\lambda, \mathcal{F})$
$x^\star \leftarrow \mathcal{A}(1^\lambda, \widetilde{\mathsf{pk}})$
$\mathsf{ct}^\star \leftarrow \widetilde{\mathsf{Enc}}(\widetilde{\mathsf{msk}})$
$\alpha \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KeyGen}}(\cdot)}(\mathsf{ct}^\star)$

In the real experiment, the key generation oracle $\mathcal{O}_{\mathsf{KeyGen}}$*, when given as input* $f \in \mathcal{F}$*, returns* $\mathsf{KeyGen}(\mathsf{msk}, f)$*. In the ideal experiment, the key generation oracle* $\mathcal{O}_{\mathsf{KeyGen}}$*, when given as input* $f \in \mathcal{F}$*, computes* $f(x^\star)$*, and returns* $\widetilde{\mathsf{KeyGen}}(\widetilde{\mathsf{msk}}, f, f(x^\star))$*.*

We say an FE scheme is CPA-SIM secure if there exists a PPT simulator $\mathcal{S} := (\widetilde{\mathsf{Setup}}, \widetilde{\mathsf{Enc}}, \widetilde{\mathsf{KeyGen}})$ *such that for all PPT adversaries* $\mathcal{A}$*, we have:*

$$\mathrm{Adv}^{\mathsf{CPA\text{-}SIM}}_{\mathsf{FE},\mathcal{A}}(\lambda) := |\Pr[1 \leftarrow \mathsf{Real}^{\mathsf{CPA\text{-}FE}}_{\mathcal{A}}(1^\lambda)] - \Pr[1 \leftarrow \mathsf{Ideal}^{\mathsf{CPA\text{-}FE}}_{\mathcal{A},\mathcal{S}}(1^\lambda)]| = \mathsf{negl}(\lambda).$$

Definition 5 (Simulation CCA security). *For any FE scheme* FE *for functionality* $\mathcal{F}$*, any security parameter* λ*, any PPT simulator* $\mathcal{S} := (\widetilde{\mathsf{Setup}}, \widetilde{\mathsf{Enc}}, \widetilde{\mathsf{KeyGen}}, \widetilde{\mathsf{Dec}})$*, and any PPT stateful adversary* $\mathcal{A}$*, we define the following two experiments.*

$\underline{\mathsf{Real}^{\mathsf{CCA\text{-}FE}}_{\mathcal{A}}(1^\lambda)}$:
$(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{F})$
$x^\star \leftarrow \mathcal{A}(1^\lambda, \mathsf{pk})$
$\mathsf{ct}^\star \leftarrow \mathsf{Enc}(x)$
$\alpha \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KeyGen}}(\cdot), \mathcal{O}_{\mathsf{Dec}}(\cdot,\cdot)}(\mathsf{ct}^\star)$

$\underline{\mathsf{Ideal}^{\mathsf{CCA\text{-}FE}}_{\mathcal{A},\mathcal{S}}(1^\lambda)}$:
$(\widetilde{\mathsf{pk}}, \widetilde{\mathsf{msk}}) \leftarrow \widetilde{\mathsf{Setup}}(1^\lambda, \mathcal{F})$
$x^\star \leftarrow \mathcal{A}(1^\lambda, \widetilde{\mathsf{pk}})$
$\mathsf{ct}^\star \leftarrow \widetilde{\mathsf{Enc}}(\widetilde{\mathsf{msk}})$
$\alpha \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KeyGen}}(\cdot), \mathcal{O}_{\mathsf{Dec}}(\cdot,\cdot)}(\mathsf{ct}^\star)$

In the real experiment, the oracle $\mathcal{O}_{\mathsf{KeyGen}}$*, when given as input* $f \in \mathcal{F}$*, returns* $\mathsf{KeyGen}(\mathsf{msk}, f)$*; the oracle* $\mathcal{O}_{\mathsf{Dec}}$*, given as input a ciphertext* ct *different from the challenge ciphertext* $\mathsf{ct}^\star$ *and a function* $f \in \mathcal{F}$*, computes* $\mathsf{sk}_f \leftarrow \mathsf{KeyGen}(\mathsf{msk}, f)$*, and returns* $\mathsf{Dec}(\mathsf{ct}, \mathsf{sk}_f)$*. If* $\mathcal{O}_{\mathsf{Dec}}$ *is queried on an input that contains the challenge ciphertext* $\mathsf{ct}^\star$*, it returns* $\perp$*.*

In the ideal experiment, the oracle $\mathcal{O}_{\mathsf{KeyGen}}$*, when given as input* $f \in \mathcal{F}$*, computes* $f(x^\star)$*, and returns* $\widetilde{\mathsf{KeyGen}}(\widetilde{\mathsf{msk}}, f, f(x^\star))$*. The oracle* $\mathcal{O}_{\mathsf{Dec}}$*, when given as input a ciphertext* ct *different from the challenge ciphertext* $\mathsf{ct}^\star$ *and a function* $f \in \mathcal{F}$*, returns* $\widetilde{\mathsf{Dec}}(\widetilde{\mathsf{msk}}, f, \mathsf{ct})$*. If* $\mathcal{O}_{\mathsf{Dec}}$ *is queried on an input that contains the challenge ciphertext* $\mathsf{ct}^\star$*, it returns* $\perp$*.*

We say an FE scheme is CCA-SIM secure if there exists a PPT simulator $\mathcal{S} := (\widetilde{\mathsf{Setup}}, \widetilde{\mathsf{Enc}}, \widetilde{\mathsf{KeyGen}}, \widetilde{\mathsf{Dec}})$ *such that for all PPT adversaries* $\mathcal{A}$*, we have:*

$$\mathrm{Adv}^{\mathsf{CCA\text{-}SIM}}_{\mathsf{FE},\mathcal{A}}(\lambda) := |\Pr[1 \leftarrow \mathsf{Real}^{\mathsf{CCA\text{-}FE}}_{\mathcal{A}}(1^\lambda)] - \Pr[1 \leftarrow \mathsf{Ideal}^{\mathsf{CCA\text{-}FE}}_{\mathcal{A},\mathcal{S}}(1^\lambda)]| = \mathsf{negl}(\lambda).$$

2.4 Quasi-adaptive Non-Interactive Zero-Knowledge

This part is taken almost verbatim from [KW15]. Quasi-Adaptive NIZK (QA-NIZK) proofs are NIZK proofs where the common reference string (CRS) is allowed to depend on the specific language for which proofs have to be generated [JR13]. The CRS is generated in a specific way and contains a fixed part par, produced by an algorithm $\mathsf{Gen}_{\mathsf{par}}$, and a language-dependent part crs. However, for the zero-knowledge property there should exist a single simulator for the entire class of languages.

For public parameters par produced by $\mathsf{Gen}_{\mathsf{par}}$, let $\mathcal{D}_{\mathsf{par}}$ be a probability distribution over a collection of relations $R = \{R_\rho\}$ parametrized by a string ρ with an associated language $\mathcal{L}_\rho = \{y : \exists x \text{ s.t. } R_\rho(y, x) = 1\}$. We now recall the tag definition of QANIZK for $\mathcal{D}_{\mathsf{par}}$, in its tag-based variant.

Definition 6 (QANIZK Argument). *A* Quasi-adaptive Non-Interactive Zero Knowledge Argument *(QANIZK) Π for a language distribution $\mathcal{D}_{\mathsf{par}}$ consists of five PPT algorithms $\Pi = (\mathsf{Gen}_{\mathsf{par}}, \mathsf{Gen}_{\mathsf{crs}}, \mathsf{Prove}, \mathsf{Sim}, \mathsf{Ver})$:*

- $\mathsf{Gen}_{\mathsf{par}}(1^\lambda)$*: returns the public parameters* par*.*
- $\mathsf{Gen}_{\mathsf{crs}}(\mathsf{par}, \rho)$*: returns a common reference string* crs*, and a trapdoor* trap*. We assume that* crs *implicitly contains* par *and ρ, and that it defines a tag space $\mathcal{T}$.*
- $\mathsf{Prove}(\mathsf{crs}, \tau, x, y)$*: on input the* crs*, a tag $\tau \in \mathcal{T}$, a witness x and a statement y, it returns a proof π.*
- $\mathsf{Ver}(\mathsf{crs}, \tau, y, \pi)$*: on input* crs*, a tag $\tau \in \mathcal{T}$, a statement y, and a proof π, it returns 1 or 0, where 1 means that π is a valid proof of $y \in \mathcal{L}_\rho$, with respect to tag τ.*
- $\mathsf{Sim}(\mathsf{crs}, \mathsf{trap}, \tau, y)$*: returns a proof π for some $y \in \mathcal{Y}$ (not necessarily in $\mathcal{L}_\rho$).*

We require that the algorithms satisfy the following properties:

Perfect completeness. *For all λ, all* par *output by* $\mathsf{Gen}_{\mathsf{par}}(\lambda)$*, all ρ output by $\mathcal{D}_{\mathsf{par}}$, all (x, y) with $R_\rho(y, x) = 1$, all $\tau \in \mathcal{T}$, we have:*

$$\Pr[\mathsf{Ver}(\mathsf{crs}, \tau, y, \pi) = 1 | (\mathsf{crs}, \mathsf{trap}) \leftarrow_R \mathsf{Gen}_{\mathsf{crs}}(\mathsf{par}, \rho); \pi \leftarrow_R \mathsf{Prove}(\mathsf{crs}, \tau, x, y)] = 1.$$

$$\Pr\left[\mathsf{Ver}(\mathsf{crs}, \tau, y, \pi) = 1 \;\middle|\; \begin{array}{l}(\mathsf{crs}, \mathsf{trap}) \leftarrow_R \mathsf{Gen}_{\mathsf{crs}}(\mathsf{par}, \rho) \\ \pi \leftarrow_R \mathsf{Prove}(\mathsf{crs}, \tau, x, y)\end{array}\right] = 1.$$

Perfect zero-knowledge. *For all λ, all* par *output by* $\mathsf{Gen}_{\mathsf{par}}(\lambda)$*, all ρ output by $\mathcal{D}_{\mathsf{par}}$, all* $(\mathsf{crs}, \mathsf{trap})$ *output by* $\mathsf{Gen}_{\mathsf{crs}}(\mathsf{par}, \rho)$*, all (x, y) with $R_\rho(y, x) = 1$, all tags $\tau \in \mathcal{T}$, the distributions*

$$\mathsf{Prove}(\mathsf{crs}, \tau, x, y) \text{ and } \mathsf{Sim}(\mathsf{crs}, \mathsf{trap}, \tau, y)$$

are the same (where the coin tosses are taken over $\mathsf{Prove}, \mathsf{Sim}$*).*

Simulation Soundness. *For all PPT adversaries $\mathcal{A}$ and any QANIZK argument Π the following advantage*

$$\mathrm{Adv}^{\Pi}_{\mathcal{A}}(\lambda) := \Pr\left[\begin{array}{l|l}\mathsf{Ver}(\mathsf{crs}, \tau^\star, y^\star, \pi^\star) = 1 & \mathsf{par} \leftarrow_R \mathsf{Gen}_{\mathsf{par}}(\lambda), \rho \leftarrow_R \mathcal{D}_{\mathsf{par}} \\ \wedge y^\star \notin \mathcal{L}_\rho \wedge \tau^\star \notin \mathcal{T}_{\mathsf{sim}} & (\mathsf{crs}, \mathsf{trap}) \leftarrow_R \mathsf{Gen}_{\mathsf{crs}}(\mathsf{par}, \rho) \\ & (y^\star, \tau^\star, \pi^\star) \leftarrow \mathcal{A}^{\mathsf{SimO}(\cdot,\cdot)}(\mathsf{crs})\end{array}\right]$$

is negligible, where $\mathsf{SimO}(\tau, y)$ *returns $\pi := \mathsf{Sim}(\mathsf{crs}, \mathsf{trap}, \tau, y)$ and sets $\mathcal{T}_{\mathsf{sim}} := \mathcal{T}_{\mathsf{sim}} \cup \{\tau\}$, where $\mathcal{T}_{\mathsf{sim}}$ is initially empty.*

One-time Simulation Soundness. For any PPT adversary $\mathcal{A}$ and QANIZK argument Π, we define $\mathrm{Adv}^{\mathsf{OT}\text{-}\Pi}_{\mathcal{A}}(\lambda)$ as $\mathrm{Adv}^{\Pi}_{\mathcal{A}}(\lambda)$, except the adversary can only make one query *to the oracle* SimO*.*

3 Quadratic FE from Inner-Product FE

In this section we build a functional encryption scheme for bounded-norm quadratic functions, namely, for the functionality $\mathcal{F}_{\mathsf{quad},B} : [0, B]^n \times [0, B]^m \to [0, n \cdot m \cdot B^3]$, $\mathcal{X} := [0, B]^n \times [0, B]^m$, $\mathcal{Z} := [0, n \cdot m \cdot B^3]$, such that each $\mathbf{F} \in \mathcal{F}_{\mathsf{quad},B}$ is represented by a matrix in $[0, B]^{n \times m}$, and for all $(\mathbf{x}, \mathbf{y}) \in [0, B]^n \times [0, B]^m$, the output of the function is $\mathbf{x}^\top \mathbf{F} \mathbf{y} \in [0, n \cdot m \cdot B^3]$. We consider B, n, m all polynomials in the security parameter.

Our quadratic FE is built from a so-called partially function-hiding inner product FE. After giving an overview of the quadratic FE, we define partially function-hiding inner-product FE in Section 3.1, and we use it build a simulation-secure quadratic FE in Section 3.2, based on the DLIN assumption in a type-3 pairing group $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$.

Overview of the quadratic FE. To encrypt the pair of vectors $\mathbf{x}$ and $\mathbf{y}$, we provide an encryption of $\mathbf{x}$ which contains group elements from $\mathbb{G}_1$, and an encryption of $\mathbf{y}$, which contains group elements from $\mathbb{G}_2$. Equipped with a pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, we multiply these encryptions to obtain the desired value in $\mathbb{G}_T$. A natural starting point is to use the ElGamal encryption [ElG84]. That is, the ciphertext $\mathsf{ct}_{\mathbf{x},\mathbf{y}}$ includes the encryption of $\mathbf{x} \in \mathbb{Z}^n$ in $\mathbb{G}_1$: $\mathsf{ct}_{\mathbf{x}} = (c_1 := [r]_1, c_2 := [\mathbf{x}+\mathbf{a}r]_1)$ with randomness $r \leftarrow_{\mathrm{R}} \mathbb{Z}_p$, public key $[\mathbf{a}]_1 \in \mathbb{G}_1^n$ and secret key $\mathbf{a} \in \mathbb{Z}_p^n$; and an ElGamal encryption of $\mathbf{y} \in \mathbb{Z}^m$ in $\mathbb{G}_2$: $\mathsf{ct}_{\mathbf{y}} = (c_3 := [s]_2, c_4 := [\mathbf{y}+\mathbf{b}s]_2)$, with randomness $s \leftarrow_{\mathrm{R}} \mathbb{Z}_p$, public key $[\mathbf{b}]_2 \in \mathbb{G}_2^m$, and secret key $\mathbf{b} \in \mathbb{Z}_p^m$. Decryption computes the product $c_2^\top \mathbf{F} c_4$, using the pairing, to recover:

$$[\mathbf{x}^\top \mathbf{F}\mathbf{y} + \underbrace{(\mathbf{a}r)^\top \mathbf{F}\mathbf{y} + \mathbf{x}^\top \mathbf{F}\mathbf{b}s + (\mathbf{a}r)^\top \mathbf{F}\mathbf{b}s)}_{\text{extra terms}}]_T,$$

where the output $[\mathbf{x}^\top \mathbf{F}\mathbf{y}]_T$ is masked by extra terms, which can be expressed as the inner product between $\begin{pmatrix} r \cdot \mathbf{y} \\ s \cdot \mathbf{x} \\ r \cdot s \end{pmatrix}$ and $\begin{pmatrix} \mathbf{F}^\top \mathbf{a} \\ \mathbf{F}\mathbf{b} \\ \mathbf{a}^\top \mathbf{F}\mathbf{b} \end{pmatrix}$.

Note that the first vector only contains elements known to the encryptor (the randomness used by the encryption and the input vectors $\mathbf{x}$ and $\mathbf{y}$), while the second vector only contains elements known to the decryption key generator (the master secret key $\mathsf{msk} = (\mathbf{a}, \mathbf{b})$, and the input $\mathbf{F}$ to the key generation algorithm). Besides, the dimension of both vectors are linear in $n + m$. Thus, to compute these extra terms (without compromising succinctness), we use an FE for inner products $\mathsf{IPFE.Enc}, \mathsf{IPFE.KeyGen}$, and we add $\mathsf{IPFE.Enc}\begin{pmatrix} r \cdot \mathbf{y} \\ s \cdot \mathbf{x} \\ r \cdot s \end{pmatrix}$ to the ciphertext $\mathsf{ct}_{\mathbf{x},\mathbf{y}}$, and we define $\mathsf{sk}_{\mathbf{F}} = \mathsf{IPFE.KeyGen}\begin{pmatrix} \mathbf{F}^\top \mathbf{a} \\ \mathbf{F}\mathbf{b} \\ \mathbf{a}^\top \mathbf{F}\mathbf{b} \end{pmatrix}$. This underlying inner-product FE needs to be function-hiding, since revealing the vector input to $\mathsf{IPFE.KeyGen}$, which contains the master secret key $\mathsf{msk} = (\mathbf{a}, \mathbf{b})$, would be fatal for the security of the ElGamal encryptions. However, function-hiding FE is an inherently private-key primitive, since a public encryption would allow to recover $\mathbf{y}$ from the decryption key $\mathsf{sk}_{\mathbf{y}}$, simply by encrypting sufficiently many well-chosen vectors $\mathbf{x}$ and decrypting them using $\mathsf{sk}_{\mathbf{y}}$.

To obtain a public-key quadratic FE, we use an encryption scheme that has more structure than ElGamal, namely, Damgård ElGamal [Dam92]. This gives the possibility to relax the function-hiding property required from the inner-product FE, and bypass the impossibility result for public-key function-hiding FE.

Namely, the ciphertext $\mathsf{ct}_{\mathbf{x},\mathbf{y}}$ contains a Damgård ElGamal encryption in $\mathbb{G}_1$: $\mathsf{ct}_{\mathbf{x}} = (c_1 := [\mathbf{a}r]_1, c_2 := [\mathbf{x} + \mathbf{U}\mathbf{a}r]_1)$ with randomness $r \leftarrow_{\mathrm{R}} \mathbb{Z}_p$, public key $([\mathbf{a}]_1 \in \mathbb{G}_1^2, [\mathbf{U}\mathbf{a}]_1 \in \mathbb{G}_1^n)$, and secret key $\mathbf{U} \in \mathbb{Z}_p^{n \times 2}$; and a Damgård ElGamal encryption of $\mathbf{y} \in \mathbb{Z}^m$ in $\mathbb{G}_2$: $\mathsf{ct}_{\mathbf{y}} = (c_3 := [\mathbf{b}s]_2, c_4 := [\mathbf{y} + \mathbf{V}\mathbf{b}s]_2)$, with randomness $s \leftarrow_{\mathrm{R}} \mathbb{Z}_p$, public key $([\mathbf{b}]_2 \in \mathbb{G}_2, [\mathbf{V}\mathbf{b}]_2 \in \mathbb{G}_2^m)$, and secret key $\mathbf{V} \in \mathbb{Z}_p^{m \times 2}$. Decryption computes the product $c_2^\top \mathbf{F} c_4$, using the pairing, to recover:

$$[\mathbf{x}^\top \mathbf{F}\mathbf{y} + \underbrace{(\mathbf{a}r)^\top (\mathbf{U}^\top \mathbf{F})(\mathbf{y} + \mathbf{V}\mathbf{b}s) + \mathbf{x}^\top (\mathbf{F}\mathbf{V})(\mathbf{b}s)}_{\text{extra terms}}]_T,$$

where the output $[\mathbf{x}^\top \mathbf{F}\mathbf{y}]_T$ is masked by extra terms, which can be expressed as the inner product between $\begin{pmatrix} \mathbf{a}r \otimes (\mathbf{y} + \mathbf{V}\mathbf{b}s) \\ \mathbf{x} \otimes \mathbf{b}s \end{pmatrix}$ and $\begin{pmatrix} \mathsf{vect}(\mathbf{U}^\top \mathbf{F}) \\ \mathsf{vect}(\mathbf{F}\mathbf{V}) \end{pmatrix}$, where for any vector $\mathbf{x} \in \mathbb{Z}_p^n$, $\mathbf{y} \in \mathbb{Z}_p^m$, and matrix $\mathbf{M} \in \mathbb{Z}_p^{n \times m}$, we denote by $\mathsf{vect}(\mathbf{M}) \in \mathbb{Z}_p^{nm}$ the vector such that the inner product of $\mathbf{x} \otimes \mathbf{y}$ with $\mathsf{vect}(\mathbf{M})$ is $\mathbf{x}^\top \mathbf{M}\mathbf{y}$.

As before, the first vector only contains elements known to the encryptor (the randomness used by the encryption, the input vectors $\mathbf{x}$ and $\mathbf{y}$, and the public keys), while the second vector only contains elements known to the decryption key generator (the master secret key $\mathsf{msk} = (\mathbf{U}, \mathbf{V})$, and the input $\mathbf{F}$ to the key generation algorithm). Besides, the dimension of both vectors are linear in $n + m$.

As before, to compute these extra terms, we use an FE for inner products $\mathsf{IPFE.Enc}$, $\mathsf{IPFE.KeyGen}$, and we add $\mathsf{IPFE.Enc}\begin{pmatrix}\mathbf{a}r\otimes(\mathbf{y}+\mathbf{V}\mathbf{b}s)\\ \mathbf{x}\otimes\mathbf{b}s\end{pmatrix}$ to the ciphertext $\mathsf{ct}_{\mathbf{x},\mathbf{y}}$, and we define $\mathsf{sk}_{\mathbf{F}}=\mathsf{IPFE.KeyGen}\begin{pmatrix}\mathsf{vect}(\mathbf{U}^\top\mathbf{F})\\ \mathsf{vect}(\mathbf{F}\mathbf{V})\end{pmatrix}$.

Now, we make the crucial observation that the underlying function-hiding FE for inner products is only used for vectors that lie in some specific subspace, strictly included in the whole space, namely, vectors (column) spanned by the matrix: $\mathbf{M}:=\left(\begin{array}{c|c}\mathbf{a}\otimes(\mathrm{Id}_m|\mathbf{V}\mathbf{b}) & \mathbf{0}\\ \hline \mathbf{0} & \mathrm{Id}_n\otimes\mathbf{b}\end{array}\right)$, where Id_n (resp. Id_m) denotes the identity matrix of dimension n (resp. m). Thus, we create, and make public, a restricted key that can only generate ciphertexts for these vectors, while still providing some meaningful function-hiding. Namely, we prove that only $\mathbf{M}^\top\mathbf{y}$ leaks from a decryption key $\mathsf{sk}_{\mathbf{y}}$ (in addition to what is supposed to leak by correctness of the scheme), which turns out to be sufficient for the security proof of the overall quadratic FE. [Lin17] also builds quadratic FE from function-hiding FE for inner products, but it uses [ABDP15] encryption of $\mathbf{x}$ and $\mathbf{y}$ in the ciphertext, and it requires a full-fledged function-hiding FE, which can only be private-key.

3.1 Partially Function-Hiding Inner-Product FE

A partially function-hiding functional encryption for inner products is defined with respect to a pairing group $\mathcal{PG}:=(\mathbb{G}_1,\mathbb{G}_2,P_1,P_2,e)\leftarrow\mathsf{PGGen}(1^\lambda)$, a full rank matrix $\mathbf{M}\in\mathbb{Z}_p^{n\times m}$, with $n>m$, and such that $\mathbf{M}^\top\mathbf{M}\in\mathbb{Z}_p^{m\times m}$ is invertible; and a tag space $\mathcal{T}$. It consists of the following PPT algorithms:

- $\mathsf{Setup}(1^\lambda,\mathcal{PG},[\mathbf{M}]_1)$: returns the public key pk (implicitly input of all other algorithms), and the master secret key msk. We assume pk contains a description of $[\mathbf{M}]_1$ and $\mathcal{T}$.
- $\mathsf{Enc}(\mathbf{t}\in\mathbb{Z}_p^m,\tau\in\mathcal{T})$: returns a ciphertext $\mathsf{ct}_{\mathbf{Mt}}$, associated with vector $\mathbf{Mt}\in\mathbb{Z}_p^n$ and tag τ.
- $\mathsf{Enc}'(\mathsf{msk},[\mathbf{x}]_1\in\mathbb{G}_1^n,\tau\in\mathcal{T})$: returns a ciphertext $\mathsf{ct}_{\mathbf{x}}$, associated with vector $\mathbf{x}\in\mathbb{Z}_p^n$ and tag τ.
- $\mathsf{KeyGen}(\mathsf{msk},\mathbf{y}\in\mathbb{Z}_p^n)$: returns a decryption key $\mathsf{sk}_{\mathbf{y}}$.
- $\mathsf{Dec}(\tau,\mathsf{ct}_{\mathbf{x}},\mathsf{sk}_{\mathbf{y}})$: deterministic algorithm that returns a value in $\mathbb{G}_T$, or $\perp$ if it fails.

Note that Enc is public key, and can only encrypt vectors in the span of $[\mathbf{M}]_1$, while Enc' needs the msk, but can encrypt any vector $[\mathbf{x}]_1\in\mathbb{G}_1^n$. Another crucial difference is that Enc' works on vector of group elements, while Enc needs to get the exponents as input. We require these two encryption algorithms agree on the their common input space, namely: for all $\mathbf{t}\in\mathbb{Z}_p^m$ and $\tau\in\mathcal{T}$, $\mathsf{Enc}(\mathbf{t},\tau)$ is identically distributed from $\mathsf{Enc}'(\mathsf{msk},[\mathbf{Mt}]_1,\tau)$, where $(\mathsf{pk},\mathsf{msk})\leftarrow\mathsf{Setup}(1^\lambda,\mathcal{PG},[\mathbf{M}]_1)$.

To build quadratic FE in Section 3, we require a tag-free partially function-hiding inner-product FE (which corresponds to the case $\mathcal{T}:=\{\varepsilon\}$). The latter can be obtained generically from any tag-based partially function-hiding inner-product FE, using one-time signature.

Correctness. For all $\mathbf{t}\in\mathbb{Z}_p^m$, $\mathbf{y}\in\mathbb{Z}_p^n$, $\tau\in\mathcal{T}$, $\Pr[\mathsf{Dec}(\tau,\mathsf{ct}_{\mathbf{Mt}},\mathsf{sk}_{\mathbf{y}})=[(\mathbf{Mt})^\top\mathbf{y}]_T]=1$, where the probability is taken over $(\mathsf{pk},\mathsf{msk})\leftarrow\mathsf{Setup}(1^\lambda,\mathcal{PG},[\mathbf{M}]_1)$, $\mathsf{ct}_{\mathbf{Mt}}\leftarrow\mathsf{Enc}(\mathbf{t},\tau)$, $\mathsf{sk}_{\mathbf{y}}\leftarrow\mathsf{KeyGen}(\mathsf{msk},\mathbf{y})$.

Security. We define simulation security for partially function-hiding inner-product FE, which captures the fact that the only information that leaks from a ciphertext $\mathsf{ct}_{\mathbf{x}}$ and keys $\mathsf{sk}_{\mathbf{y}}$ is $\mathbf{x}^\top\mathbf{y}$, and some partial information on $\mathbf{y}$, namely, $\mathbf{M}(\mathbf{M}^\top\mathbf{M})^{-1}\mathbf{M}^\top\mathbf{y}$. We give both CPA and CCA variant of security notions.

Definition 7 (partially function-hiding, CPA Simulation security). *For any inner-product FE scheme* FE*, any PPT simulator* $\mathcal{S}:=(\widetilde{\mathsf{Setup}},\widetilde{\mathsf{Enc}},\widetilde{\mathsf{KeyGen}})$*, and any PPT stateful adversary* $\mathcal{A}$*, we define the following two experiments.*

$\underline{\mathsf{Real}^{\mathsf{CPA\text{-}FE}}_{\mathcal{A}}(1^\lambda)}$:

$(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{PG}, [\mathbf{M}]_1)$

$(\tau^\star, \mathbf{x}) \leftarrow \mathcal{A}(1^\lambda, \mathsf{pk})$

$\mathsf{ct}^\star \leftarrow \mathsf{Enc}'(\mathsf{msk}, [\mathbf{x}]_1, \tau^\star)$

$\alpha \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KeyGen}}(\cdot)}(\mathsf{ct}^\star)$

$\underline{\mathsf{Ideal}^{\mathsf{CPA\text{-}FE}}_{\mathcal{A},\mathcal{S}}(1^\lambda)}$:

$(\widetilde{\mathsf{pk}}, \widetilde{\mathsf{msk}}) \leftarrow \widetilde{\mathsf{Setup}}(1^\lambda, \mathcal{PG}, [\mathbf{M}]_1)$

$(\tau^\star, \mathbf{x}) \leftarrow \mathcal{A}(1^\lambda, \widetilde{\mathsf{pk}})$

$\mathsf{ct}^\star \leftarrow \widetilde{\mathsf{Enc}}(\widetilde{\mathsf{msk}}, \tau^\star)$

$\alpha \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KeyGen}}(\cdot)}(\mathsf{ct}^\star)$

In the real experiment, the key generation oracle $\mathcal{O}_{\mathsf{KeyGen}}$, when given as input $\mathbf{y} \in \mathbb{Z}_p^n$, returns $\mathsf{KeyGen}(\mathsf{msk}, \mathbf{y})$. In the ideal experiment, when $\mathcal{O}_{\mathsf{KeyGen}}$ is given as input $\mathbf{y} \in \mathbb{Z}_p^n$, it computes $\mathbf{x}^\top\mathbf{y}$, $\widetilde{\mathbf{y}} := \mathbf{M}(\mathbf{M}^\top\mathbf{M})^{-1}\mathbf{M}^\top\mathbf{y}$, and returns $\widetilde{\mathsf{KeyGen}}(\widetilde{\mathsf{msk}}, \mathbf{x}^\top\mathbf{y}, \widetilde{\mathbf{y}})$.

We say an FE scheme is partially function-hiding simulation-secure if there exists a PPT simulator $\mathcal{S} := (\widetilde{\mathsf{Setup}}, \widetilde{\mathsf{Enc}}, \widetilde{\mathsf{KeyGen}})$ such that for all PPT adversaries $\mathcal{A}$, we have:

$$\mathrm{Adv}^{\mathsf{CPA\text{-}PFH\text{-}SIM}}_{\mathsf{FE},\mathcal{A}}(\lambda) := |\Pr[1 \leftarrow \mathsf{Real}^{\mathsf{CPA\text{-}FE}}_{\mathcal{A}}(1^\lambda)] - \Pr[1 \leftarrow \mathsf{Ideal}^{\mathsf{CPA\text{-}FE}}_{\mathcal{A},\mathcal{S}}(1^\lambda)]| = \mathsf{negl}(\lambda).$$

Definition 8 (partially function-hiding, CCA Simulation security). *For any inner-product FE scheme FE, any PPT simulator $\mathcal{S} := (\widetilde{\mathsf{Setup}}, \widetilde{\mathsf{Enc}}, \widetilde{\mathsf{KeyGen}}, \widetilde{\mathsf{Dec}})$, and any PPT stateful adversary $\mathcal{A}$, we define the following two experiments.*

$\underline{\mathsf{Real}^{\mathsf{CCA\text{-}FE}}_{\mathcal{A}}(1^\lambda)}$:

$(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{PG}, [\mathbf{M}]_1)$

$(\tau^\star, \mathbf{x}) \leftarrow \mathcal{A}(1^\lambda, \mathsf{pk})$

$\mathsf{ct}^\star \leftarrow \mathsf{Enc}'(\mathsf{msk}, [\mathbf{x}]_1, \tau^\star)$

$\alpha \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KeyGen}}(\cdot), \mathcal{O}_{\mathsf{Dec}}(\cdot,\cdot,\cdot)}(\mathsf{ct}^\star)$

$\underline{\mathsf{Ideal}^{\mathsf{CCA\text{-}FE}}_{\mathcal{A},\mathcal{S}}(1^\lambda)}$:

$(\widetilde{\mathsf{pk}}, \widetilde{\mathsf{msk}}) \leftarrow \widetilde{\mathsf{Setup}}(1^\lambda, \mathcal{PG}, [\mathbf{M}]_1)$

$(\tau^\star, \mathbf{x}) \leftarrow \mathcal{A}(1^\lambda, \widetilde{\mathsf{pk}})$

$\mathsf{ct}^\star \leftarrow \widetilde{\mathsf{Enc}}(\widetilde{\mathsf{msk}}, \tau^\star)$

$\alpha \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KeyGen}}(\cdot), \mathcal{O}_{\mathsf{Dec}}(\cdot,\cdot,\cdot)}(\mathsf{ct}^\star)$

In the real experiment, $\mathcal{O}_{\mathsf{KeyGen}}(\mathbf{y} \in \mathbb{Z}_p^n)$ returns $\mathsf{KeyGen}(\mathsf{msk}, \mathbf{y})$. The oracle $\mathcal{O}_{\mathsf{Dec}}(\tau, \mathsf{ct}, \mathbf{y})$ returns $\perp$ if $\tau = \tau^\star$; otherwise, it computes $\mathsf{sk}_{\mathbf{y}} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathbf{y})$, and returns $\mathsf{Dec}(\tau, \mathsf{ct}, \mathsf{sk}_{\mathbf{y}})$.

In the ideal experiment, $\mathcal{O}_{\mathsf{KeyGen}}(\mathbf{y} \in \mathbb{Z}_p^n)$ computes $\mathbf{x}^\top\mathbf{y}$, $\widetilde{\mathbf{y}} := \mathbf{M}(\mathbf{M}^\top\mathbf{M})^{-1}\mathbf{M}^\top\mathbf{y}$, and returns $\widetilde{\mathsf{KeyGen}}(\widetilde{\mathsf{msk}}, \mathbf{x}^\top\mathbf{y}, \widetilde{\mathbf{y}})$. The oracle $\mathcal{O}_{\mathsf{Dec}}(\tau, \mathsf{ct}, \mathbf{y})$ returns $\perp$ if $\tau \neq \tau^\star$; otherwise, it computes $\widetilde{\mathbf{y}} := \mathbf{M}(\mathbf{M}^\top\mathbf{M})^{-1}\mathbf{M}^\top\mathbf{y}$, and returns $\widetilde{\mathsf{Dec}}(\tau, \mathsf{ct}, \widetilde{\mathbf{y}})$.

We say an FE scheme is CCA partially function-hiding, simulation secure if there exists a PPT simulator $\mathcal{S} := (\widetilde{\mathsf{Setup}}, \widetilde{\mathsf{Enc}}, \widetilde{\mathsf{KeyGen}}, \widetilde{\mathsf{Dec}})$ such that for all PPT adversaries $\mathcal{A}$, we have:

$$\mathrm{Adv}^{\mathsf{CCA\text{-}PFH\text{-}SIM}}_{\mathsf{FE},\mathcal{A}}(\lambda) := |\Pr[1 \leftarrow \mathsf{Real}^{\mathsf{CCA\text{-}FE}}_{\mathcal{A}}(1^\lambda)] - \Pr[1 \leftarrow \mathsf{Ideal}^{\mathsf{CCA\text{-}FE}}_{\mathcal{A},\mathcal{S}}(1^\lambda)]| = \mathsf{negl}(\lambda).$$

3.2 Quadratic FE from Partially Function-Hiding Inner-Product FE

We describe our quadratic FE in Fig.4, for the functionality $\mathcal{F}_{\mathsf{quad},B}$. Its security relies on the security of the underlying partially function-hiding inner-product FE, the bilateral DLIN assumption, and the DDH assumption in $\mathbb{G}_1$.

$\underline{\mathsf{Setup}(1^\lambda, \mathcal{F}_{\mathsf{quad},B})}$:

$\mathcal{PG} := (\mathbb{G}_1, \mathbb{G}_2, g_1, g_2, p, e) \leftarrow \mathsf{PGGen}(1^\lambda)$, $\mathbf{a} \leftarrow_{\mathrm{R}} \mathsf{DDH}$, $\mathbf{B} \leftarrow_{\mathrm{R}} \mathsf{DLIN}$, $\mathbf{U} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{n\times 2}$, $\mathbf{V} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{m\times 3}$

$\mathbf{M} := \left(\begin{array}{c|c} \mathbf{a} \otimes (\mathrm{Id}_m | \mathbf{VB}) & \mathbf{0} \\ \hline \mathbf{0} & \mathrm{Id}_n \otimes \mathbf{B} \end{array}\right)$, $(\mathsf{pk}_{\mathsf{IPFE}}, \mathsf{msk}_{\mathsf{IPFE}}) \leftarrow \mathsf{Setup}_{\mathsf{IPFE}}(1^\lambda, \mathcal{PG}, [\mathbf{M}]_1)$

Return $\mathsf{pk} := ([\mathbf{Ua}]_1, [\mathbf{VB}]_2, \mathsf{pk}_{\mathsf{IPFE}})$, $\mathsf{msk} := (\mathbf{U}, \mathbf{V}, \mathsf{msk}_{\mathsf{IPFE}})$

$\underline{\mathsf{Enc}(\mathbf{x}, \mathbf{y})}$:

$r \leftarrow_{\mathrm{R}} \mathbb{Z}_p$, $\mathsf{ct}_{\mathbf{x}} := \mathbf{x} + \mathbf{U}\mathbf{a}r$, $\mathbf{s} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^2$, $\mathsf{ct}_{\mathbf{y}} := \mathbf{y} + \mathbf{VBs}$, $\mathsf{ct}_{\mathsf{IPFE}} := \mathsf{Enc}_{\mathsf{IPFE}}\left(\mathsf{pk}_{\mathsf{IPFE}}, \begin{pmatrix} r \otimes \binom{\mathbf{y}}{\mathbf{s}} \\ \mathbf{x} \otimes \mathbf{s} \end{pmatrix}\right)$

Return $\mathsf{ct}_{\mathbf{x},\mathbf{y}} := ([\mathsf{ct}_{\mathbf{x}}]_1, [\mathsf{ct}_{\mathbf{y}}]_2, \mathsf{ct}_{\mathsf{IPFE}})$

$\underline{\mathsf{KeyGen}(\mathsf{msk}, \mathbf{F})}$:

$\mathsf{sk}_{\mathsf{IPFE}} \leftarrow \mathsf{KeyGen}_{\mathsf{IPFE}}\left(\mathsf{msk}_{\mathsf{IPFE}}, \begin{pmatrix} \mathsf{vect}(\mathbf{U}^\top \mathbf{F}) \\ \mathsf{vect}(\mathbf{FV}) \end{pmatrix}\right)$

Return $\mathsf{sk}_{\mathbf{F}} := (\mathbf{F}, \mathsf{sk}_{\mathsf{IPFE}})$

$\underline{\mathsf{Dec}\big(\mathsf{ct}_{\mathbf{x},\mathbf{y}} := ([\mathsf{ct}_{\mathbf{x}}]_1, [\mathsf{ct}_{\mathbf{y}}]_2, \mathsf{ct}_{\mathsf{IPFE}}), \mathsf{sk}_{\mathbf{F}} := (\mathbf{F}, \mathsf{sk}_{\mathsf{IPFE}})\big)}$:

$[d]_T \leftarrow \mathsf{Dec}_{\mathsf{IPFE}}(\mathsf{ct}_{\mathsf{IPFE}}, \mathsf{sk}_{\mathsf{IPFE}})$, $[v]_T := e([\mathsf{ct}_{\mathbf{x}}]_1^\top, [\mathbf{F}\mathsf{ct}_{\mathbf{y}}]_2) - [d]_T$

If $v \in [0, n \cdot m \cdot B^3]$, return $\log([v]_T)$. Otherwise, return $\perp$.

Fig. 4. Quadratic FE: Quad. Here, $\mathsf{IPFE} := (\mathsf{Setup}_{\mathsf{IPFE}}, \mathsf{Enc}_{\mathsf{IPFE}}, \mathsf{Enc}'_{\mathsf{IPFE}}, \mathsf{KeyGen}_{\mathsf{IPFE}}, \mathsf{Dec}_{\mathsf{IPFE}})$ is a (tag-free) partially function-hiding inner-product FE, as defined in Section 3.1.

Correctness. By correctness of the underlying inner-product FE, we have:

$$\begin{aligned} d &= \begin{pmatrix} r \otimes \binom{\mathbf{y}}{\mathbf{s}} \\ \mathbf{x} \otimes \mathbf{s} \end{pmatrix}^\top \mathbf{M}^\top \begin{pmatrix} \mathsf{vect}(\mathbf{U}^\top \mathbf{F}) \\ \mathsf{vect}(\mathbf{FV}) \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{a}r \otimes \mathsf{ct}_{\mathbf{y}} \\ \mathbf{x} \otimes \mathbf{Bs} \end{pmatrix}^\top \begin{pmatrix} \mathsf{vect}(\mathbf{U}^\top \mathbf{F}) \\ \mathsf{vect}(\mathbf{FV}) \end{pmatrix} \\ &= (\mathbf{U}\mathbf{a}r)^\top \mathbf{F}\mathsf{ct}_{\mathbf{y}} + \mathbf{x}^\top \mathbf{FVBs}, \end{aligned}$$

which corresponds exactly to the extra terms obtained when computing $\mathsf{ct}_{\mathbf{x}}^\top \mathbf{F}\mathsf{ct}_{\mathbf{y}}$, that is, we have: $\mathsf{ct}_{\mathbf{x}}^\top \mathbf{F}\mathsf{ct}_{\mathbf{y}} = \mathbf{x}^\top \mathbf{Fy} + d$. Finally, Dec computes the discrete log of $[\mathbf{x}^\top \mathbf{Fy}]_T$, which is efficient since the output $\mathbf{x}^\top \mathbf{Fy}$ is bounded by $n \cdot m \cdot B^3$, which is a polynomial in the security parameter.

Theorem 1 (Simulation security of the quadratic FE). *The quadratic FE from Fig. 4 for the functionality is simulation CPA (resp. CCA) secure if the underlying inner-product FE is partially function-hiding CPA (resp. CCA) simulation secure, assuming the bilateral DLIN assumption and the DDH assumption in* $\mathbb{G}_1$.

Namely, for any PPT adversary $\mathcal{A}$*, there exist PPT adversaries* $\mathcal{B}_1$*,* $\mathcal{B}_2$*, and* $\mathcal{B}_3$ *such that:*

$$\mathrm{Adv}^{\mathsf{CPA\text{-}SIM}}_{\mathsf{Quad},\mathcal{A}} \leq \mathrm{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1,\mathcal{B}_1}(\lambda) + \mathrm{Adv}^{\mathsf{DLIN}}_{\mathcal{PG},\mathcal{B}_2}(\lambda) + \mathrm{Adv}^{\mathsf{CPA\text{-}PFH\text{-}SIM}}_{\mathsf{IPFE},\mathcal{B}_3}(\lambda) + \frac{4}{p}.$$

Besides, for any PPT adversary $\mathcal{A}'$*, there exist PPT adversaries* $\mathcal{B}'_1$*,* $\mathcal{B}'_2$*, and* $\mathcal{B}'_3$ *such that:*

$$\mathrm{Adv}^{\mathsf{CCA\text{-}SIM}}_{\mathsf{Quad},\mathcal{A}'} \leq \mathrm{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1,\mathcal{B}'_1}(\lambda) + \mathrm{Adv}^{\mathsf{DLIN}}_{\mathcal{PG},\mathcal{B}'_2}(\lambda) + \mathrm{Adv}^{\mathsf{CCA\text{-}PFH\text{-}SIM}}_{\mathsf{IPFE},\mathcal{B}'_3}(\lambda) + \frac{4}{p}.$$

Proof. We prove the second part of the theorem, that is, CCA security. The CPA security proof is a straightforward simplification, hence omitted. Let $\mathcal{A}$ be a PPT adversary against the CCA security of Quad. We use a sequence of hybrid games Game_i for $i \in \{1,2,3,4\}$, defined in Fig.5, and we denote the advantage $\varepsilon_i := \Pr[1 \leftarrow \mathsf{Game}_i(\mathcal{A}, 1^\lambda)]$. We show that these games are computationally indistinguishable: $\mathsf{Real}_{\mathcal{A}}^{\mathsf{CCA\text{-}Quad}}(1^\lambda) \equiv \mathsf{Game}_1 \approx_c \mathsf{Game}_2 \approx_c \mathsf{Game}_3 \approx_c \mathsf{Game}_4 \approx_s \mathsf{Ideal}_{\mathcal{A},\mathcal{S}}^{\mathsf{CCA\text{-}Quad}}(1^\lambda)$, for the PPT simulator $\mathcal{S} := (\widetilde{\mathsf{Setup}}, \widetilde{\mathsf{Enc}}, \widetilde{\mathsf{KeyGen}}, \widetilde{\mathsf{Dec}})$ defined in Fig.6.

Game_1*:* is as $\mathsf{Real}_{\mathcal{A}}^{\mathsf{CCA\text{-}Quad}}(1^\lambda)$, except the encryption algorithm Enc' is used instead of Enc. Since these algorithms are identically distributed on input vectors in the span on $[\mathbf{M}]_1$, this does not change the advantage of $\mathcal{A}$:

$$\Pr[1 \leftarrow \mathsf{Real}_{\mathcal{A}}^{\mathsf{CCA\text{-}Quad}}(1^\lambda)] = \varepsilon_1.$$

Game_2*:* we use the DDH assumption in $\mathbb{G}_1$ to switch the distribution of the vector $[\mathbf{a}r]_1$ in the challenge ciphertext to uniformly random over $\mathbb{G}_1^2$. Namely, we build a PPT adversary $\mathcal{B}_1$ against the DDH assumption such that

$$\varepsilon_2 - \varepsilon_1 \leq \mathrm{Adv}_{\mathbb{G}_1,\mathcal{B}_1}^{\mathsf{DDH}}(\lambda).$$

Upon receiving the DDH challenge $(\mathcal{PG}, [\mathbf{a}]_1, [\mathbf{c}]_1)$, $\mathcal{B}_1$ samples $\mathbf{U} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{n\times 2}$, $\mathbf{V} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{m\times 3}$, $\mathbf{B} \leftarrow \mathsf{DLIN}$, computes $[\mathbf{M}]_1$ as defined in Fig.4, and runs $(\mathsf{pk}_{\mathsf{IPFE}}, \mathsf{msk}_{\mathsf{IPFE}}) \leftarrow \mathsf{Setup}_{\mathsf{IPFE}}(1^\lambda, \mathcal{PG}, [\mathbf{M}]_1)$, tanks to which it can simulate the public key pk for $\mathcal{A}$, and answer its queries to $\mathcal{O}_{\mathsf{KeyGen}}$ and $\mathcal{O}_{\mathsf{Dec}}$. When $\mathcal{A}$ submits its challenge $(\mathbf{x}, \mathbf{y})$, $\mathcal{B}_1$ samples $\mathbf{s} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^2$, computes $[\mathsf{ct}_{\mathbf{x}}]_1 := [\mathbf{x} + \mathbf{U}\mathbf{c}]_1$, $[\mathsf{ct}_{\mathbf{y}}]_2 := [\mathbf{y} + \mathbf{V}\mathbf{B}\mathbf{s}]_2$, $[\mathbf{z}]_1 := \begin{bmatrix} \mathbf{c} \otimes \mathsf{ct}_{\mathbf{y}} \\ \mathbf{x} \otimes \mathbf{B}\mathbf{s} \end{bmatrix}_1$ and returns the challenge ciphertext $([\mathsf{ct}_{\mathbf{x}}]_1, [\mathsf{ct}_{\mathbf{y}}]_2, \mathsf{Enc}'_{\mathsf{IPFE}}(\mathsf{msk}_{\mathsf{IPFE}}, [\mathbf{z}]_1))$ to $\mathcal{A}$. When $[\mathbf{c}]_1$ is a real DDH challenge, that is, of the form $[\mathbf{c}]_1 := [\mathbf{a}r]_1$ for some $r \leftarrow_{\mathrm{R}} \mathbb{Z}_p$, $\mathcal{B}_1$ simulates Game_1, whereas it simulates Game_2 when $[\mathbf{c}]_1$ is uniformly random over $\mathbb{G}_1^2$.

Game_3*:* we use the bilateral DLIN assumption to switch the distribution of the vector $[\mathbf{B}\mathbf{s}]_2$ to uniformly random over $\mathbb{G}_2^3$. Namely, we build a PPT adversary $\mathcal{B}_2$ such

$$\varepsilon_3 - \varepsilon_2 \leq \mathrm{Adv}_{\mathcal{PG},\mathcal{B}_2}^{\mathsf{DLIN}}(\lambda).$$

Upon receiving the DLIN challenge $(\mathcal{PG}, \{[\mathbf{B}]_s, [\mathbf{t}]_s\}_{s\in\{1,2\}})$, $\mathcal{B}_2$ samples $\mathbf{U} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{n\times 2}$, $\mathbf{V} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{m\times 3}$, $\mathbf{a} \leftarrow_{\mathrm{R}} \mathsf{DDH}$, computes $[\mathbf{M}]_1$ as defined in Fig.4, and runs $(\mathsf{pk}_{\mathsf{IPFE}}, \mathsf{msk}_{\mathsf{IPFE}}) \leftarrow \mathsf{Setup}_{\mathsf{IPFE}}(1^\lambda, \mathcal{PG}, [\mathbf{M}]_1)$, tanks to which it can simulate the public key pk for $\mathcal{A}$, and answer its queries to $\mathcal{O}_{\mathsf{KeyGen}}$ and $\mathcal{O}_{\mathsf{Dec}}$. When $\mathcal{A}$ submits its challenge $(\mathbf{x}, \mathbf{y})$, $\mathcal{B}$ samples $\mathbf{c} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^2$, computes $[\mathsf{ct}_{\mathbf{x}}]_1 := [\mathbf{x} + \mathbf{U}\mathbf{c}]_1$, $[\mathsf{ct}_{\mathbf{y}}]_2 := [\mathbf{y} + \mathbf{V}\mathbf{t}]_2$, $[\mathbf{z}]_1 := \begin{bmatrix} \mathbf{c} \otimes (\mathbf{y} + \mathbf{V}\mathbf{t}) \\ \mathbf{x} \otimes \mathbf{t} \end{bmatrix}_1$ and returns the challenge ciphertext $([\mathsf{ct}_{\mathbf{x}}]_1, [\mathsf{ct}_{\mathbf{y}}]_2, \mathsf{Enc}'_{\mathsf{IPFE}}(\mathsf{msk}_{\mathsf{IPFE}}, [\mathbf{z}]_1))$ to $\mathcal{A}$. When $\{[\mathbf{t}]_s\}_{s\in\{1,2\}}$ is a real DLIN challenge, that is, of the form $[\mathbf{t}]_s := [\mathbf{B}\mathbf{s}]_s$ for some $\mathbf{s} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^2$, $\mathcal{B}_2$ simulates Game_2, whereas it simulates Game_3 when $[\mathbf{t}]_s$ is uniformly random over $\mathbb{G}_s^2$.

Game_4*:* we use the simulator $(\widetilde{\mathsf{Setup}}_{\mathsf{IPFE}}, \widetilde{\mathsf{Enc}}_{\mathsf{IPFE}}, \widetilde{\mathsf{KeyGen}}_{\mathsf{IPFE}}, \widetilde{\mathsf{Dec}}_{\mathsf{IPFE}})$ of IPFE, as described in Fig.5. Namely, there exists a PPT adversary $\mathcal{B}_3$ such that

$$\varepsilon_4 - \varepsilon_3 \leq \mathrm{Adv}_{\mathsf{IPFE},\mathcal{B}_3}^{\mathsf{CCA\text{-}PFH\text{-}SIM}}(\lambda).$$

Adversary $\mathcal{B}_3$ samples $\mathbf{a} \leftarrow_{\mathrm{R}} \mathsf{DDH}$, $\mathbf{B} \leftarrow_{\mathrm{R}} \mathsf{DLIN}$, $\mathbf{U} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{n\times 2}$, $\mathbf{V} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{m\times 3}$, and simulates $\mathcal{A}$'s view straightforwardly, using the fact that for all $\mathbf{x} \in \mathbb{Z}_p^n$, $\mathbf{y} \in \mathbb{Z}_p^m$, $\mathbf{F} \in \mathbb{Z}_p^{n\times m}$, $\mathbf{c} \in \mathbb{Z}_p^2$, $\mathbf{t} \in \mathbb{Z}_p^3$, we have:

$$\begin{pmatrix} \mathbf{c} \otimes \mathsf{ct}_{\mathbf{y}} \\ \mathbf{x} \otimes \mathbf{t} \end{pmatrix}^\top \begin{pmatrix} \mathsf{vect}(\mathbf{U}^\top \mathbf{F}) \\ \mathsf{vect}(\mathbf{F}\mathbf{V}) \end{pmatrix} = \mathsf{ct}_{\mathbf{x}}^\top \mathbf{F} \mathsf{ct}_{\mathbf{y}} - \mathbf{x}^\top \mathbf{F} \mathbf{y},$$

where $\mathsf{ct}_{\mathbf{x}} = \mathbf{x} + \mathbf{U}\mathbf{c}$, and $\mathsf{ct}_{\mathbf{y}} = \mathbf{y} + \mathbf{V}\mathbf{t}$.

Game_1, $\boxed{\mathsf{Game}_2, \boxed{\mathsf{Game}_3, \mathsf{Game}_4}}$:

$\mathcal{PG} \leftarrow \mathsf{GGen}(1^\lambda)$, $\mathbf{a} \leftarrow_{\mathrm{R}} \mathsf{DDH}$, $\mathbf{B} \leftarrow_{\mathrm{R}} \mathsf{DLIN}$, $\mathbf{U} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{n\times 2}$, $\mathbf{V} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{m\times 3}$

$$\mathbf{M} := \left(\begin{array}{c|c} \mathbf{a}\otimes(\mathrm{Id}_m|\mathbf{VB}) & \mathbf{0} \\ \hline \mathbf{0} & \mathrm{Id}_n\otimes\mathbf{B} \end{array}\right)$$

$(\mathsf{pk}_{\mathsf{IPFE}}, \mathsf{msk}_{\mathsf{IPFE}}) \leftarrow \mathsf{Setup}_{\mathsf{IPFE}}(1^\lambda, \mathcal{PG}, [\mathbf{M}]_1)$, $\mathsf{pk}' := \mathsf{pk}_{\mathsf{IPFE}}$, $\mathsf{msk}' := \mathsf{msk}_{\mathsf{IPFE}}$

$(\widetilde{\mathsf{pk}_{\mathsf{IPFE}}}, \widetilde{\mathsf{msk}_{\mathsf{IPFE}}}) \leftarrow \widetilde{\mathsf{Setup}_{\mathsf{IPFE}}}(1^\lambda, \mathcal{PG}, [\mathbf{M}]_1)$, $\mathsf{pk}' := \widetilde{\mathsf{pk}_{\mathsf{IPFE}}}$, $\mathsf{msk}' := \widetilde{\mathsf{msk}_{\mathsf{IPFE}}}$

$\mathsf{pk} := ([\mathbf{Ua}]_1, [\mathbf{VB}]_2, \mathsf{pk}')$, $\mathsf{msk} := (\mathbf{U}, \mathbf{V}, \mathsf{msk}')$

$(\mathbf{x}, \mathbf{y}) \leftarrow \mathcal{A}(\mathsf{pk})$

$\mathsf{ct}^\star \leftarrow \mathcal{O}_{\mathsf{Enc}}(\mathbf{x}, \mathbf{y})$

$\alpha \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KeyGen}}(\cdot), \mathcal{O}_{\mathsf{Dec}}(\cdot,\cdot)}(\mathsf{ct}^\star)$

$\underline{\mathcal{O}_{\mathsf{Enc}}(\mathbf{x}, \mathbf{y})}$:

$r \leftarrow_{\mathrm{R}} \mathbb{Z}_p$, $\mathbf{c} := \mathbf{a}r$, $\boxed{\mathbf{c} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^2}$, $\mathsf{ct}_{\mathbf{x}} := \mathbf{x} + \mathbf{Uc}$

$\mathbf{s} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^2$, $\mathbf{t} := \mathbf{Bs}$, $\boxed{\mathbf{t} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^3}$, $\mathsf{ct}_{\mathbf{y}} := \mathbf{y} + \mathbf{Vt}$

$$\mathsf{ct}_{\mathsf{IPFE}} := \mathsf{Enc}'_{\mathsf{IPFE}}\left(\mathsf{msk}_{\mathsf{IPFE}}, \begin{bmatrix} \mathbf{c}\otimes\mathsf{ct}_{\mathbf{y}} \\ \mathbf{x}\otimes\mathbf{t} \end{bmatrix}_1\right)$$

$\mathsf{ct}_{\mathsf{IPFE}} := \widetilde{\mathsf{Enc}_{\mathsf{IPFE}}}\left(\widetilde{\mathsf{msk}_{\mathsf{IPFE}}}\right)$

Return $([\mathsf{ct}_{\mathbf{x}}]_1, [\mathsf{ct}_{\mathbf{y}}]_2, \mathsf{ct}_{\mathsf{IPFE}})$

$\underline{\mathcal{O}_{\mathsf{KeyGen}}(\mathbf{F})}$:

$\mathsf{sk}_{\mathbf{F}} := \mathsf{KeyGen}(\mathsf{msk}, \mathbf{F})$

$$d_{\mathbf{F}} := \mathsf{ct}_{\mathbf{x}}^\top \mathbf{F} \mathsf{ct}_{\mathbf{y}} - \mathbf{x}^\top \mathbf{F}\mathbf{y}, \mathbf{v}_{\mathbf{F}} := \mathbf{M}(\mathbf{M}^\top\mathbf{M})^{-1}\mathbf{M}^\top \begin{pmatrix} \mathsf{vect}(\mathbf{U}^\top\mathbf{F}) \\ \mathsf{vect}(\mathbf{FV}) \end{pmatrix}$$

$$\mathsf{sk}_{\mathbf{F}} := \widetilde{\mathsf{KeyGen}_{\mathsf{IPFE}}}\left(\widetilde{\mathsf{msk}_{\mathsf{IPFE}}}, d_{\mathbf{F}}, \mathbf{v}_{\mathbf{F}}\right)$$

Return $\mathsf{sk}_{\mathbf{F}}$

$\underline{\mathcal{O}_{\mathsf{Dec}}(\mathsf{ct} := ([\mathbf{c}_1]_1, [\mathbf{c}_2]_2, \mathsf{ct}_{\mathsf{IPFE}}), \mathbf{F})}$:

$$\mathsf{sk}_{\mathbf{F}} := \mathsf{KeyGen}_{\mathsf{IPFE}}\left(\mathsf{msk}_{\mathsf{IPFE}}, \begin{pmatrix} \mathsf{vect}(\mathbf{U}^\top\mathbf{F}) \\ \mathsf{vect}(\mathbf{FV}) \end{pmatrix}\right), [d]_T \leftarrow \mathsf{Dec}_{\mathsf{IPFE}}(\mathsf{ct}_{\mathsf{IPFE}}, \mathsf{sk}_{\mathbf{F}})$$

$$\mathbf{v}_{\mathbf{F}} := \mathbf{M}(\mathbf{M}^\top\mathbf{M})^{-1}\mathbf{M}^\top \begin{pmatrix} \mathsf{vect}(\mathbf{U}^\top\mathbf{F}) \\ \mathsf{vect}(\mathbf{FV}) \end{pmatrix}, [d]_T \leftarrow \widetilde{\mathsf{Dec}_{\mathsf{IPFE}}}(\widetilde{\mathsf{msk}_{\mathsf{IPFE}}}, \mathsf{ct}_{\mathsf{IPFE}}, \mathbf{v}_{\mathbf{F}})$$

$[v]_T := [\mathbf{c}_1^\top \mathbf{F} \mathbf{c}_2]_T - [d]_T$

If $v \in [0, n \cdot m \cdot B^3]$, return v; otherwise, return $\perp$.

Fig. 5. Games for the security proof of the quadratic FE from Fig.4. In each procedure, the components inside a solid (dotted, gray) frame are only present in the games marked by a solid (dotted, gray) frame. $(\widetilde{\mathsf{Setup}_{\mathsf{IPFE}}}, \widetilde{\mathsf{Enc}_{\mathsf{IPFE}}}, \widetilde{\mathsf{KeyGen}_{\mathsf{IPFE}}}, \widetilde{\mathsf{Dec}_{\mathsf{IPFE}}})$ is a PPT simulator for the partially function-hiding SEL-SIM secure inner-product FE: IPFE.

$\mathsf{Ideal}^{\mathsf{CCA\text{-}Quad}}_{\mathcal{A},\mathcal{S}}(1^\lambda)$*:* is as Game_4, except that $\mathsf{ct}_{\mathbf{x}}$ and $\mathsf{ct}_{\mathbf{y}}$ in the challenge ciphertext are uniformly distributed. We show that these two games are statistically close. Namely, we show that the vectors $\mathbf{Uc}$ and $\mathbf{Vt}$ are statistically close to uniformly random over $\mathbb{Z}_p^2$ and $\mathbb{Z}_p^3$, respectively.

$\widetilde{\mathsf{Setup}}(1^\lambda, \mathcal{F}_{\mathsf{quad}})$:

$\mathcal{PG} \leftarrow \mathsf{GGen}(1^\lambda)$, $\mathbf{a} \leftarrow_{\mathrm{R}} \mathsf{DDH}$, $\mathbf{B} \leftarrow_{\mathrm{R}} \mathsf{DLIN}$, $\mathbf{U} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{n\times 2}$, $\mathbf{V} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{m\times 3}$

$\mathbf{M} := \left(\begin{array}{c|c} \mathbf{a} \otimes (\mathrm{Id}_m | \mathbf{VB}) & \mathbf{0} \\ \hline \mathbf{0} & \mathrm{Id}_n \otimes \mathbf{B} \end{array}\right)$, $(\widetilde{\mathsf{pk}}_{\mathsf{IPFE}}, \widetilde{\mathsf{msk}}_{\mathsf{IPFE}}) \leftarrow \widetilde{\mathsf{Setup}}_{\mathsf{IPFE}}(1^\lambda, \mathcal{PG}, [\mathbf{M}]_1)$

Return $\widetilde{\mathsf{pk}} := \left([\mathbf{Ua}]_1, [\mathbf{VB}]_2, \widetilde{\mathsf{pk}}_{\mathsf{IPFE}}\right)$, $\widetilde{\mathsf{msk}} := \left(\mathbf{U}, \mathbf{V}, \widetilde{\mathsf{msk}}_{\mathsf{IPFE}}\right)$

$\widetilde{\mathsf{Enc}}(\widetilde{\mathsf{msk}})$:

$\mathsf{ct}_{\mathbf{x}} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^n$, $\mathsf{ct}_{\mathbf{y}} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^m$

Return $\mathsf{ct}_{\mathbf{x},\mathbf{y}} := \left([\mathsf{ct}_{\mathbf{x}}]_1, [\mathsf{ct}_{\mathbf{y}}]_2, \widetilde{\mathsf{Enc}}_{\mathsf{IPFE}}\left(\widetilde{\mathsf{msk}}_{\mathsf{IPFE}}\right)\right)$

$\widetilde{\mathsf{KeyGen}}(\widetilde{\mathsf{msk}}, \mathbf{F}, \mathbf{x}^\top \mathbf{F} \mathbf{y})$:

$d_{\mathbf{F}} := \mathsf{ct}_{\mathbf{x}}^\top \mathbf{F} \mathsf{ct}_{\mathbf{y}} - \mathbf{x}^\top \mathbf{F} \mathbf{y}$, $\mathbf{v}_{\mathbf{F}} := \mathbf{M}(\mathbf{M}^\top \mathbf{M})^{-1} \mathbf{M}^\top \begin{pmatrix} \mathsf{vect}(\mathbf{U}^\top \mathbf{F}) \\ \mathsf{vect}(\mathbf{FV}) \end{pmatrix}$

Return $\widetilde{\mathsf{KeyGen}}_{\mathsf{IPFE}}\left(\widetilde{\mathsf{msk}}_{\mathsf{IPFE}}, d_{\mathbf{F}}, \mathbf{v}_{\mathbf{F}}\right)$

$\widetilde{\mathsf{Dec}}(\widetilde{\mathsf{msk}}, \mathsf{ct}, \mathbf{F})$:

$\mathbf{v}_{\mathbf{F}} := \mathbf{M}(\mathbf{M}^\top \mathbf{M})^{-1} \mathbf{M}^\top \begin{pmatrix} \mathsf{vect}(\mathbf{U}^\top \mathbf{F}) \\ \mathsf{vect}(\mathbf{FV}) \end{pmatrix}$, $[d]_T \leftarrow \widetilde{\mathsf{Dec}}_{\mathsf{IPFE}}(\widetilde{\mathsf{msk}}_{\mathsf{IPFE}}, \mathsf{ct}, \mathbf{v}_{\mathbf{F}})$, $[v]_T := [\mathsf{ct}_{\mathbf{x}}^\top \mathbf{F} \mathsf{ct}_{\mathbf{y}}]_T - [d]_T$.

If $v \in [0, n \cdot m \cdot B^3]$, return v; otherwise, return $\perp$.

Fig. 6. PPT simulator for the security proof of the quadratic FE from Fig.4. Here, $(\widetilde{\mathsf{Setup}}_{\mathsf{IPFE}}, \widetilde{\mathsf{Enc}}_{\mathsf{IPFE}}, \widetilde{\mathsf{KeyGen}}_{\mathsf{IPFE}}, \widetilde{\mathsf{Dec}}_{\mathsf{IPFE}})$ is a PPT simulator for the partially function-hiding, simulation secure inner-product FE, IPFE, used in the quadratic FE.

To prove so, we use the following basis of $\mathbb{Z}_p^2$ and $\mathbb{Z}_p^3$: $(\mathbf{a}|\mathbf{a}^\perp)$ and $(\mathbf{B}|\mathbf{b}^\perp)$ with $\mathbf{a} \leftarrow_{\mathrm{R}} \mathsf{DDH}$, $\mathbf{B} \leftarrow_{\mathrm{R}} \mathsf{DLIN}$, and $\mathbf{a}^\perp \in \mathbb{Z}_p^2$, $\mathbf{b}^\perp \in \mathbb{Z}_p^3$ are such that $\mathbf{a}^\top \mathbf{a}^\perp = 0$ and $\mathbf{B}^\top \mathbf{b}^\perp = \mathbf{0}$. Such basis exist assuming $\mathbf{a}$ and $\mathbf{B}$ are both full rank, which happens with probability at least $1 - \frac{2}{p}$ over the choice of $\mathbf{a} \leftarrow_{\mathrm{R}} \mathsf{DDH}$ and $\mathbf{B} \leftarrow_{\mathrm{R}} \mathsf{DLIN}$. We have: $\mathbf{U}^\top := \mathbf{a}\mathbf{u}_0^\top + \mathbf{a}^\perp \mathbf{u}_1^\top$, and $\mathbf{V}^\top := \mathbf{B}\mathbf{V}_0 + \mathbf{b}^\perp \mathbf{v}_1^\top$, with $\mathbf{u}_0, \mathbf{u}_1 \leftarrow_{\mathrm{R}} \mathbb{Z}_p^n$, $\mathbf{V}_0 \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{2\times m}$, and $\mathbf{v}_1 \leftarrow_{\mathrm{R}} \mathbb{Z}_p^m$. We will show that $\mathbf{u}_1$ and $\mathbf{v}_1$ only appear in the adversary view as $(\mathbf{c}^\top \mathbf{a}^\perp)\mathbf{u}_1$ in $\mathsf{ct}_{\mathbf{x}}$, and as $(\mathbf{t}^\top \mathbf{b}^\perp)\mathbf{v}_1$ in $\mathsf{ct}_{\mathbf{y}}$. Since we have $\mathsf{ct}_{\mathbf{x}} = \mathbf{x} + \mathbf{u}_0(\mathbf{c}^\top \mathbf{a}) + \mathbf{u}_1(\mathbf{c}^\top \mathbf{a}^\perp)$ and $\mathsf{ct}_{\mathbf{y}} = \mathbf{y} + \mathbf{V}_0^\top(\mathbf{B}^\top \mathbf{t}) + \mathbf{v}_1(\mathbf{t}^\top \mathbf{b}^\perp)$, when $\mathbf{c}^\top \mathbf{a}^\perp \neq 0$, and $\mathbf{t}^\top \mathbf{b}^\perp \neq 0$, the vectors $\mathsf{ct}_{\mathbf{x}}$ and $\mathsf{ct}_{\mathbf{y}}$ are uniformly random over $\mathbb{Z}_p^2$ and $\mathbb{Z}_p^3$, respectively. With probability $1 - \frac{2}{p}$ over the choice of $\mathbf{c} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^2$ and $\mathbf{t} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^3$, we have $\mathbf{c}^\top \mathbf{a}^\perp \neq 0$, and $\mathbf{t}^\top \mathbf{b}^\perp \neq 0$, which proves

$$\varepsilon_4 \leq \Pr\left[1 \leftarrow \mathsf{Ideal}_{\mathcal{A}}^{\mathsf{CCA\text{-}Quad}}(1^\lambda)\right] + \frac{4}{p}.$$

We now show that $\mathbf{u}_1$ and $\mathbf{v}_1$ only appear in $\mathsf{ct}_{\mathbf{x}}$ and $\mathsf{ct}_{\mathbf{y}}$, as indicated above. In the public key, we have $\mathbf{a}^\top \mathbf{U}^\top = \mathbf{a}^\top(\mathbf{a}\mathbf{u}_0^\top + \mathbf{a}^\perp \mathbf{u}_1^\top) = (\mathbf{a}^\top \mathbf{a})\mathbf{u}_0^\top$, and $\mathbf{B}^\top \mathbf{V}^\top = \mathbf{B}^\top(\mathbf{B}\mathbf{V}_0 + \mathbf{b}^\perp \mathbf{v}_1^\top) = (\mathbf{B}^\top \mathbf{B})\mathbf{v}_0^\top$. The information needed to simulate $\mathcal{O}_{\mathsf{KeyGen}}$ and $\mathcal{O}_{\mathsf{Dec}}$ is contained in $\mathbf{F}$, $\mathsf{ct}_{\mathbf{x}}$, $\mathsf{ct}_{\mathbf{y}}$, $\mathbf{x}^\top \mathbf{F} \mathbf{y}$, $\mathbf{M}$, and

$$\mathbf{M}^\top \begin{pmatrix} \mathsf{vect}((\mathbf{a}\mathbf{u}_0^\top + \mathbf{a}^\perp \mathbf{u}_1^\top)\mathbf{F}) \\ \mathsf{vect}(\mathbf{F}(\mathbf{B}\mathbf{V}_0 + \mathbf{b}^\perp \mathbf{v}_1^\top)^\top) \end{pmatrix} = \mathbf{M}^\top \begin{pmatrix} \mathsf{vect}(\mathbf{a}\mathbf{u}_0^\top \mathbf{F}) \\ \mathsf{vect}(\mathbf{FBV}_0) \end{pmatrix},$$

where the equality holds by definition of $\mathbf{M}$. That is, the only information about $\mathbf{u}_1$ and $\mathbf{v}_1$ is contained in $\mathsf{ct}_{\mathbf{x}}$, $\mathsf{ct}_{\mathbf{y}}$, which concludes the proof. □

4 Partially-Hiding Inner Product FE

In this section we build a partially-hiding, simulation-secure inner-product FE scheme, as defined in Section 3.1, based on the SXDH assumption. Together with the generic construction from Section 3, this gives the quadratic FE advertised in Section 1, Fig.2.

Overview of the partially-hiding inner-product FE. We now highlight the construction of our new partially-hiding public-key FE for inner products. Our starting point is the FE for inner products from [ALS16], described in Fig.7. It is not function-hiding since $\mathbf{y}$ is part of the decryption keys generated by KeyGen. As in

$$\mathsf{pk} = [\mathbf{a}], [\mathbf{U}\mathbf{a}] \qquad //[\mathbf{a}] \leftarrow_{\mathrm{R}} \mathbb{G}^2,\ \mathbf{U} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{d\times 2}$$

$$\mathsf{Enc}(\mathbf{x} \in \mathbb{Z}^n) = \begin{bmatrix} \mathbf{a}r \\ \mathbf{x} + \mathbf{U}\mathbf{a}r \end{bmatrix} \in \mathbb{G}^{n+2} \qquad //r \leftarrow_{\mathrm{R}} \mathbb{Z}_p$$

$$\mathsf{KeyGen}(\mathbf{y} \in \mathbb{Z}^n) = \left(\mathbf{y}, \begin{pmatrix} -\mathbf{U}^\top \mathbf{y} \\ \mathbf{y} \end{pmatrix}\right) \in \mathbb{Z}^n \times \mathbb{Z}_p^{n+2}$$

Fig. 7. FE for inner products, from [ALS16].

[Lin17, Section 6.3], we use the fact that the decryption computes the inner product of $\mathsf{Enc}(\mathbf{x})$ and $\mathsf{KeyGen}(\mathbf{y})$ to obtain $[\mathbf{x}^\top \mathbf{y}] \in \mathbb{G}$. Namely, we replace the vector $\mathbf{y}$ in each decryption key by an ALS encryption of $\mathbf{y}$, and $\mathbf{x}$ in each ciphertext is replaced by an ALS decryption key for $\mathbf{x}$ (see Fig.3). Function-Hiding (hiding $\mathbf{y}$) follows from the security of the inner ALS FE, whereas security (hiding $\mathbf{x}$) follows from the security of the outter ALS FE. Note that we use the fact that the KeyGen algorithm from [ALS16] FE can act on vectors in $\mathbb{G}_2$, and the multiplications can be computed using the pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ to recover the inner product of $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{G}_T$.

To make our scheme public-key, we need to publish a restricted secret key for the inner layer FE that lets $\mathsf{KeyGen}^{\mathsf{in}}(\mathbf{x})$ run on vectors $\mathbf{x}$ spanned by the matrix $\mathbf{M}$ described in Section 3 (recall that $\mathbf{M}$ is a full rank, n times m matrix, with $n > m$). If we denote the master secret key of the inner layer FE by $\mathsf{msk} := \mathbf{U} \in \mathbb{Z}_p^{d\times 2}$, the restricted key would simply be $\mathbf{U}^\top \mathbf{M}$.

We prove simulation security, which is necessary to be of use in our quadratic FE scheme, and which is stronger than the classical indistinguishability based security for FE. To do so, we use the simulation security of [ALS16], which was proven in [AGRW17,Wee17]. We obtain simulation security in the semi-adaptive setting, where the adversary sends its challenge before querying any secret keys, but after receiving the public key. This is the best we can hope for, since a straightforward adaptation of the results from [BSW11,AGVW13] show that simulation security is impossible in the adaptive setting (where the adversary can query secret keys before sending its challenge ciphertext).

[Lin17] also builds function-hiding FE from a two layered FE encryption, but uses [ABDP15] instead of [ALS16], and only obtains indistinguishability security, in the private key setting (see Fig.3).

Our partially-hiding, simulation-secure inner-product FE is described in Fig.8.

$\underline{\mathsf{Setup}(1^\lambda, \mathcal{PG}, [\mathbf{M}]_1 \in \mathbb{G}_1^{n\times m})}$:

$\mathbf{a}, \mathbf{b} \leftarrow_{\mathrm{R}} \mathsf{DDH}$, $(\mathsf{crs}, \mathsf{trap}) \leftarrow \mathsf{Gen}_{\mathsf{crs}}(\mathsf{par}, [\mathbf{a}]_1)$, $\mathbf{U} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(n+2)\times 2}$, $\mathbf{V} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{n\times 2}$

Return $\mathsf{pk} := \big([\mathbf{a}]_1, [\mathbf{U}\mathbf{a}]_1, [\mathbf{V}^\top \mathbf{M}]_1, [\mathbf{M}]_1, \mathsf{crs}\big)$, $\mathsf{msk} := ([\mathbf{b}]_2, \mathbf{V}, \mathbf{U})$

$\underline{\mathsf{Enc}(\mathbf{t} \in \mathbb{Z}^m, \tau)}$:

$r \leftarrow_{\mathrm{R}} \mathbb{Z}_p$, $\mathbf{c} := \mathbf{a}r$, $\pi \leftarrow \mathsf{Prove}(\mathsf{crs}, \tau, [\mathbf{c}]_1, r)$

Return $\mathsf{ct} := \left(\left[\begin{pmatrix} \mathbf{c} \\ \begin{pmatrix} -\mathbf{V}^\top \mathbf{M}\mathbf{t} \\ \mathbf{M}\mathbf{t} \end{pmatrix} + \mathbf{U}\mathbf{c} \end{pmatrix} \right]_1, \pi \right)$

$\underline{\mathsf{Enc}'(\mathsf{msk}, [\mathbf{x}]_1 \in \mathbb{G}_1^n, \tau)}$:

$r \leftarrow_{\mathrm{R}} \mathbb{Z}_p$, $\mathbf{c} := \mathbf{a}r$, $\pi \leftarrow \mathsf{Prove}(\mathsf{crs}, \tau, [\mathbf{c}]_1, r)$

Return $\mathsf{ct} := \left(\left[\begin{pmatrix} \mathbf{c} \\ \begin{pmatrix} -\mathbf{V}^\top \mathbf{x} \\ \mathbf{x} \end{pmatrix} + \mathbf{U}\mathbf{c} \end{pmatrix} \right]_1, \pi \right)$

$\underline{\mathsf{KeyGen}(\mathsf{msk}, \mathbf{y} \in \mathbb{Z}_p^n)}$:

$s \leftarrow_{\mathrm{R}} \mathbb{Z}_p$

Return $[\mathsf{sk}_\mathbf{y}]_2 := \begin{bmatrix} -\mathbf{U}^\top \begin{pmatrix} \mathbf{b}s \\ \mathbf{y} + \mathbf{V}\mathbf{b}s \end{pmatrix} \\ \begin{pmatrix} \mathbf{b}s \\ \mathbf{y} + \mathbf{V}\mathbf{b}s \end{pmatrix} \end{bmatrix}_2 \in \mathbb{G}_2^{n+4}$

$\underline{\mathsf{Dec}(\tau, [\mathsf{sk}_\mathbf{y}]_2, \mathsf{ct})}$:

Parse $\mathsf{ct} := \left(\begin{bmatrix} \mathbf{c} \\ \mathbf{c}' \end{bmatrix}_1, \pi \right)$, with $[\mathbf{c}]_1 \in \mathbb{G}_1^2$.

If $\mathsf{Ver}(\mathsf{crs}, \tau, \pi, [\mathbf{c}]_1) = 0$, then return $\perp$. Return $e\left(\begin{bmatrix} \mathbf{c} \\ \mathbf{c}' \end{bmatrix}_1^\top, [\mathsf{sk}_\mathbf{y}]_2 \right) \in \mathbb{G}_T$.

Fig. 8. simulation-secure, partially function-hiding inner-product FE. The components highlighted in gray are only present in the CCA secure scheme. Here, we use a QANIZK argument $\Pi := (\mathsf{Gen}_{\mathsf{par}}, \mathsf{Gen}_{\mathsf{crs}}, \mathsf{Prove}, \mathsf{Sim}, \mathsf{Ver})$, where $\mathsf{par} = \mathcal{PG}$, a pairing group $\mathcal{PG} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, P_1, P_2, e)$. Given $\rho := [\mathbf{a}]_1$, the language $\mathcal{L}_\rho$ is defined as $\mathcal{L}_\rho = \big\{[\mathbf{c}]_1 \in \mathbb{G}_1^2 : \exists r \in \mathbb{Z}_p \text{ s.t.} \mathbf{c} = \mathbf{a}r\big\}$.

Correctness. For all $\mathbf{t} \in \mathbb{Z}_p^m$ and $\mathbf{y} \in \mathbb{Z}_p^n$, we have:

$$\begin{pmatrix} \mathbf{a}r \\ \begin{pmatrix} -\mathbf{V}^\top \mathbf{M}\mathbf{t} \\ \mathbf{M}\mathbf{t} \end{pmatrix} + \mathbf{U}\mathbf{a}r \end{pmatrix}^\top \begin{pmatrix} -\mathbf{U}^\top \begin{pmatrix} \mathbf{b}s \\ \mathbf{y} + \mathbf{V}\mathbf{b}s \end{pmatrix} \\ \begin{pmatrix} \mathbf{b}s \\ \mathbf{y} + \mathbf{V}\mathbf{b}s \end{pmatrix} \end{pmatrix} = \begin{pmatrix} -\mathbf{V}^\top \mathbf{M}\mathbf{t} \\ \mathbf{M}\mathbf{t} \end{pmatrix}^\top \begin{pmatrix} \mathbf{b}s \\ \mathbf{y} + \mathbf{V}\mathbf{b}s \end{pmatrix}$$
$$= (\mathbf{M}\mathbf{t})^\top \mathbf{y}.$$

The first equality uses the correctness of the outer ALS encryption, while the second equality uses the correctness of the inner ALS encryption. We conclude using the completeness of the QANIZK argument.

Remark 1 (Large inputs). First, we observe that the encryption algorithm of our partially function-hiding inner product FE can take as input arbitrary vectors $\mathbf{x} \in \mathbb{Z}_p^n$, as opposed to $\mathbf{x} \in [0, B]^n$ for a polynomially bounded B. The decryption is in two step: first $\mathsf{Dec}_{\mathsf{large}}(\mathsf{Enc}(\mathbf{x}), \mathsf{KeyGen}(\mathsf{msk}, \mathbf{y}))$ for arbitrary vectors $\mathbf{x} \in \mathbb{Z}_p^n$ and $\mathbf{y} \in \mathbb{Z}_p^n$, recovers $[\mathbf{x}^\top \mathbf{y}]_T$. The second step solves the discrete logarithm to recover $\mathbf{x}^\top \mathbf{y}$. The second step is only efficient for polynomially bounded output, whereas the first step handles arbitrary large inputs.

The second observation we make is that for all $\mathbf{y} \in \mathbb{Z}_p^n$, $\mathsf{KeyGen}(\mathsf{msk}, \mathbf{y})$ outputs a vector or group elements $[\mathbf{d}]_2 \in \mathbb{G}_2^\ell$, for some dimension ℓ. The algorithm $\widetilde{\mathsf{KeyGen}}$ from the simulator of our scheme described in Fig.10, when given as input $\widetilde{\mathsf{msk}}$, $\widetilde{\mathbf{y}}$, and $\mathbf{x}^\top \mathbf{y}$, first computes $\mathbf{d} \in \mathbb{Z}_p^\ell$, and then returns $[\mathbf{d}]_2$ as the functional decryption key for $\mathbf{y}$. Moreover, it is a linear function in its input $\widetilde{\mathbf{y}}$ and $\mathbf{x}^\top \mathbf{y}$. Thus, we can run $\widetilde{\mathsf{KeyGen}}(\widetilde{\mathsf{msk}}, [\widetilde{\mathbf{y}}]_2, [\mathbf{x}^\top \mathbf{y}]_2)$ to get the functional decryption key $[\mathbf{d}]_2$. Otherwise stated, we achieved a slightly stronger simulation security than in Definition 7, since the simulator only requires to know the value $[\widetilde{\mathbf{y}}]_2$ and $\mathbf{x}^\top \mathbf{y}]_2$ in $\mathbb{G}_2$, as opposed to the values $\widetilde{\mathbf{y}}$ and $\mathbf{x}^\top \mathbf{y}$ over $\mathbb{Z}_p$.

Consequently, the quadratic FE from Section 3 that builds upon our partially function-hiding inner product FE inherits the same properties. Therefore, it can be use to encrypt random vectors $\mathbf{u} \in \mathbb{Z}_p^n$, and $\mathbf{v} \in \mathbb{Z}_p^m$. Each functional decryption key corresponds to a pair of indices $(i, j) \in [n] \times [m]$, and the decryption outputs $[u_i v_j]_T$. Simulation security ensures that the adversary view can be simulated from $[u_i v_j]_2$ only, which are pseudorandom by the DDH assumption in $\mathbb{G}_2$. This allows users to evaluate outputs of a PRG over $\mathbb{G}_T$, which is unachievable with an indistinguishability-based quadratic FE.

Theorem 2 (Security). *The inner-product FE described in Fig.8 is partially-hiding, CPA simulation-secure, assuming the SXDH assumption. Moreover, if the QANIZK argument Π is one-time simulation sound, then the FE is CCA simulation-secure. Namely, for any PPT adversary $\mathcal{A}$, there exist PPT adversaries $\mathcal{B}_1$ and $\mathcal{B}_2$ such that:*

$$\mathrm{Adv}^{\mathsf{CPA\text{-}PFH\text{-}SIM}}_{\mathsf{IPFE},\mathcal{A}}(\lambda) \leq \mathrm{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1,\mathcal{B}_1}(\lambda) + 2Q_{\mathsf{sk}} \cdot \mathrm{Adv}^{\mathsf{DDH}}_{\mathbb{G}_2,\mathcal{B}_2}(\lambda) + \frac{1 + Q_{\mathsf{sk}}}{p},$$

where Q_{sk} denotes the number of queries to $\mathcal{O}_{\mathsf{KeyGen}}$. Moreover, for any PPT adversary $\mathcal{A}'$, there exist PPT adversaries $\mathcal{B}'_1$, $\mathcal{B}'_2$ and $\mathcal{B}'_3$ such that:

$$\mathrm{Adv}^{\mathsf{CCA\text{-}PFH\text{-}SIM}}_{\mathsf{IPFE},\mathcal{A}}(\lambda) \leq \mathrm{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1,\mathcal{B}'_1}(\lambda) + 2(Q_{\mathsf{sk}} + Q_{\mathsf{Dec}}) \cdot \mathrm{Adv}^{\mathsf{DDH}}_{\mathbb{G}_2,\mathcal{B}'_2}(\lambda) + Q_{\mathsf{Dec}} \cdot \mathrm{Adv}^{\mathsf{OT}\text{-}\Pi}_{\mathcal{B}'_2}(\lambda) + \frac{1 + Q_{\mathsf{sk}} + Q_{\mathsf{Dec}}}{p},$$

where Q_{sk} denotes the number of queries to $\mathcal{O}_{\mathsf{KeyGen}}$, and Q_{Dec} denotes the number of queries to $\mathcal{O}_{\mathsf{Dec}}$.

Proof. Let $\mathcal{A}$ be a PPT adversary. We use a sequence of hybrid games Game_i for $i \in \{0, \ldots, 5\}$, defined in Fig.9, and we denote the advantage $\varepsilon_i := \Pr[1 \leftarrow \mathsf{Game}_i(\mathcal{A}, 1^\lambda)]$. We show that these games are computationally indistinguishable: $\mathsf{Game}_0 := \mathsf{Real}^{\mathsf{CCA\text{-}IPFE}}_{\mathcal{A}}(1^\lambda) \approx_c \mathsf{Game}_1 \approx_s \mathsf{Game}_2 \approx_c \mathsf{Game}_3 \equiv \mathsf{Ideal}^{\mathsf{CCA\text{-}IPFE}}_{\mathcal{A},\mathcal{S}}(1^\lambda)$, for the PPT simulator $\mathcal{S} := (\widetilde{\mathsf{Setup}}, \widetilde{\mathsf{Enc}}, \widetilde{\mathsf{KeyGen}}, \widetilde{\mathsf{Dec}})$ defined in Fig.10.

$\mathsf{Game}_0, \mathsf{Game}_1, \boxed{\mathsf{Game}_2, \mathsf{Game}_3, \mathsf{Game}_4, \mathsf{Game}_5, \mathsf{Game}_6}$:

$\mathbf{a}, \mathbf{b} \leftarrow_{\mathrm{R}} \mathsf{DDH}$, $(\mathsf{crs}, \mathsf{trap}) \leftarrow \mathsf{Gen}_{\mathsf{crs}}(\mathsf{par}, [\mathbf{a}]_1)$, $\boxed{\mathbf{a}^\perp \leftarrow_{\mathrm{R}} \mathbb{Z}_p^2 \setminus \{\mathbf{0}\} \text{ s.t. } \mathbf{a}^\top \mathbf{a}^\perp = 0}$, $\mathbf{U} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(n+2)\times 2}$, $\mathbf{V} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{n\times 3}$
$\mathsf{pk} := ([\mathbf{a}]_1, [\mathbf{U}\mathbf{a}]_1, [\mathbf{V}^\top \mathbf{M}]_1, [\mathbf{M}]_1, \mathsf{crs})$, $\mathsf{msk} := (\mathbf{U}, \mathbf{V}, \mathbf{M})$
$(\tau, \mathbf{x}) \leftarrow \mathcal{A}(\mathsf{pk})$
$\mathsf{ct}^\star \leftarrow \mathcal{O}_{\mathsf{Enc}}(\mathbf{x}, \tau)$
$\alpha \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KeyGen}}(\cdot), \mathcal{O}_{\mathsf{Dec}}(\cdot,\cdot,\cdot)}(\mathsf{ct}^\star)$

$\underline{\mathcal{O}_{\mathsf{Enc}}(\mathbf{x}, \tau)}$: //$\mathsf{Game}_0$, $\boxed{\mathsf{Game}_1, \boxed{\mathsf{Game}_2, \mathsf{Game}_3}, \mathsf{Game}_4, \mathsf{Game}_5, \mathsf{Game}_6}$

$r \leftarrow_{\mathrm{R}} \mathbb{Z}_p$, $\mathbf{c} := \mathbf{a}r$, $\boxed{\mathbf{c} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^2 \text{ s.t. } \mathbf{c}^\top \mathbf{a}^\perp = 1}$
$\pi \leftarrow \mathsf{Prove}(\mathsf{crs}, \tau, [\mathbf{c}]_1, r)$, $\boxed{\pi \leftarrow \mathsf{Sim}(\mathsf{crs}, \mathsf{trap}, \tau, [\mathbf{c}]_1)}$
$\mathsf{ct}_{\mathbf{x}} := \begin{pmatrix} \mathbf{c} \\ \begin{pmatrix} -\mathbf{V}^\top \mathbf{x} \\ \mathbf{x} \end{pmatrix} + \mathbf{U}\mathbf{c} \end{pmatrix}$, $\mathsf{ct}_{\mathbf{x}} := \begin{pmatrix} \mathbf{c} \\ \mathbf{U}\mathbf{c} \end{pmatrix}$
Return $\mathsf{ct}^\star := ([\mathsf{ct}_{\mathbf{x}}]_1, \pi)$

$\underline{\mathcal{O}_{\mathsf{KeyGen}}(\mathbf{y})}$: //$\mathsf{Game}_0, \mathsf{Game}_1, \mathsf{Game}_2, \mathsf{Game}_3, \boxed{\mathsf{Game}_4, \boxed{\mathsf{Game}_5, \mathsf{Game}_6}}$

$s \leftarrow_{\mathrm{R}} \mathbb{Z}_p$, $\widetilde{\mathbf{y}} := \mathbf{y}$, $\boxed{\widetilde{\mathbf{y}} := \mathbf{M}(\mathbf{M}^\top \mathbf{M})^{-1}(\mathbf{M}^\top \mathbf{y})}$
Return $\begin{bmatrix} -\mathbf{U}^\top \begin{pmatrix} \mathbf{b}s \\ \widetilde{\mathbf{y}} + \mathbf{V}\mathbf{b}s \end{pmatrix} - \boxed{(\mathbf{x}^\top \mathbf{y})\mathbf{a}^\perp} \\ \begin{pmatrix} \mathbf{b}s \\ \widetilde{\mathbf{y}} + \mathbf{V}\mathbf{b}s \end{pmatrix} \end{bmatrix}_2$

$\underline{\mathcal{O}_{\mathsf{Dec}}(\tau, \mathsf{ct}, \mathbf{y})}$: //$\mathsf{Game}_0, \mathsf{Game}_1, \mathsf{Game}_2, \boxed{\mathsf{Game}_3, \mathsf{Game}_4, \mathsf{Game}_5, \boxed{\mathsf{Game}_6}}$

If $\tau = \tau^\star$, return $\perp$.
Parse $\mathsf{ct} := \left(\begin{bmatrix} \mathbf{c} \\ \mathbf{c}' \end{bmatrix}_1, \pi \right)$ where $[\mathbf{c}]_1 \in \mathbb{G}_1^2$.
If $\mathsf{Ver}(\mathsf{crs}, \tau, \pi, [\mathbf{c}]_1) = 0$, return $\perp$. $\boxed{\text{If } [\mathbf{c}^\top \mathbf{a}^\perp]_1 \neq [0]_1, \text{ return } \perp.}$
$\widetilde{\mathbf{y}} := \mathbf{y}$, $\boxed{\widetilde{\mathbf{y}} := \mathbf{M}(\mathbf{M}^\top \mathbf{M})^{-1}(\mathbf{M}^\top \mathbf{y})}$
Compute $[\mathsf{sk}_{\mathbf{y}}]_2 := \begin{bmatrix} -\mathbf{U}^\top \begin{pmatrix} \mathbf{b}s \\ \widetilde{\mathbf{y}} + \mathbf{V}\mathbf{b}s \end{pmatrix} \\ \begin{pmatrix} \mathbf{b}s \\ \widetilde{\mathbf{y}} + \mathbf{V}\mathbf{b}s \end{pmatrix} \end{bmatrix}_2$, and return $e\left(\begin{bmatrix} \mathbf{c} \\ \mathbf{c}' \end{bmatrix}_1^\top, [\mathsf{sk}_{\mathbf{y}}]_2 \right)$.

Fig. 9. Games for the security proof of the inner-product FE from Fig.8. In each procedure, the components inside a solid (dotted, gray) frame are only present in the games marked by a solid (dotted, gray) frame.

Game_1*:* here, we use the algorithm Sim of Π to compute the proof in the challenge ciphertext, instead of using Prove. By perfect zero-knowledge of Π, this does not change the distribution of the game. Thus, we have

$$\varepsilon_0 = \varepsilon_1.$$

Game_2*:* here, we first switch the distribution of the vector $[\mathbf{c}]_1$ in the challenge ciphertext to uniformly random over $\mathbb{G}_1^2$, using the DDH assumption in $\mathbb{G}_1$. Namely, we build a PPT adversary $\mathcal{B}_1$ such that

$$\varepsilon_1 - \varepsilon_2 \leq \mathrm{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1,\mathcal{B}_1}(\lambda) + \frac{1}{p}.$$

Upon receiving a DDH challenge $([\mathbf{a}]_1, [\mathbf{c}]_1)$, $\mathcal{B}_1$ samples $\mathbf{U} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(n+2)\times 2}$, $\mathbf{V} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{n\times 3}$, $(\mathsf{crs}, \mathsf{trap}) \leftarrow \mathsf{Gen}_{\mathsf{crs}}(\mathsf{par}, [\mathbf{a}]_1)$, and simulates the view of $\mathcal{A}$ as described in Fig.9, using $[\mathbf{c}]_1$ as part of the challenge ciphertext. Finally, we use the fact that $\mathbf{c} \notin \mathsf{Span}(\mathbf{a})$ with probability $1 - \frac{1}{p}$ over the choice of random $\mathbf{c} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^2$. When $\mathbf{c} \notin \mathsf{Span}(\mathbf{a})$, we can choose $\mathbf{a}^\perp \leftarrow_{\mathrm{R}} \mathbb{Z}_p^2 \setminus \{\mathbf{0}\}$ such that $\mathbf{a}^\top \mathbf{a}^\perp = 0$ and $\mathbf{c}^\top \mathbf{a}^\perp = 1$.

Game_3*:* here, the oracle $\mathcal{O}_{\mathsf{Dec}}(\tau, \mathsf{ct}, \mathbf{y})$ additionally checks that the ciphertext $\mathsf{ct} := \left(\begin{bmatrix}\mathbf{c}\\ \mathbf{c}'\end{bmatrix}_1, \pi\right)$, with $[\mathbf{c}]_1 \in \mathbb{G}_1^2$, is such that $[\mathbf{c}^\top \mathbf{a}^\perp]_1 = [\mathbf{0}]_1$. If the additional check fails, but the queries passes the verification $\mathsf{Ver}(\mathsf{crs}, \tau, [\mathbf{c}]_1, \pi)$, then the tuple $([\mathbf{c}]_1, \pi, \tau)$ violates the one-time soundness of Π. Using a hybrid argument over all queries to $\mathcal{O}_{\mathsf{Dec}}$, we obtain a PPT adversary $\mathcal{B}_2$ such that:

$$\varepsilon_2 - \varepsilon_3 \leq Q_{\mathsf{Dec}} \cdot \mathrm{Adv}^{\mathsf{OT}\text{-}\Pi}_{\mathcal{B}_2}(\lambda).$$

Game_4*:* is identically distributed to Game_5, namely, we have

$$\varepsilon_3 = \varepsilon_4.$$

We first show that these advantages are equal for any adversary choosing the challenge $\mathbf{x}$ independently of the public key pk. Then, we use the fact that any adversary can be turned into an adversary that chooses $\mathbf{x}$ independently of pk, by simply guessing the challenge $\mathbf{x}$ uniformly at random. This incurs an exponential security loss, namely, this proves that $(\varepsilon_3 - \varepsilon_4) \cdot p^n = 0$, which implies $\varepsilon_3 - \varepsilon_4 = 0$. It remains to prove the statement for adversaries choosing $[\mathbf{x}]_1$ independently of pk. In this selective setting, we use the fact that $\mathbf{U} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(n+2)\times 2}$ is identically distributed to $\mathbf{U} - \begin{pmatrix}-\mathbf{V}^\top \mathbf{x}\\ \mathbf{x}\end{pmatrix} (\mathbf{a}^\perp)^\top$. Note that the former distribution corresponds to Game_3, while the latter corresponds to Game_4, using the following facts:

- in pk: $\left(\mathbf{U} - \begin{pmatrix}-\mathbf{V}^\top \mathbf{x}\\ \mathbf{x}\end{pmatrix} (\mathbf{a}^\perp)^\top\right) \mathbf{a} = \mathbf{U}\mathbf{a}$
- in $\mathcal{O}_{\mathsf{KeyGen}}(\mathbf{y})$: $\left(\mathbf{U} - \begin{pmatrix}-\mathbf{V}^\top \mathbf{x}\\ \mathbf{x}\end{pmatrix} (\mathbf{a}^\perp)^\top\right) \begin{pmatrix}\mathbf{b}s\\ \mathbf{y} + \mathbf{V}\mathbf{b}s\end{pmatrix} = \mathbf{U} \begin{pmatrix}\mathbf{b}s\\ \mathbf{y} + \mathbf{V}\mathbf{b}s\end{pmatrix} - (\mathbf{x}^\top \mathbf{y})\mathbf{a}^\perp.$
- in $\mathcal{O}_{\mathsf{Dec}}(\mathsf{ct}, \mathbf{y})$: writing $\mathsf{ct} := \left(\begin{bmatrix}\mathbf{c}\\ \mathbf{c}'\end{bmatrix}_1, \pi\right)$ with $[\mathbf{c}]_1 \in \mathbb{G}_1^2$, we know that if $[\mathbf{c}^\top \mathbf{a}^\perp]_1 \neq [0]_1$, then $\mathcal{O}_{\mathsf{Dec}}(\mathsf{ct}, \mathbf{y})$ returns $\perp$, by definition of the games. When $[\mathbf{c}^\top \mathbf{a}^\perp]_1 = [0]_1$, $\mathsf{Dec}([\mathsf{sk}_{\mathbf{y}}]_2, \mathsf{ct})$ returns

$$e\left(\begin{bmatrix}\mathbf{c}\\ \mathbf{c}'\end{bmatrix}_1^\top, \left[\mathbf{U} \begin{pmatrix}\mathbf{b}s\\ \mathbf{y} + \mathbf{V}\mathbf{b}s\end{pmatrix} - (\mathbf{x}^\top \mathbf{y})\mathbf{a}^\perp\right]_2\right) = e\left(\begin{bmatrix}\mathbf{c}\\ \mathbf{c}'\end{bmatrix}_1^\top, \left[\mathbf{U} \begin{pmatrix}\mathbf{b}s\\ \mathbf{y} + \mathbf{V}\mathbf{b}s\end{pmatrix}\right]_2\right),$$

since $\mathbf{c}^\top \mathbf{a}^\perp = 0$.

– in $\mathcal{O}_{\mathsf{Enc}}(\mathbf{x})$: $\mathsf{ct}_{\mathbf{x}} := \left(\begin{pmatrix} -\mathbf{V}^\top \mathbf{x} \\ \mathbf{x} \end{pmatrix} + \left(\mathbf{U} - \begin{pmatrix} -\mathbf{V}^\top \mathbf{x} \\ \mathbf{x} \end{pmatrix} (\mathbf{a}^\perp)^\top \right) \mathbf{c} \right) = \begin{pmatrix} \mathbf{c} \\ \mathbf{U}\mathbf{c} \end{pmatrix}$,

using the fact that $\mathbf{c}^\top \mathbf{a}^\perp = 1$ and $\mathbf{a}^\top \mathbf{a}^\perp = 0$.

Such guessing followed by a perfect satistical argument to obtain adaptive security has already been used in [ALS16], or even in [Wee14] in the context of attribute-based encryption. See also [AGRW17, Theorem 6] for an explicit exposition of this technique.

Game_5*:* we prove in Lemma 1 that there exists a PPT adversary $\mathcal{B}_4$ such that

$$\varepsilon_4 - \varepsilon_5 \leq 2Q_{\mathsf{sk}} \cdot \mathrm{Adv}^{\mathsf{DDH}}_{\mathbb{G}_2, \mathcal{B}_4}(\lambda) + \frac{Q_{\mathsf{sk}}}{p},$$

where Q_{sk} denotes the number of queries to $\mathcal{O}_{\mathsf{KeyGen}}$.

Game_6*:* we prove in Lemma 2 that there exists a PPT adversary $\mathcal{B}_5$ such that

$$\varepsilon_5 - \varepsilon_6 \leq 2Q_{\mathsf{Dec}} \cdot \mathrm{Adv}^{\mathsf{DDH}}_{\mathbb{G}_2, \mathcal{B}_5}(\lambda) + \frac{Q_{\mathsf{Dec}}}{p},$$

where Q_{Dec} denotes the number of queries to $\mathcal{O}_{\mathsf{Dec}}$. It is clear from the description of the simulator $\mathcal{S} := (\widetilde{\mathsf{Setup}}, \widetilde{\mathsf{Enc}}, \widetilde{\mathsf{KeyGen}}, \widetilde{\mathsf{Dec}})$ from Fig.10 that Game_6 is identically distributed to $\mathsf{Ideal}^{\mathsf{CCA\text{-}IPFE}}_{\mathcal{A}}(1^\lambda)$. □

Lemma 1 (Transition from Game_4 to Game_5). *There exists a PPT adversary $\mathcal{B}_4$ such that*

$$\varepsilon_4 - \varepsilon_5 \leq 2Q_{\mathsf{sk}} \cdot \mathrm{Adv}^{\mathsf{DDH}}_{\mathbb{G}_2, \mathcal{B}_4}(\lambda) + \frac{Q_{\mathsf{sk}}}{p},$$

where Q_{sk} denotes the number of queries to $\mathcal{O}_{\mathsf{KeyGen}}$.

Proof (of Lemma 1). We use a sequence of hybrid games G_{i} for $i \in \{1, \ldots, Q_{\mathsf{sk}} + 1\}$ and $\mathsf{G}_{\mathsf{i.1}}, \mathsf{G}_{\mathsf{i.2}}$ for $i \in \{1, \ldots, Q_{\mathsf{sk}}\}$ described in Fig.11. Note that G_1 is Game_4 and $\mathsf{G}_{Q_{\mathsf{sk}}+1}$ is Game_5. Thus, it suffices to show that for all $i \in \{1, \ldots, Q_{\mathsf{sk}}\}$, $\mathsf{G}_i \approx_c \mathsf{G}_{i+1}$, which we prove using the hybrids $\mathsf{G}_{\mathsf{i.1}}$ and $\mathsf{G}_{\mathsf{i.2}}$. For any index x, we denote by ε_x the advantage of adversary $\mathcal{A}$ in game G_{x}, that is, $\varepsilon_x := \Pr[1 \leftarrow_{\mathrm{R}} \mathsf{G}_{\mathsf{x}}(1^\lambda, \mathcal{A})]$.

$\mathsf{G}_{\mathsf{i.1}}$*:* is as G_{i}, except that we switch the distribution of the vector $\mathbf{d}$ used in the i'th query to $\mathcal{O}_{\mathsf{KeyGen}}$, to uniformly random over $\mathbb{Z}_p^2$, using the DDH assumption in $\mathbb{G}_2$. Namely, we build a PPT adversary $\mathcal{B}_{i.1}$ such that

$$\varepsilon_i - \varepsilon_{i.1} \leq \mathrm{Adv}^{\mathsf{DDH}}_{\mathbb{G}_2, \mathcal{B}_{i.1}}(\lambda).$$

Upon receiving its DDH challenge, $\mathcal{B}_{i.1}$ samples $\mathbf{a} \leftarrow_{\mathrm{R}} \mathsf{DDH}$, $\mathbf{a}^\perp \leftarrow_{\mathrm{R}} \mathbb{Z}_p^2 \backslash \{\mathbf{0}\}$ such that $\mathbf{a}^\top \mathbf{a}^\perp \neq 0$, $(\mathsf{crs}, \mathsf{trap}) \leftarrow \mathsf{Gen}_{\mathsf{crs}}(\mathsf{par}, [\mathbf{a}]_1)$, $\mathbf{U} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(n+2)\times 2}$, $\mathbf{V} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{n\times 3}$, and simulates $\mathcal{A}$'s view as described in Fig.11, embedding its DDH challenge in the i'th query to $\mathcal{O}_{\mathsf{KeyGen}}$.

$\mathsf{G}_{\mathsf{i.2}}$*:* is statistically close to $\mathsf{G}_{\mathsf{i.1}}$. We show this using fact that the following are statistically close (within statistical distance $1/p$):

$$\mathbf{d} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^2 \quad \text{and} \quad \mathbf{d} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^2 \setminus \mathsf{Span}(\mathbf{b}).$$

Assuming $\mathbf{d} \notin \mathsf{Span}(\mathbf{b})$, there exists a vector $\mathbf{b}^\perp \in \mathbb{Z}_p^2 \setminus \{\mathbf{0}\}$ such that $\mathbf{b}^\top \mathbf{b}^\perp = 0$, and $\mathbf{d}^\top \mathbf{b}^\perp = 1$. Note that $(\mathbf{b}|\mathbf{b}^\perp)$ forms a basis of $\mathbb{Z}_p^2$. We also use the basis $(\mathbf{M}|\mathbf{M}^\perp)$ for $\mathbb{Z}_p^{n\times n}$, where $\mathbf{M}^\perp$ is a full rank matrix such that $\mathbf{M}^\top \mathbf{M}^\perp = \mathbf{0}$ (recall that $\mathbf{M}$ itself is full rank). We can write $\mathbf{V} = \mathbf{M}\mathbf{v}_{00}\mathbf{b}^\top + \mathbf{M}\mathbf{v}_{01}(\mathbf{b}^\perp)^\top + \mathbf{M}^\perp \mathbf{v}_{10}\mathbf{b}^\top + \mathbf{M}^\perp \mathbf{v}_{11}(\mathbf{b}^\perp)^\top$, where $\mathbf{v}_{00}, \mathbf{v}_{01} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^m$, and $\mathbf{v}_{10}, \mathbf{v}_{11} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{n-m}$.

$\widetilde{\mathsf{Setup}}(1^\lambda, \mathcal{PG}, [\mathbf{M}]_1 \in \mathbb{G}_1^{n\times m})$:

$\mathbf{a} \leftarrow_{\mathrm{R}} \mathsf{DDH}$, $(\mathsf{crs}, \mathsf{trap}) \leftarrow \mathsf{Gen}_{\mathsf{crs}}(\mathsf{par}, [\mathbf{a}]_1)$, $\mathbf{a}^\perp \leftarrow_{\mathrm{R}} \mathbb{Z}_p^2 \setminus \{\mathbf{0}\}$ such that $\mathbf{a}^\top \mathbf{a}^\perp = 0$, $\mathbf{U} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(n+2)\times 2}$, $\mathbf{V} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{n\times 3}$

$\widetilde{\mathsf{pk}} := ([\mathbf{a}]_1, [\mathbf{U}\mathbf{a}]_1, [\mathbf{V}^\top \mathbf{M}]_1, [\mathbf{M}]_1, \mathsf{crs})$, $\widetilde{\mathsf{msk}} := (\mathbf{U}, \mathbf{V}, \mathbf{a}^\perp, \mathsf{trap})$

Return $(\widetilde{\mathsf{pk}}, \widetilde{\mathsf{msk}})$

$\widetilde{\mathsf{Enc}}(\widetilde{\mathsf{msk}}, \tau)$:

$\mathbf{c} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^2$ such that $\mathbf{c}^\top \mathbf{a}^\perp = 1$, $\pi \leftarrow \mathsf{Sim}(\mathsf{crs}, \mathsf{trap}, \tau, [\mathbf{c}]_1)$

Return $\left(\begin{bmatrix} \mathbf{c} \\ \mathbf{U}\mathbf{c} \end{bmatrix}_1, \pi \right)$.

$\widetilde{\mathsf{KeyGen}}(\widetilde{\mathsf{msk}}, \mathbf{x}^\top \mathbf{y}, \widetilde{\mathbf{y}})$:

$s \leftarrow_{\mathrm{R}} \mathbb{Z}_p$, return $\begin{bmatrix} -\mathbf{U}^\top \begin{pmatrix} \mathbf{b}s \\ \widetilde{\mathbf{y}} + \mathbf{V}\mathbf{b}s \end{pmatrix} - (\mathbf{x}^\top \mathbf{y})\mathbf{a}^\perp \\ \begin{pmatrix} \mathbf{b}s \\ \widetilde{\mathbf{y}} + \mathbf{V}\mathbf{b}s \end{pmatrix} \end{bmatrix}_2$

$\widetilde{\mathsf{Dec}}(\widetilde{\mathsf{msk}}, \tau, \mathsf{ct}, \widetilde{\mathbf{y}})$:

If $\tau = \tau^\star$, return $\perp$.

Parse $\mathsf{ct} := \left(\begin{bmatrix} \mathbf{c} \\ \mathbf{c}' \end{bmatrix}_1, \pi \right)$, where $[\mathbf{c}]_1 \in \mathbb{G}_1^2$.

If $[\mathbf{c}^\top \mathbf{a}^\perp]_1 \neq [0]_1$, return $\perp$.

$s \leftarrow_{\mathrm{R}} \mathbb{Z}_p$, $[\mathsf{sk}_{\mathbf{y}}]_2 := \begin{bmatrix} -\mathbf{U}^\top \begin{pmatrix} \mathbf{b}s \\ \widetilde{\mathbf{y}} + \mathbf{V}\mathbf{b}s \end{pmatrix} \\ \begin{pmatrix} \mathbf{b}s \\ \widetilde{\mathbf{y}} + \mathbf{V}\mathbf{b}s \end{pmatrix} \end{bmatrix}_2$

Return $e\left(\begin{bmatrix} \mathbf{c} \\ \mathbf{c}' \end{bmatrix}_1^\top, [\mathsf{sk}_{\mathbf{y}}]_2 \right)$.

Fig. 10. PPT simulator for the security proof of the partially-hiding simulation-secure inner-product FE from Fig.8.

Now, we show that the adversary view is independent of $\mathbf{v}_{11}$ up to its i'th query to $\mathcal{O}_{\mathsf{KeyGen}}$. This is because: $\mathbf{V}^\top \mathbf{M} = \left(\mathbf{M}\mathbf{v}_{00}\mathbf{b}^\top + \mathbf{M}\mathbf{v}_{01}(\mathbf{b}^\perp)^\top + \mathbf{M}^\perp \mathbf{v}_{10}\mathbf{b}^\top + \mathbf{M}^\perp \mathbf{v}_{11}(\mathbf{b}^\perp)^\top\right)^\top \mathbf{M} = \left(\mathbf{M}\mathbf{v}_{00}\mathbf{b}^\top + \mathbf{M}\mathbf{v}_{01}(\mathbf{b}^\perp)^\top\right)^\top \mathbf{M}$, so $\mathbf{v}_{11}$ does not appear in pk, and: $\mathbf{V}\mathbf{b} = \left(\mathbf{M}\mathbf{v}_{00}\mathbf{b}^\top + \mathbf{M}\mathbf{v}_{01}(\mathbf{b}^\perp)^\top + \mathbf{M}^\perp \mathbf{v}_{10}\mathbf{b}^\top + \mathbf{M}^\perp \mathbf{v}_{11}(\mathbf{b}^\perp)^\top\right)\mathbf{b} = \left(\mathbf{M}\mathbf{v}_{00}\mathbf{b}^\top + \mathbf{M}^\perp \mathbf{v}_{10}\mathbf{b}^\top\right)\mathbf{b}$, so the output of the $i+1$'st queries to $\mathcal{O}_{\mathsf{KeyGen}}$, and the output to the queries to $\mathcal{O}_{\mathsf{Dec}}$, are independent of $\mathbf{v}_{11}$.

Thus, we can use the entropy of $\mathbf{v}_{11}$ to partially hide the vector $\mathbf{y}$ associated to the i'th query to $\mathcal{O}_{\mathsf{KeyGen}}$. Namely, writing $\mathbf{y} := \mathbf{M}\mathbf{y}_0 + \mathbf{M}^\perp \mathbf{y}_1$, where $\mathbf{y}_0 \in \mathbb{Z}_p^m$, and $\mathbf{y}_1 \in \mathbb{Z}_p^{n-m}$, we show that only $\mathbf{M}\mathbf{y}_0 = \mathbf{M}(\mathbf{M}^\top \mathbf{M})^{-1}\mathbf{M}^\top \mathbf{y}$ leaks from $\mathsf{sk}_{\mathbf{y}}$. We prove so using the fact for all $\mathbf{y}_1 \in \mathbb{Z}_p^{n-m}$, the following are identically distributed:

$$\mathbf{v}_{11} \quad \text{and} \quad \mathbf{v}_{11} - \mathbf{y}_1,$$

where $\mathbf{v}_{11} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{n-m}$ (independent of $\mathbf{y}_1$). The leftmost distribution corresponds to $\mathsf{G}_{\mathsf{i}.1}$, whereas the rightmost distribution corresponds to $\mathsf{G}_{\mathsf{i}.2}$ (assuming $\mathbf{d} \notin \mathsf{Span}(\mathbf{b})$), since this change of variable appears in the

i'th decryption key as (highlighted in gray):

$$\mathsf{sk}_{\mathbf{y}} := \begin{pmatrix} -\mathbf{U}^\top \begin{pmatrix} \mathbf{d} \\ \mathbf{y}\,\boxed{-\mathbf{M}^\perp \mathbf{y}_1 (\mathbf{b}^\perp)^\top \mathbf{d}} + \mathbf{V}\mathbf{d} \end{pmatrix} - (\mathbf{x}^\top \mathbf{y})\mathbf{a}^\perp \\ \begin{pmatrix} \mathbf{d} \\ \mathbf{y}\,\boxed{-\mathbf{M}^\perp \mathbf{y}_1 (\mathbf{b}^\perp)^\top \mathbf{d}} + \mathbf{V}\mathbf{d} \end{pmatrix} \end{pmatrix}.$$

We conclude using the facts that $(\mathbf{b}^\perp)^\top \mathbf{d} = 1$, and $\mathbf{y} - \mathbf{M}^\perp \mathbf{y}_1 = \mathbf{M}\mathbf{y}_0 = \mathbf{M}(\mathbf{M}^\top \mathbf{M})^{-1}\mathbf{M}\mathbf{y}$.

That proves:

$$\varepsilon_{i.1} - \varepsilon_{i.2} \leq \frac{1}{p}.$$

$\mathsf{G}_{\mathsf{i+1}}$*:* is the same as $\mathsf{G}_{\mathsf{i.2}}$, except the distribution of the vector $\mathbf{d}$ used in the i'th decryption key is switched back to $\mathbf{d} = \mathbf{b}s$ for $s \leftarrow_{\mathrm{R}} \mathbb{Z}_p$, using the DDH assumption in $\mathbb{G}_2$. This is the reverse transition than from G_{i} to $\mathsf{G}_{\mathsf{i.1}}$. Thus, we have a PPT adversary $\mathcal{B}_{i.2}$ such that

$$\varepsilon_{i.2} - \varepsilon_{i+1} \leq \mathrm{Adv}^{\mathsf{DDH}}_{\mathbb{G}_2, \mathcal{B}_{i.2}}(\lambda).$$

Summing up for all $i \in \{1, \ldots, Q_{\mathsf{sk}}\}$, we obtain the lemma. □

Lemma 2 (Transition from Game_5 to Game_6). *There exists a PPT adversary $\mathcal{B}_5$ such that*

$$\varepsilon_5 - \varepsilon_6 \leq 2Q_{\mathsf{Dec}} \cdot \mathrm{Adv}^{\mathsf{DDH}}_{\mathbb{G}_2, \mathcal{B}_5}(\lambda) + \frac{Q_{\mathsf{Dec}}}{p},$$

where Q_{Dec} denotes the number of queries to $\mathcal{O}_{\mathsf{Dec}}$.

Proof (of Lemma 2). We use a sequence of hybrid games H_{i} for $i \in \{1, \ldots, Q_{\mathsf{Dec}} + 1\}$ and $\mathsf{H}_{\mathsf{i.1}}, \mathsf{H}_{\mathsf{i.2}}$ for $i \in \{1, \ldots, Q_{\mathsf{Dec}}\}$ described in Fig.12. Note that H_1 is Game_5 and H_{Q+1} is Game_6. We show that for all $i \in \{1, \ldots, Q_{\mathsf{Dec}}\}$, $\mathsf{H}_i \approx_c \mathsf{H}_{i+1}$, which we prove using the hybrids $\mathsf{H}_{\mathsf{i.1}}$ and $\mathsf{H}_{\mathsf{i.2}}$. We prove that these hybrid games are indistinguishable in the same way as for Lemma 1. We defer to the proof of the latter for further details. □

$\underline{\mathsf{G}_{\mathsf{i}}, \boxed{\mathsf{G}_{\mathsf{i}.1}, \boxed{\mathsf{G}_{\mathsf{i}.2}}}:}$

$\mathbf{a}, \mathbf{b} \leftarrow_{\mathrm{R}} \mathsf{DDH}$, $\mathbf{a}^{\perp} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^2 \setminus \{\mathbf{0}\}$ s.t. $\mathbf{a}^{\top}\mathbf{a}^{\perp} = 0$, $(\mathsf{crs}, \mathsf{trap}) \leftarrow \mathsf{Gen}_{\mathsf{crs}}(\mathsf{par}, [\mathbf{a}]_1)$, $\mathbf{U} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(n+2)\times 2}$, $\mathbf{V} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{n\times 3}$

$\mathsf{pk} := ([\mathbf{a}]_1, [\mathbf{U}\mathbf{a}]_1, [\mathbf{V}^{\top}\mathbf{M}]_1, [\mathbf{M}]_1, \mathsf{crs})$, $\mathsf{msk} := (\mathbf{U}, \mathbf{V}, \mathbf{a}^{\perp}, \mathbf{M}, \mathsf{trap})$

$(\tau^{\star}, \mathbf{x}) \leftarrow \mathcal{A}(\mathsf{pk})$

$\mathsf{ct}^{\star} \leftarrow \mathcal{O}_{\mathsf{Enc}}(\mathbf{x}, \tau^{\star})$

$\alpha \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KeyGen}}(\cdot), \mathcal{O}_{\mathsf{Dec}}(\cdot,\cdot,\cdot)}(\mathsf{ct}^{\star})$

$\underline{\mathcal{O}_{\mathsf{Enc}}(\tau^{\star}, \mathbf{x}):}$

$\mathbf{c} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^2$ s.t. $\mathbf{c}^{\top}\mathbf{a}^{\perp} = 1$, $\pi \leftarrow \mathsf{Sim}(\mathsf{crs}, \mathsf{trap}, \tau^{\star}, [\mathbf{c}]_1)$

Return $\mathsf{ct}^{\star} := \left(\begin{bmatrix} \mathbf{c} \\ \mathbf{U}\mathbf{c} \end{bmatrix}_1, \pi \right)$

$\underline{\mathcal{O}_{\mathsf{KeyGen}}(\mathbf{y}):}$

On the ρ'th query

-for $\rho < i$: $\widetilde{\mathbf{y}} := \mathbf{M}(\mathbf{M}^{\top}\mathbf{M})^{-1}(\mathbf{M}^{\top}\mathbf{y})$, $s \leftarrow_{\mathrm{R}} \mathbb{Z}_p$, $\mathbf{d} := \mathbf{b}s$

-for $\rho = i$: $\widetilde{\mathbf{y}} := \mathbf{y}$, $s \leftarrow_{\mathrm{R}} \mathbb{Z}_p$, $\mathbf{d} := \mathbf{b}s$, $\boxed{\mathbf{d} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^2}$, $\boxed{\widetilde{\mathbf{y}} := \mathbf{M}(\mathbf{M}^{\top}\mathbf{M})^{-1}(\mathbf{M}^{\top}\mathbf{y})}$

-for $\rho > i$: $\widetilde{\mathbf{y}} := \mathbf{y}$, $s \leftarrow_{\mathrm{R}} \mathbb{Z}_p$, $\mathbf{d} := \mathbf{b}s$

$$\mathsf{sk}_{\mathbf{y}} := \begin{pmatrix} -\mathbf{U}^{\top}\begin{pmatrix} \mathbf{d} \\ \widetilde{\mathbf{y}} + \mathbf{V}\mathbf{d} \end{pmatrix} - (\mathbf{x}^{\top}\mathbf{y})\mathbf{a}^{\perp} \\ \begin{pmatrix} \mathbf{d} \\ \widetilde{\mathbf{y}} + \mathbf{V}\mathbf{d} \end{pmatrix} \end{pmatrix}$$

Return $[\mathsf{sk}_{\mathbf{y}}]_2$

$\underline{\mathcal{O}_{\mathsf{Dec}}(\tau, \mathsf{ct}, \mathbf{y}):}$

If $\tau = \tau^{\star}$, return $\perp$.

Parse $\mathsf{ct} := \left(\begin{bmatrix} \mathbf{c} \\ \mathbf{c}' \end{bmatrix}_1, \pi \right)$ with $[\mathbf{c}]_1 \in \mathbb{G}_1^2$. If $[\mathbf{c}^{\top}\mathbf{a}^{\perp}]_1 = [0]_1$, return $\perp$.

$$s \leftarrow_{\mathrm{R}} \mathbb{Z}_p,\ [\mathsf{sk}_{\mathbf{y}}]_2 := \begin{bmatrix} -\mathbf{U}^{\top}\begin{pmatrix} \mathbf{b}s \\ \mathbf{y} + \mathbf{V}\mathbf{d} \end{pmatrix} \\ \begin{pmatrix} \mathbf{d} \\ \mathbf{y} + \mathbf{V}\mathbf{b}s \end{pmatrix} \end{bmatrix}_2$$

Return $\mathsf{Dec}(\tau, [\mathsf{sk}_{\mathbf{y}}]_2, \mathsf{ct})$

Fig. 11. Games G_{i} for $i \in \{1, \ldots, Q_{\mathsf{sk}} + 1\}$, $\mathsf{G}_{\mathsf{i}.1}, \mathsf{G}_{\mathsf{i}.2}$ for $i \in \{1, \ldots, Q_{\mathsf{sk}}\}$ for the proof of Lemma 1. In each procedure, the components inside a solid (dotted) frame are only present in the games marked by a solid (dotted) frame.

H_i, $\boxed{\mathsf{H}_{i.1}, \boxed{\mathsf{H}_{i.2}}}$:

$\mathbf{a}, \mathbf{b} \leftarrow_{\mathrm{R}} \mathsf{DDH}$, $\mathbf{a}^\perp \leftarrow_{\mathrm{R}} \mathbb{Z}_p^2 \setminus \{\mathbf{0}\}$ s.t. $\mathbf{a}^\top \mathbf{a}^\perp = 0$, $(\mathsf{crs}, \mathsf{trap}) \leftarrow \mathsf{Gen}_{\mathsf{crs}}(\mathsf{par}, [\mathbf{a}]_1)$, $\mathbf{U} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(n+2)\times 2}$, $\mathbf{V} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{n\times 3}$
$\mathsf{pk} := ([\mathbf{a}]_1, [\mathbf{U}\mathbf{a}]_1, [\mathbf{V}^\top \mathbf{M}]_1, [\mathbf{M}]_1, \mathsf{crs})$, $\mathsf{msk} := (\mathbf{U}, \mathbf{V}, \mathbf{a}^\perp, \mathbf{M}, \mathsf{trap})$
$(\tau^\star, \mathbf{x}) \leftarrow \mathcal{A}(\mathsf{pk})$
$\mathsf{ct}^\star \leftarrow \mathcal{O}_{\mathsf{Enc}}(\mathbf{x}, \tau^\star)$
$\alpha \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KeyGen}}(\cdot), \mathcal{O}_{\mathsf{Dec}}(\cdot,\cdot,\cdot)}(\mathsf{ct}^\star)$

$\underline{\mathcal{O}_{\mathsf{Enc}}(\tau^\star, \mathbf{x})}$:
$\mathbf{c} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^2$ s.t. $\mathbf{c}^\top \mathbf{a}^\perp = 1$, $\pi \leftarrow \mathsf{Sim}(\mathsf{crs}, \mathsf{trap}, \tau^\star, [\mathbf{c}]_1)$
Return $\mathsf{ct}^\star := \left(\begin{bmatrix} \mathbf{c} \\ \mathbf{U}\mathbf{c} \end{bmatrix}_1, \pi \right)$

$\underline{\mathcal{O}_{\mathsf{KeyGen}}(\mathbf{y})}$:
$\widetilde{\mathbf{y}} := \mathbf{M}(\mathbf{M}^\top \mathbf{M})^{-1}(\mathbf{M}^\top \mathbf{y})$, $s \leftarrow_{\mathrm{R}} \mathbb{Z}_p$ $\mathsf{sk}_{\mathbf{y}} := \begin{pmatrix} -\mathbf{U}^\top \begin{pmatrix} \mathbf{d} \\ \widetilde{\mathbf{y}} + \mathbf{V}\mathbf{b}s \end{pmatrix} - (\mathbf{x}^\top \mathbf{y})\mathbf{a}^\perp \\ \begin{pmatrix} \mathbf{d} \\ \widetilde{\mathbf{y}} + \mathbf{V}\mathbf{b}s \end{pmatrix} \end{pmatrix}$
Return $[\mathsf{sk}_{\mathbf{y}}]_2$

$\underline{\mathcal{O}_{\mathsf{Dec}}(\tau, \mathsf{ct}, \mathbf{y})}$:
If $\tau = \tau^\star$, return $\perp$.
Parse $\mathsf{ct} := \left(\begin{bmatrix} \mathbf{c} \\ \mathbf{c}' \end{bmatrix}_1, \pi \right)$ with $[\mathbf{c}]_1 \in \mathbb{G}_1^2$. If $[\mathbf{c}^\top \mathbf{a}^\perp]_1 = [0]_1$, return $\perp$.
On the ρ'th query
-for $\rho < i$: $\widetilde{\mathbf{y}} := \mathbf{M}(\mathbf{M}^\top \mathbf{M})^{-1}(\mathbf{M}^\top \mathbf{y})$, $s \leftarrow_{\mathrm{R}} \mathbb{Z}_p$, $\mathbf{d} := \mathbf{b}s$
-for $\rho = i$: $\widetilde{\mathbf{y}} := \mathbf{y}$, $s \leftarrow_{\mathrm{R}} \mathbb{Z}_p$, $\mathbf{d} := \mathbf{b}s$, $\boxed{\mathbf{d} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^2}$, $\boxed{\widetilde{\mathbf{y}} := \mathbf{M}(\mathbf{M}^\top \mathbf{M})^{-1}(\mathbf{M}^\top \mathbf{y})}$
-for $\rho > i$: $\widetilde{\mathbf{y}} := \mathbf{y}$, $s \leftarrow_{\mathrm{R}} \mathbb{Z}_p$, $\mathbf{d} := \mathbf{b}s$
$[\mathsf{sk}_{\mathbf{y}}]_2 := \begin{bmatrix} -\mathbf{U}^\top \begin{pmatrix} \mathbf{d} \\ \widetilde{\mathbf{y}} + \mathbf{V}\mathbf{d} \end{pmatrix} \\ \begin{pmatrix} \mathbf{d} \\ \widetilde{\mathbf{y}} + \mathbf{V}\mathbf{d} \end{pmatrix} \end{bmatrix}_2$
Return $\mathsf{Dec}(\tau, [\mathsf{sk}_{\mathbf{y}}]_2, \mathsf{ct})$

Fig. 12. Games H_i for $i \in \{1, \ldots, Q_{\mathsf{Dec}}+1\}$, $\mathsf{H}_{i.1}, \mathsf{H}_{i.2}$ for $i \in \{1, \ldots, Q_{\mathsf{Dec}}\}$ for the proof of Lemma 2. In each procedure, the components inside a solid (dotted) frame are only present in the games marked by a solid (dotted) frame.

References

ABDP15. M. Abdalla, F. Bourse, A. De Caro, and D. Pointcheval. Simple functional encryption schemes for inner products. In *PKC 2015*, *LNCS* 9020, pages 733–751. Springer, Heidelberg, March / April 2015.

AGRW17. M. Abdalla, R. Gay, M. Raykova, and H. Wee. Multi-input inner-product functional encryption from pairings. In *EUROCRYPT 2017, Part I*, *LNCS* 10210, pages 601–626. Springer, Heidelberg, April / May 2017.

AGVW13. S. Agrawal, S. Gorbunov, V. Vaikuntanathan, and H. Wee. Functional encryption: New perspectives and lower bounds. In *CRYPTO 2013, Part II*, *LNCS* 8043, pages 500–518. Springer, Heidelberg, August 2013.

AJ15. P. Ananth and A. Jain. Indistinguishability obfuscation from compact functional encryption. In *CRYPTO 2015, Part I*, *LNCS* 9215, pages 308–326. Springer, Heidelberg, August 2015.

ALS16. S. Agrawal, B. Libert, and D. Stehlé. Fully secure functional encryption for inner products, from standard assumptions. In *CRYPTO 2016, Part III*, *LNCS* 9816, pages 333–362. Springer, Heidelberg, August 2016.

AS17. P. Ananth and A. Sahai. Projective arithmetic functional encryption and indistinguishability obfuscation from degree-5 multilinear maps. In *EUROCRYPT 2017, Part I*, *LNCS* 10210, pages 152–181. Springer, Heidelberg, April / May 2017.

BBL17. F. Benhamouda, F. Bourse, and H. Lipmaa. CCA-secure inner-product functional encryption from projective hash functions. In *PKC 2017, Part II*, *LNCS* 10175, pages 36–66. Springer, Heidelberg, March 2017.

BCFG17. C. E. Z. Baltico, D. Catalano, D. Fiore, and R. Gay. Practical functional encryption for quadratic functions with applications to predicate encryption. In *CRYPTO 2017, Part I*, *LNCS* 10401, pages 67–98. Springer, Heidelberg, August 2017.

BGI+01. B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. In *CRYPTO 2001*, *LNCS* 2139, pages 1–18. Springer, Heidelberg, August 2001.

BSW11. D. Boneh, A. Sahai, and B. Waters. Functional encryption: Definitions and challenges. In *TCC 2011*, *LNCS* 6597, pages 253–273. Springer, Heidelberg, March 2011.

BV15. N. Bitansky and V. Vaikuntanathan. Indistinguishability obfuscation from functional encryption. In *56th FOCS*, pages 171–190. IEEE Computer Society Press, October 2015.

BW07. D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data. In *TCC 2007*, *LNCS* 4392, pages 535–554. Springer, Heidelberg, February 2007.

CHK04. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *EUROCRYPT 2004*, *LNCS* 3027, pages 207–222. Springer, Heidelberg, May 2004.

CW14. J. Chen and H. Wee. Semi-adaptive attribute-based encryption and improved delegation for Boolean formula. In *SCN 14*, *LNCS* 8642, pages 277–297. Springer, Heidelberg, September 2014.

Dam92. I. Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In *CRYPTO'91*, *LNCS* 576, pages 445–456. Springer, Heidelberg, August 1992.

DGP18. E. Dufour Sans, R. Gay, and D. Pointcheval. Reading in the dark: Classifying encrypted digits with functional encryption. Cryptology ePrint Archive, Report 2018/206, 2018. https://eprint.iacr.org/2018/206.

ElG84. T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *CRYPTO'84*, *LNCS* 196, pages 10–18. Springer, Heidelberg, August 1984.

FO99. E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *CRYPTO'99*, *LNCS* 1666, pages 537–554. Springer, Heidelberg, August 1999.

GGG+14. S. Goldwasser, S. D. Gordon, V. Goyal, A. Jain, J. Katz, F.-H. Liu, A. Sahai, E. Shi, and H.-S. Zhou. Multi-input functional encryption. In *EUROCRYPT 2014*, *LNCS* 8441, pages 578–602. Springer, Heidelberg, May 2014.

GKW16. R. Goyal, V. Koppula, and B. Waters. Semi-adaptive security and bundling functionalities made generic and easy. In *TCC 2016-B, Part II*, *LNCS* 9986, pages 361–388. Springer, Heidelberg, October / November 2016.

GPSW06. V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM CCS 06*, pages 89–98. ACM Press, October / November 2006. Available as Cryptology ePrint Archive Report 2006/309.

GVW15. S. Gorbunov, V. Vaikuntanathan, and H. Wee. Predicate encryption for circuits from LWE. In *CRYPTO 2015, Part II*, *LNCS* 9216, pages 503–523. Springer, Heidelberg, August 2015.

JR13. C. S. Jutla and A. Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. In *ASIACRYPT 2013, Part I*, *LNCS* 8269, pages 1–20. Springer, Heidelberg, December 2013.

KLM+18. S. Kim, K. Lewi, A. Mandal, H. Montgomery, A. Roy, and D. J. Wu. Function-hiding inner product encryption is practical. In *SCN 18*, *LNCS* 11035, pages 544–562. Springer, Heidelberg, September 2018.

KSW08. J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *EUROCRYPT 2008*, *LNCS* 4965, pages 146–162. Springer, Heidelberg, April 2008.

KW15. E. Kiltz and H. Wee. Quasi-adaptive NIZK for linear subspaces revisited. In *EUROCRYPT 2015, Part II*, *LNCS* 9057, pages 101–128. Springer, Heidelberg, April 2015.

Lin17. H. Lin. Indistinguishability obfuscation from SXDH on 5-linear maps and locality-5 PRGs. In *CRYPTO 2017, Part I*, *LNCS* 10401, pages 599–629. Springer, Heidelberg, August 2017.

LT17. H. Lin and S. Tessaro. Indistinguishability obfuscation from trilinear maps and block-wise local PRGs. In *CRYPTO 2017, Part I*, *LNCS* 10401, pages 630–660. Springer, Heidelberg, August 2017.

LV16. H. Lin and V. Vaikuntanathan. Indistinguishability obfuscation from DDH-like assumptions on constant-degree graded encodings. In *57th FOCS*, pages 11–20. IEEE Computer Society Press, October 2016.

NY90. M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd ACM STOC*, pages 427–437. ACM Press, May 1990.

O'N10. A. O'Neill. Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556, 2010. http://eprint.iacr.org/2010/556.

SW14. A. Sahai and B. Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In *46th ACM STOC*, pages 475–484. ACM Press, May / June 2014.

Wee14. H. Wee. Dual system encryption via predicate encodings. In *TCC 2014*, *LNCS* 8349, pages 616–637. Springer, Heidelberg, February 2014.

Wee17. H. Wee. Attribute-hiding predicate encryption in bilinear groups, revisited. In *TCC 2017, Part I*, *LNCS* 10677, pages 206–233. Springer, Heidelberg, November 2017.

YAHK11. S. Yamada, N. Attrapadung, G. Hanaoka, and N. Kunihiro. Generic constructions for chosen-ciphertext secure attribute based encryption. In *PKC 2011*, *LNCS* 6571, pages 71–89. Springer, Heidelberg, March 2011.