Theoretical Computer Science

A Tighter Proof for CCA Secure Inner Product Functional Encryption: Genericity Meets Efficiency

--Manuscript Draft--

Manuscript Number:	
Article Type:	Long Paper (&gt; 40 pages)
Section/Category:	A - Algorithms, automata, complexity and games
Keywords:	Public key cryptography; Functional encryption for inner products; Cryptography based on class groups of an imaginary quadratic field; Security proofs; Projective hash functions.
Corresponding Author:	Ida Heather Jane Tucker, Ph.D. IMDEA Software Institute Madrid, SPAIN
First Author:	Guilhem Castagnos, Maître de conférences
Order of Authors:	Guilhem Castagnos, Maître de conférences
	Fabien Laguillaumie, Professor
	Ida Tucker, Ph.D.

Powered by Editorial Manager® and ProduXion Manager® from Aries Systems Corporation

Manuscript (PDF)

A Tighter Proof for CCA Secure Inner Product Functional Encryption: Genericity Meets Efficiency

Guilhem Castagnos[1], Fabien Laguillaumie[2], and Ida Tucker[3]

[1] Université de Bordeaux, INRIA, CNRS, IMB UMR 5251,
F-33405 Talence, France.
[2] LIRMM, Université de Montpellier, CNRS, France.
[3] IMDEA Software Institute, Madrid, Spain.

Abstract. Inner product functional encryption (IPFE) is a primitive which produces, from a master secret key, decryption keys $\mathsf{sk}_{\boldsymbol{k}}$ associated to vectors $\boldsymbol{k}$ over some specified base ring. Decrypting an encryption of vector $\boldsymbol{m}$ with $\mathsf{sk}_{\boldsymbol{k}}$ only reveals $\langle \boldsymbol{k}, \boldsymbol{m} \rangle$. Benhamouda et al. [BBL17] provided a generic construction for CCA-secure IPFE from projective hash functions (PHFs), unfortunately their security reduction suffers an exponential loss. Their only instantiation capable of decrypting inner products of large sizes, which relies on the decisional composite residuosity (DCR) assumption, is impractical due to the large size of ciphertexts, decryption keys, and the prohibitive speed of the scheme. Our core contribution is a new approach to proving CCA security. Our constructions maintain the genericity of [BBL17], while our security proof relaxes the requirements on the underlying PHFs and gains in reduction tightness. We instantiate these constructions from the DCR assumption, an assumption in class groups ($\mathsf{HSM}_{\mathsf{CL}}$) and the decision Diffie Hellman (DDH) assumption. Our CCA-secure constructions from DCR and $\mathsf{HSM}_{\mathsf{CL}}$ are the first such schemes to efficiently decrypt inner products of large size, improving by *multiple orders of magnitude* upon the work of [BBL17]. A single-core C implementation of these schemes shows that, for an 112 bit security, and $100-$dimensional vectors, their DCR-based scheme takes 1h20min to encrypt, and over 5min to decrypt, whereas our *slowest* scheme takes 5.2s to encrypt and 0.5s to decrypt. Similarly a ciphertext for their scheme is of 283MB; those of our $\mathsf{HSM}_{\mathsf{CL}}$-based scheme are of 30kB.

Keywords: Public key cryptography; Functional encryption for inner products; Cryptography based on class groups of an imaginary quadratic field; Security proofs; Projective hash functions.

1 Introduction

Traditional public key encryption (PKE) provides an all-or-nothing access to data: given an encryption of m, a receiver either decrypts, and recovers the entire message m, or learns nothing about m. However many real life applications call for a more subtle disclosure of information, according to a receivers' privileges.

Functional encryption (FE) [SW05, BSW11, O'N10] is a refinement of PKE, providing control over how much of the encrypted data each user can recover. Specifically, it allows for a receiver to recover a function $f(m)$ of the encrypted message m, without revealing anything else about m. The primitive derives functional decryption keys sk_{f_i} – for specific functionalities f_i – from a master secret key msk. A single ciphertext c encrypting m is made available, from which a user possessing sk_{f_i} can recover $f_i(m) = \mathsf{Dec}(\mathsf{sk}_{f_i}, c)$.

Security. Two main security definitions exist for FE: indistinguishability-based and a stronger simulation-based security. The former is the model adopted throughout this paper, while we keep the latter for future work. Indistinguishability asks that no polynomial time adversary $\mathcal{A}$ can distinguish the encryptions of plaintexts m_0 and m_1 of its' choice. Granting different degrees of power to $\mathcal{A}$ defines various levels of security [BSW11, O'N10]. *Adaptive security against chosen plaintext attacks* ($\mathsf{ind\text{-}fe\text{-}cpa}$) requires indistinguishability holds when $\mathcal{A}$ is given keys sk_f for functions f of its' choice, satisfying $f(m_0) = f(m_1)$ (otherwise one can trivially distinguish both ciphertexts). The weaker *selective* security model requires $\mathcal{A}$ commits to m_0 and m_1 before seeing the public key, or being granted functional keys. In both these models, the adversary is *passive*: it attempts to obtain confidential information but follows the protocol. This ensures that a user, who is granted specific functional keys, learns no more information than that these keys are intended to reveal. It does not however capture adversaries which coerce honest users to run the decryption protocol (with keys unknown to $\mathcal{A}$) on potentially malformed ciphertexts. To deal with such *active* adversaries, which deviate from the protocol, one needs *security against chosen ciphertext attacks* ($\mathsf{ind\text{-}fe\text{-}cca}$-security) [NP15, BBL17]. Here $\mathcal{A}$ can request the decryption of any ciphertext (except the challenge ciphertext) with decryption key sk_f for *any* functionality f (even if $f(m_0) \neq f(m_1)$).

Inner Product Functional Encryption. Much effort has gone into building efficient FE schemes for restricted classes of functions to develop our understanding of FE, and to benefit practical applications. One such primitive is inner-product functional encryption (IPFE). Among other applications, it allows for statistical analysis and polynomial evaluation over encrypted data [KSW08]; for the construction of bounded collusion FE schemes for all circuits [ALS16]; and for that of trace-and-revoke systems [ABP$^+$17]. In short, IPFE is defined as follows: plaintexts are vectors $\boldsymbol{m} \in \mathcal{R}^\ell$, where $\mathcal{R}$ is a commutative ring; and functional decryption keys $\mathsf{sk}_{\boldsymbol{k}}$, derived from vectors $\boldsymbol{k} \in \mathcal{R}^\ell$, allow to recover the inner product $\langle \boldsymbol{m}, \boldsymbol{k} \rangle \in \mathcal{R}$ but reveal nothing else about $\boldsymbol{m}$.

This line of research was initiated by Abdalla *et al.* in [ABDP15] by providing the first selectively secure IPFE schemes relying on standard assumptions. Though of great theoretical interest, such schemes are not sufficiently secure for practical applications. Subsequent work improved security: in the public key setting, the first $\mathsf{ind\text{-}fe\text{-}cpa}$-secure schemes were put forth by Agrawal *et al.* [ALS16] under the learning with errors (LWE), decision Diffie Hellman (DDH)

and decision composite residuosity (DCR) assumptions. The ensuing schemes of Castagnos *et al.* [CLT18] improved efficiency in particular using the $\mathsf{HSM}_{\mathsf{CL}}$ assumption, a class group variant of DCR. Independently, many concurrent works have aimed at increasing the expressiveness of IPFE, rather than security e.g., [DOT18, ABG19, CDG$^+$18, DPP20]. The aforementioned schemes are secure against chosen plaintext attacks, this is the *minimum* security required of any public key cryptosystem.

Generic solutions from projective hash functions. Zhang *et al.* [ZMY17] and then independently, and more formally Benhamouda *et al.* [BBL17] provided generic constructions from projective hash functions with homomorphic properties to both ind-fe-cpa and ind-fe-cca-secure IPFE schemes, which they instantiate from DCR and DDH-like assumptions.

Projective hash functions (PHF) were introduced by Cramer and Shoup in [CS02] to build efficient chosen ciphertext (ind-cca) secure PKE from a range of cryptographic assumptions. Their ciphertexts have three components: a random word x in some NP language, the message masked by a hash of x for a (smooth) PHF, which ensures confidentiality, and a second hash of x for a (universal$_2$) PHF, which ensures ciphertext integrity as it is used to reject corrupted ciphertexts. The evaluation of the hash function in x can be computed either by someone knowing a witness for x together with the public key (called projection key), or without the witness if one knows the secret key (called hashing key).

Benhamouda *et al.* proceed similarly: to build ind-fe-cpa-secure IPFE for vectors of length ℓ, each message coordinate is masked with a different hash value of the same word x. If the PHF is homomorphic, a linear combination of the hashing keys enables decryption of the same linear combination of the coordinates, which is the inner product of the message and the coefficients of the linear combination. To reach ind-fe-cca security, they add ℓ hash values of x for ℓ independent universal$_2$ PHFs. Regrettably, their security proof suffers an exponential loss in a term of the security reduction, which induces a prohibitive blow up of key and ciphertext sizes. A natural question hence arises:

Can ind-fe-cca-*secure IPFE be efficient?*

Our contributions & techniques. This paper brings a positive answer to the above by providing a security proof for generic ind-fe-cca-secure IPFE using PHFs, whose reduction quality hugely improves upon previous work. The tightness of the proof is remarkable, since it was not clear how to overcome the loss (proportional to the size of the message space) present in pre-existing security reductions. The fact our result sacrifices neither efficiency nor genericity indicates that PHFs are the right abstraction for the construction of IPFE schemes.

Technique. In [BBL17]'s security proof the challenger $\mathcal{C}$ guesses the difference between challenge messages at the onset of the security experiment; $\mathcal{C}$ then constructs a hashing key which *depends on this difference*, and which allows to answer all the adversary $\mathscr{A}$'s queries without leaking information on the challenge bit. If $\mathcal{C}$'s guess turns out to be wrong, it aborts (hence the security loss).

To avoid this loss we adopt a starkly different proof technique, which brings together and extends ideas apparent in [ALS16] to build ind-fe-cpa schemes. We demonstrate that – conditioned on $\mathscr{A}$'s view and its choice of challenge messages – the challenge bit remains statistically hidden. Conditional probabilities allow us to carry out the analysis a posteriori, while ensuring security against adversaries which are adaptive w.r.t. key queries, decryption queries, and the choice of challenge messages. Hence our proof is independent of the chosen challenge messages and avoids the exponential loss inherent to the [BBL17] methodology.

New properties for PHFs. For each considered assumption, [ALS16] provide a specific proof, all of which follow a similar structure: they carefully evaluate the maximum information leaked to the adversary by the public key, decryption key queries and by the challenge ciphertext, and show that given this information, the part of the hashing key masking the challenge bit follows a distribution close to uniform. This technique resembles that used in [CS02] where the definition of smoothness allows to do exactly this, only in the context of PKE. Inspired by the above, we introduce the notion of *vector smoothness* for PHFs, which extends smoothness to the IPFE setting. This extension is not straight forward: to be generic we must capture vector spaces over $\mathbf{Z}/q\mathbf{Z}$ and lattices; and finding the exact definition (to e.g. finely upper bound the minimum for proofs using lattices) is technically challenging. From vector smooth PHFs we then generically build ind-fe-cpa-secure IPFE. When instantiated from $\mathsf{DDH}, \mathsf{DCR}$ and $\mathsf{HSM}_{\mathsf{CL}}$, our ind-fe-cpa-secure construction yields the schemes of [ALS16] and [CLT18]. These are the most efficient ind-fe-cpa-secure IPFE schemes to date. We thus provide a unified view of these schemes, retrieving them from a generic approach by extracting the essence of their multiple proofs.

The tightness of our ind-fe-cca security reduction relies on the decryption oracle rejecting *exactly* the decryption queries leaking *more* information than in an ind-fe-cpa attack. Identifying these queries and ensuring that *only* these be rejected is an important contribution of this article. At a high level, these decryption queries satisfy two criteria: $\mathscr{A}$ cannot request the decryption key $\mathsf{sk}_{\boldsymbol{k}}$, and the first component x of the ciphertext is not in the language. To ensure they are rejected, we modify the ind-fe-cpa scheme as follows: upon encryption of a message vector of length ℓ, one also computes ℓ evaluations of another PHF over x, using independently sampled hashing keys. The resulting ciphertext contains x, the masked message components, and all the aforementioned evaluations. Before decrypting the masked message values, the decryption algorithm checks that a combination (which depends on $\boldsymbol{k}$ associated to $\mathsf{sk}_{\boldsymbol{k}}$) of the hash evaluations in the ciphertext yield the expected value. For this check to ensure ciphertext integrity, we define a new property for PHFs: *vector universality.* Vector universality ensures that conditioned on all the information chosen by and given to $\mathscr{A}$, one cannot predict a linear combination of ℓ evaluations of the PHF (needed to compute a ciphertext that is not rejected) over an x not in the language (see the discussion prior Definition 14 for a more detailed intuition). Hence adding vector universal PHFs to our ind-fe-cpa scheme yields an ind-fe-cca scheme. We emphasize that this new property is crucial to the quality of our security reduction. We also believe that it can inspire properties for projective hash functions allowing for the construction of *other* advanced cryptographic primitives with high-quality security reductions.

Though our constructions are similar, the properties we require of our PHFs are notably different to those of [BBL17]. An involved comparison shows that the properties we require are implied by those required in [BBL17] (*cf.* Appx. E).

Efficiency, instantiations and C implementation. Table 1 gives a simplified comparison of the costs of our work compared to previous schemes for the instantiations that we will consider, based on DDH, DCR and $\mathsf{HSM}_{\mathsf{CL}}$. In this table, a is the dimension of the hash key space ($a = 2$ for DDH, $a = 1$ for DCR and $\mathsf{HSM}_{\mathsf{CL}}$). Our work shows that the cost of ind-fe-cca compared to ind-fe-cpa-security can be analogous to that of ind-cca security for PKE using the Cramer Shoup technique: with a factor 2 or 3 on sizes.

	ind-fe-cca		ind-fe-cpa
	[BBL17]	This work	[ALS16, BBL17, CLT18]
msk ring elts.	$a\ell(1+2\nu)$	$3a\ell$	$a\ell$
mpk group elts.	$\ell(1+2\nu)$	3ℓ	ℓ
Decryption key ring elts.	$a(1+2\nu)$	$3a$	a
Ciphertext group elts.	$a+\ell(1+\nu)$	$a+2\ell$	$a+\ell$

Table 1: Simplified summary of generic improvements

With [BBL17]'s proof technique of guessing the difference between challenge messages, a term related to the cardinality of the message space appears in the security reduction. To compensate this factor, they need to repeat ν times in parallel their ℓ universal$_2$ PHFs impacting the costs as shown in Table 1. In practice, ν is small (say 7) for DDH as the message space size is extremely restricted by a discrete logarithm computation during decryption. A large message space, provided by instantiations from DCR and $\mathsf{HSM}_{\mathsf{CL}}$ is more suited for more precise inner products computations. For the DCR scheme of [BBL17], the parameter ν grows (at least) linearly in ℓ and blows up for concrete instantiation (say 5000). Our proof technique, independent of the challenge messages chosen by the adversary, allows to remove this term in the security reduction. Consequently, we get a huge efficiency improvement corresponding to setting $\nu = 1$. The improvement is similar regarding computational complexity.

In the main body of the paper we detail the DDH case as it provides a framework with which most readers are familiar. We also present the $\mathsf{HSM}_{\mathsf{CL}}$ case, using the PHF of [CCL$^+$19], inspired by [CLT18]. We only detail the DCR instantiation in Appx. D, as it resembles that from $\mathsf{HSM}_{\mathsf{CL}}$, only with less technical subtleties.

We implemented in C our $\mathsf{ind\text{-}fe\text{-}cca}$-secure schemes from $\mathsf{HSM_{CL}}$ and DCR; and the DCR based scheme of [BBL17]. In accordance with Table 1, our improvements are staggering. Both our schemes are multiple orders of magnitude more efficient than those of [BBL17] (in terms of speed and size of elements). Our DCR scheme is slightly faster than the $\mathsf{HSM_{CL}}$ one, but less compact. Precisely for a security level of 112 bits, and vectors of length 100, our DCR (resp. $\mathsf{HSM_{CL}}$) ciphertexts are of 103kB (resp. 30kB), and decryption keys of 1.5kB (resp. 0.3kB), as opposed to 283MB and 13.4MB respectively for Benhamouda *et al.*'s DCR based scheme. In terms of timings, encryption and decryption take us respectively 1.6s (resp. 5.2s) and 0.08s (resp. 0.5s), as opposed to 1h20min and 5.43min.

Road map. Sec. 2 provides basic notations; defines the assumptions underlying our running examples; defines the IPFE primitive and considered security model; and provides crucial definitions and notations related to PHFs. Sec. 3 defines new properties for PHFs allowing to build correct and secure IPFE. Sec. 4 and 5 present generic $\mathsf{ind\text{-}fe\text{-}cpa}$ and $\mathsf{ind\text{-}fe\text{-}cca}$-secure constructions for IPFE from PHFs. Sec. 6 compares efficiency of our schemes compared to those of [BBL17]. Finally Sec. 7 concludes and raises ideas for future work.

2 Preliminaries

2.1 Basic terminology and notations

For our constructions we use one time signatures (OTS) and families of collision resistant hash functions (CRHF). An OTS (resp. hash function) is δ-strongly unforgeable (resp. δ-collision resistant) if no adversary attacking the scheme can break the strong unforgeability (resp. collision resistance) with probability $\geqslant \delta$.

We denote sets by upper-case letters, matrices by bold upper-case letters, and vectors by bold lower-case letters s.t. for $\boldsymbol{a} \in A^\ell$, $\boldsymbol{a} = (a_1, \ldots, a_\ell)$. For an element g of a group G, $\langle g \rangle$ is the subgroup of G generated by g. For an integer x, $|x|$ denotes its' size and $[x]$ is the set of integers $\{1, \ldots, x\}$. The inner product of $\boldsymbol{x}$ and $\boldsymbol{y}$ is denoted $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$; their coordinate-wise product is denoted $\boldsymbol{x} \odot \boldsymbol{y}$. If $f : A \mapsto B$ is a function defined over A with co-domain B, and $\boldsymbol{a} \in A^\ell$, then $f(\boldsymbol{a}) := (f(a_1), \ldots, f(a_\ell)) \in B^\ell$. Given a ring $\mathcal{R}$, and a positive integer a, consider vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_a \in \mathcal{R}^\ell$, we define inner products between $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_a) \in (\mathcal{R}^\ell)^a$ and $\boldsymbol{y} \in \mathcal{R}^\ell$ as $\lessdot \boldsymbol{X}, \boldsymbol{y} \gtrdot_a := (\langle \boldsymbol{x}_1, \boldsymbol{y} \rangle, \ldots, \langle \boldsymbol{x}_a, \boldsymbol{y} \rangle) \in \mathcal{R}^a$. If $\mathcal{R}$ is either $\mathbf{Z}$ or $\mathbf{Z}/q\mathbf{Z}$ for some prime q, we denote $\boldsymbol{m}^\perp$ the subset of $\mathcal{R}^\ell$ defined as $\boldsymbol{m}^\perp := \{\boldsymbol{k} \in \mathcal{R}^\ell \mid \langle \boldsymbol{m}, \boldsymbol{k} \rangle = 0 \in \mathcal{R}\}$.

For any $\boldsymbol{c} \in \mathbf{R}^\ell$, real $\sigma > 0$, and ℓ-dimensional lattice Λ, $\mathcal{D}_{\Lambda,\sigma,\boldsymbol{c}}$ denotes the usual discrete Gaussian distribution of support Λ, standard deviation parameter σ and center $\boldsymbol{c}$. If $\boldsymbol{c} = \boldsymbol{0}$, we shorten the notation to $\mathcal{D}_{\Lambda,\sigma}$. Background on lattices is provided in Appx. A. For a distribution $\mathcal{D}$, we write $d \hookleftarrow \mathcal{D}$ to refer to d being sampled from $\mathcal{D}$; in our proofs we overload this notation to refer to a random variable d following the probability distribution $\mathcal{D}$. We denote $\mathcal{U}(B)$ the uniform distribution over the finite set B, and sometimes overload the notation as $b \hookleftarrow B$

to say $b \hookleftarrow \mathcal{U}(B)$. We say that a set (or a group) E is efficiently recognisable if its elements are uniquely encoded as bit strings of length bounded by $\mathsf{poly}(\lambda)$ (λ is the security parameter), and there exists an algorithm that determines, in time $\mathsf{poly}(\lambda)$, if a bit string is a valid encoding of an element of E. A PTA refers to an algorithm running in polynomial time (w.r.t. the length of its inputs); a PPTA (resp DPTA) a probabilistic (resp. deterministic) PTA.

2.2 An instantiation of the CL framework

Castagnos and Laguillaumie introduced the framework of a group with an easy discrete logarithm (DL) subgroup in [CL15] (enhanced in [CLT18, CCL$^+$19]), and gave a concrete instantiation from class groups of quadratic fields. We refer to this framework as the CL framework. This is the basis of our running example 1. For background on class groups in cryptography see [BH01] and [CL15, Appx. B].

We briefly sketch the instantiation given in [CCL$^+$19, Sec. 4.1] and the resulting group generator GenGroup that we use in this paper. The interested reader can refer to [CL15, CCL$^+$19] for details.

Given a prime p consider another random prime q, the fundamental discriminant $\Delta_K = -pq$ and the associated class group $C(\Delta_K)$. By choosing q s.t. $pq \equiv -1 \pmod 4$ and $(p/q) = -1$, we have that the $2-$Sylow subgroup of $C(\Delta_K)$ has order 2. The size of q is chosen s.t. computing the class number $h(\Delta_K)$ takes time 2^λ. We then consider the class group $C(\Delta_p)$ of discriminant $\Delta_p = p^2\Delta_K$, and denote $(\widehat{G}, \cdot)$ the finite abelian subgroup of squares of $C(\Delta_p)$, which corresponds to the odd part. Elements of $\widehat{G}$ are efficiently recognisable (*cf.* [Lag80]). One can exhibit a cyclic subgroup F of $\widehat{G}$ of order p generated by $f \in \widehat{G}$ where f is represented by an ideal of norm p^2. There exists a DPTA for the DL problem in F (*cf.* [CL15, Prop. C–1]).

Let $\widehat{s} := h(\Delta_K)/2$, and denote $\widehat{n} := h(\Delta_p)/2$ the order of $\widehat{G}$; one can show that $\widehat{n} := \widehat{s}p$. For our applications $|p| \geqslant \lambda$, where λ is the security parameter. So p is prime to $h(\Delta_K)$ (and hence $\widehat{s}$) with overwhelming probability. We define $\widehat{G}^p$ as the subgroup of all p-th powers in $\widehat{G}$; one can show that $\widehat{G} \simeq \widehat{G}^p \times F$, and that $\widehat{G}^p$ is of order $\widehat{s}$. The exponent of a finite Abelian group is the least common multiple of the orders of its elements. We denote ϖ the group exponent of $\widehat{G}^p$. As such, ϖ and p are co-prime, and the order of any $x \in \widehat{G}^p$ divides ϖ.

Then we build deterministically a $p-$th power of $\widehat{G}$ by lifting the class of an ideal of discriminant Δ_K above the smallest splitting prime. In the following, we denote g_p this deterministic generator. One can compute an upper bound $\tilde{s}$ for the order s of g_p, using an upper bound of $h(\Delta_K)$. For this, one can use the fact that $h(\Delta_K) < \frac{1}{\pi}\log|\Delta_K|\sqrt{|\Delta_K|}$, or obtain a slightly better bound from the analytic class number formula. We let $g := g_p f$ and denote G the subgroup generated by g of order $n := ps$. Since p and s are co-prime (s divides ϖ), one can write $G \simeq G^p \times F$ where $G^p = \langle g_p \rangle$.

Notation. We denote GenGroup the PPTA that on input a security parameter λ and a prime p, outputs $(\tilde{s}, f, g_p, \widehat{G}, F)$ defined as above. We denote Solve the

DPTA that solves the DL problem in F. This pair of algorithms is an instance of the framework of a group with an easy DL subgroup (*cf.* [CCL$^+$19, Def. 4]).

Hard subgroup membership assumption. We recall the definition of the $\mathsf{HSM}_{\mathsf{CL}}$ assumption, which states it is hard to distinguish elements of G^p in G. It is closely related to Paillier's DCR assumption: they are essentially the same assumption in different groups. $\mathsf{HSM}_{\mathsf{CL}}$ was first used by [CLT18] within class groups. This will be the basis of running example 1 throughout Sec. 2.5 and 3.

Definition 1 **($\mathsf{HSM}_{\mathsf{CL}}$).** *For $(\tilde{s}, f, g_p, \widehat{G}, F) \leftarrow \mathsf{GenGroup}(1^\lambda, p)$, and $g := g_p f$, we denote $\mathcal{D}$ (resp. $\mathcal{D}_p$) a distribution over the integers s.t. the distribution $\{g^x, x \hookleftarrow \mathcal{D}\}$ (resp. $\{g_p^x, x \hookleftarrow \mathcal{D}_p\}$) is at distance less than $2^{-\lambda}$ from the uniform distribution in $G := \langle g \rangle$ (resp. in $G^p := \langle g_p \rangle$). The $\mathsf{HSM}_{\mathsf{CL}}$ problem is hard for $\mathsf{GenGroup}$ if for all PPT algorithm $\mathcal{A}$,*

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{HSM}_{\mathsf{CL}}}(\lambda) := \Big| 2 \cdot \Pr\big[b = b^\star \mid (\tilde{s}, f, g_p, \widehat{G}, F) &\leftarrow \mathsf{GenGroup}(1^\lambda, p), \\ x \hookleftarrow \mathcal{D}, x' \hookleftarrow \mathcal{D}_p, b \hookleftarrow \{0,1\}, Z_0 &\leftarrow g^x, Z_1 \leftarrow g_p^{x'}, \\ b^\star \leftarrow \mathcal{A}(p, \tilde{s}, f, g_p, \widehat{G}, F, Z_b, \mathsf{Solve}(.))\big] - 1 \Big| &= \mathsf{negl}(\lambda).\end{aligned}$$

Remark 1. To construct ind-cca-secure schemes, we will need to work with recognisable groups ($\widehat{G}$ instead of G), so we sample exponents from $\widehat{\mathcal{D}}$ and $\widehat{\mathcal{D}}_p$, such that $\{x \bmod \widehat{n}; x \hookleftarrow \widehat{\mathcal{D}}\}$ and $\{x \bmod \widehat{s}; x \hookleftarrow \widehat{\mathcal{D}}\}$ are statistically close to $\mathcal{U}(\mathbf{Z}/\widehat{n}\mathbf{Z})$ and $\mathcal{U}(\mathbf{Z}/\widehat{s}\mathbf{Z})$ respectively. In practice since the upper bound $\tilde{s}$ output by $\mathsf{GenGroup}$ is an upper bound for $\widehat{s}$, we can set $\widehat{\mathcal{D}} := \mathcal{D}$ (resp. $\widehat{\mathcal{D}}_p := \mathcal{D}_p$). We instantiate $\widehat{\mathcal{D}} = \mathcal{D}_{\mathbf{Z},\sigma}$ (resp. $\widehat{\mathcal{D}}_p = \mathcal{D}_{\mathbf{Z},\sigma'}$) as folded gaussians provide shorter keys than folded uniforms. Choosing $\sigma = \sqrt{\lambda} \cdot \tilde{s} \cdot p$ (resp. $\sigma' = \sqrt{\lambda} \cdot \tilde{s}$) ensures the aforementioned statistical distances are less than $2^{-\lambda}$ [CLT18, Lemma 4].

2.3 The DDH assumption

This will be the basis of running example 2 throughout Sections 2.5 and 3.

Definition 2. *Let $\mathsf{Gen}_{\mathsf{DDH}}(1^\lambda)$ be a generator outputting the description of a cyclic group $(G, \cdot)$, of prime order q and generated by g. The Decision Diffie-Hellman (DDH) problem is hard for $\mathsf{Gen}_{\mathsf{DDH}}$ if for all PPT algorithm $\mathcal{A}$,*

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{DDH}}(\lambda) := \Big| 2 \cdot \Pr\big[b = b^\star \mid (G, g, q) \leftarrow \mathsf{Gen}_{\mathsf{DDH}}(1^\lambda), \alpha, \beta, \gamma \hookleftarrow \mathcal{U}(\mathbf{Z}/q\mathbf{Z}), \\ b \hookleftarrow \{0,1\}, Z_0 \leftarrow g^{\alpha\beta}, Z_1 \leftarrow g^\gamma, b^\star \leftarrow \mathcal{A}(G, g, q, g^\alpha, g^\beta, Z_b)\big] - 1 \Big| = \mathsf{negl}(\lambda).\end{aligned}$$

2.4 Inner product functional encryption

Inner product functional encryption is a special case of functional encryption, as first formalised in [BSW11].

Definition 3 (Inner product functional encryption). *Let ℓ be a positive integer. Consider a ring $\mathcal{R} \in \{\mathcal{R}_\lambda\}_{\lambda \in \mathbf{N}}$, a key space $\mathcal{K}$ and a message space $\mathcal{M}$; where $\mathcal{K}$ and $\mathcal{M}$ are efficiently recognisable subsets of $\mathcal{R}^\ell$. An inner product functional encryption scheme over $\mathcal{R}$, for vectors of length ℓ, key space $\mathcal{K}$ and message space $\mathcal{M}$ is a tuple* $(\mathsf{Setup}, \mathsf{KeyDer}, \mathsf{Enc}, \mathsf{Dec})$ *of algorithms where:*

- Setup *on input a security parameter 1^λ, outputs a master public key* mpk *and a master secret key* msk*;*
- KeyDer *on input* msk *and a key $\boldsymbol{k} \in \mathcal{K}$, outputs a decryption key $\mathsf{sk}_{\boldsymbol{k}}$;*
- Enc *on input* mpk *and a message $\boldsymbol{m} \in \mathcal{M}$, outputs a ciphertext c;*
- Dec *on input* mpk*, a key $\mathsf{sk}_{\boldsymbol{k}}$ and a ciphertext c, outputs $v \in \mathcal{R} \cup \{\perp\}$, where $\perp$ is a special error symbol.*

Correctness requires that for all $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$*, all keys $\boldsymbol{k} \in \mathcal{K}$ and all messages $\boldsymbol{m} \in \mathcal{M}$, if $\mathsf{sk}_{\boldsymbol{k}} \leftarrow \mathsf{KeyDer}(\mathsf{msk}, \boldsymbol{k})$ and $c \leftarrow \mathsf{Enc}(\mathsf{mpk}, \boldsymbol{m})$, then for $v \leftarrow \mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_{\boldsymbol{k}}, c)$ it holds that $v = \langle \boldsymbol{k}, \boldsymbol{m} \rangle \in \mathcal{R}$ whenever $v \neq \perp$.*

Security. Intuitively, given a ciphertext encrypting $\boldsymbol{m}$, the only information obtained from decryption key $\mathsf{sk}_{\boldsymbol{k}}$ should be the evaluation $\langle \boldsymbol{k}, \boldsymbol{m} \rangle$. We consider an extension of the existing game-based definition of FE [BSW11] which deals with *active adversaries*, by allowing them to perform decryption queries for ciphertexts of their choice. The following definition is that of *adaptive* security, meaning that $\mathcal{A}$ has access to the systems' public parameters, and can perform a series of decryption and key derivation queries *before* choosing $\boldsymbol{m}_0$ and $\boldsymbol{m}_1$.

The experiment $\mathsf{Exp}^{\mathsf{ind\text{-}fe\text{-}cca}}_{\mathsf{IPFE},\mathcal{A}}$. Let $\mathsf{IPFE} := (\mathsf{Setup}, \mathsf{KeyDer}, \mathsf{Enc}, \mathsf{Dec})$ be an IPFE scheme over ring $\mathcal{R}$, a key space $\mathcal{K} \subseteq \mathcal{R}^\ell$ and a message space $\mathcal{M} \subseteq \mathcal{R}^\ell$. For $\lambda \in \mathbf{N}$, $\mathsf{Exp}^{\mathsf{ind\text{-}fe\text{-}cca}}_{\mathsf{IPFE},\mathcal{A}}(\lambda)$ denotes the random variable defined via the following experiment, involving a PPT adversary $\mathcal{A}$ and a challenger $\mathcal{C}$:

1. **Setup:** $\mathcal{C}$ samples $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and $\beta \hookleftarrow \{0, 1\}$.
2. **Pre-challenge:** $\mathcal{A}$ on input $(1^\lambda, \mathsf{mpk})$ adaptively issues queries:
   - $(\mathsf{key}, \boldsymbol{k})$ where $\boldsymbol{k} \in \mathcal{K}$. Upon receiving query $(\mathsf{key}, \boldsymbol{k})$, $\mathcal{C}$ computes $\mathsf{sk}_{\boldsymbol{k}} \leftarrow \mathsf{KeyDer}(\mathsf{msk}, \boldsymbol{k})$; and sends $\mathsf{sk}_{\boldsymbol{k}}$ to $\mathcal{A}$.
   - $(\mathsf{decrypt}, c, \boldsymbol{k})$ where $\boldsymbol{k} \in \mathcal{K}$ and c is a ciphertext. Upon receiving query $(\mathsf{decrypt}, c, \boldsymbol{k})$, $\mathcal{C}$ computes $\mathsf{sk}_{\boldsymbol{k}} \leftarrow \mathsf{KeyDer}(\mathsf{msk}, \boldsymbol{k})$; $\mathsf{res} \leftarrow \mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_{\boldsymbol{k}}, c)$; and sends res to $\mathcal{A}$.
3. **Challenge:** $\mathcal{A}$ outputs $\boldsymbol{m}_0, \boldsymbol{m}_1 \in \mathcal{M}$, then $\mathcal{C}$ computes $c^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, \boldsymbol{m}_\beta)$ and sends c^* to $\mathcal{A}$.
4. **Post-challenge:** $\mathcal{A}$ adaptively issues queries as in the pre-challenge phase.
5. **Output:** $\mathcal{A}$ outputs β'. $\mathsf{Exp}^{\mathsf{ind\text{-}fe\text{-}cca}}_{\mathsf{IPFE},\mathcal{A}}(\lambda)$ outputs 1 if and only if $\beta = \beta'$.

Valid adversaries. As standard in FE, we rule out adversaries that can easily distinguish between the challenge messages $\boldsymbol{m}_0$ and $\boldsymbol{m}_1$ using their queries. Specifically, an adversary is *valid* if all key queries $(\mathsf{key}, \boldsymbol{k})$ satisfy $\langle \boldsymbol{k}, \boldsymbol{m}_0 \rangle = \langle \boldsymbol{k}, \boldsymbol{m}_1 \rangle$, and all decryption queries $(\mathsf{decrypt}, c, \boldsymbol{k})$ satisfy $c \neq c^*$.

Having defined the experiment $\mathsf{Exp}^{\mathsf{ind\text{-}fe\text{-}cca}}_{\mathsf{IPFE},\mathcal{A}}$ and valid adversaries, we can now define adaptive security against chosen ciphertext attacks for IPFE schemes.

Definition 4. *An IPFE scheme* $\mathsf{IPFE} := (\mathsf{Setup}, \mathsf{KeyDer}, \mathsf{Enc}, \mathsf{Dec})$ *over ring* $\mathcal{R}$*, key space* $\mathcal{K} \subseteq \mathcal{R}^\ell$ *and message space* $\mathcal{M} \subseteq \mathcal{R}^\ell$ *is adaptively secure against chosen ciphertext attacks* ($\mathsf{ind\text{-}fe\text{-}cca}$) *if for any PPT valid adversary* $\mathcal{A}$*,*

$$\mathsf{Adv}^{\mathsf{ind\text{-}fe\text{-}cca}}_{\mathsf{IPFE},\mathcal{A}}(\lambda) := \left| \Pr\left[\mathsf{Exp}^{\mathsf{ind\text{-}fe\text{-}cca}}_{\mathsf{IPFE},\mathcal{A}}(\lambda) = 1\right] - \frac{1}{2} \right| = \mathsf{negl}(\lambda).$$

The definition of adaptive security against chosen plaintext attacks ($\mathsf{ind\text{-}fe\text{-}cpa}$-security) is similar, only $\mathcal{A}$ cannot perform decryption queries.

Remark 2. This definition of $\mathsf{ind\text{-}fe\text{-}cca}$-security is equivalent to that of [BBL17]. The work of Zhang *et al.* [ZMY17] does not provide a formal definition for $\mathsf{ind\text{-}cca}$-security. This being said, their model is more restrictive as they bound the number of key derivation queries allowed by the adversary.

2.5 Projective hash functions

Using the formalism of Cramer and Shoup [CS02], we define projective hash functions. The definitions we provide are for a general class of group-theoretic language membership problems; they are a slight adaptation of the definitions of [CS02], since we consider the more general case where the considered groups may not be efficiently recognisable, thus allowing for a wider range of instantiations. To build PHFs one starts with an instance of a subgroup membership problem.

Definition 5. *A generator for a* $\delta_{\mathcal{L}}$*-hard subgroup membership problem (SMP) is a PPT algorithm* $\mathsf{Gen}_{\mathcal{SM}}$ *which on input* 1^λ *returns the description of a subgroup membership problem* $\mathcal{SM} := (\widehat{\mathcal{X}}, \mathcal{X}, \widehat{\mathcal{L}}, \mathcal{W}, \mathsf{R})$*, where:*

- $\widehat{\mathcal{X}}$ *is an efficiently recognisable finite Abelian group;*
- $\mathcal{X} \subseteq \widehat{\mathcal{X}}$ *is a subgroup of* $\widehat{\mathcal{X}}$ *(which may not be recognisable);*
- $\widehat{\mathcal{L}} \subset \widehat{\mathcal{X}}$ *is a subgroup of* $\widehat{\mathcal{X}}$*, and* $\mathcal{L} := \mathcal{X} \cap \widehat{\mathcal{L}}$*;*
- $\mathsf{R} \subset \mathcal{X} \times \mathcal{W}$ *is a binary relation. For* $x \in \mathcal{L}$ *and* $w \in \mathcal{W}$*,* w *is a witness for* x *if* $(x, w) \in \mathsf{R}$*. The relation* R *is efficiently samplable: one samples a random* $x \in \mathcal{L}$ *along with a witness* $w \in \mathcal{W}$ *for* x*, this implicitly defines a way to sample random elements of* $\mathcal{L}$*. We denote this sampling* $(x, w) \leftarrow \mathsf{R}$*;*
- *It is hard to distinguish random elements of* $\mathcal{L}$ *from those of* $\mathcal{X}$*. Precisely* $\delta_{\mathcal{L}}$ *is the maximal advantage of any PPT adversary in solving this problem.*

If $\widehat{\mathcal{X}} = \mathcal{X}$ *then* $\widehat{\mathcal{L}} = \mathcal{L}$*, and we simply denote* $\mathcal{SM} := (\mathcal{X}, \mathcal{L}, \mathcal{W}, \mathsf{R})$*.*

Remark 3. In this definition 'random' and 'samplable' mean for a distribution that is to be defined when instantiating the subgroup membership problem.

Example 1 − $\mathsf{HSM}_{\mathsf{CL}}$ **.** Consider $(\tilde{s}, f, g_p, \widehat{G}, F) \leftarrow \mathsf{GenGroup}(1^\lambda, p)$, and $g := g_p f$. Let $\widehat{\mathcal{X}} := \widehat{G}$, $\mathcal{X} := G = \langle g \rangle$ and $\widehat{\mathcal{L}} := \widehat{G}^p$. Then $\mathcal{L} := \mathcal{X} \cap \widehat{\mathcal{L}} = G^p = \langle g_p \rangle$. A witness for $x \in \mathcal{L}$ is $w \in \mathbf{Z}$ satisfying $x = g_p^w$; we denote $\mathsf{R}_{\mathsf{CL}} := \{(x, w) \in (G^p \times \mathbf{Z}) \mid x = g_p^w\}$. Thus $\mathcal{SM}_{\mathsf{CL}} = (\widehat{G}, G, \widehat{G}^p, \mathbf{Z}, \mathsf{R}_{\mathsf{CL}})$. Witnesses are sampled from $\mathcal{D}_p$, a distribution on $\mathbf{Z}$ which induces a distribution δ-close to uniform on G^p (*cf.* Remark 1). Sampling in G is done by sampling $w \hookleftarrow \mathcal{D}_p$, $u \hookleftarrow \mathbf{Z}/p\mathbf{Z}$, and outputting $g_p^w f^u$, this induces a distribution δ-close to uniform on G. Recall that the $\mathsf{HSM}_{\mathsf{CL}}$ assumption states it is hard to distinguish elements of G^p in G. So if the $\mathsf{HSM}_{\mathsf{CL}}$ problem is hard, $\mathcal{SM}_{\mathsf{CL}}$ is δ_{CL}-hard where $\delta_{\mathsf{CL}} = \mathsf{negl}(\lambda)$.

Example 2 – DDH **.** Consider $(G, g, q) \leftarrow \mathsf{Gen}_{\mathsf{DDH}}(1^\lambda)$, and g_0, g_1 two generators of G. Here G is assumed efficiently recognisable, so we can set $\widehat{\mathcal{X}} = \mathcal{X} := G \times G$, and $\widehat{\mathcal{L}} = \mathcal{L}$ is the subgroup of $\mathcal{X}$ generated by (g_0, g_1). A witness for $(x_0, x_1) \in \mathcal{L}$ is $w \in \mathbf{Z}/q\mathbf{Z}$ satisfying $(x_0, x_1) = (g_0^w, g_1^w)$; we denote $\mathsf{R}_{\mathsf{DDH}} := \{((x_0, x_1), w) \in \mathcal{L} \times \mathbf{Z}/q\mathbf{Z} \mid (x_0, x_1) = (g_0^w, g_1^w)\}$. Thus $\mathcal{SM}_{\mathsf{DDH}} := (G \times G, \langle (g_0, g_1) \rangle, \mathbf{Z}/q\mathbf{Z}, \mathsf{R}_{\mathsf{DDH}})$. One samples elements of $\mathcal{L}$ by sampling a witness $w \hookleftarrow \mathcal{U}(\mathbf{Z}/q\mathbf{Z})$, and outputting (g_0^w, g_1^w). Sampling on $G \times G$ is done by sampling $r, r' \hookleftarrow \mathcal{U}(\mathbf{Z}/q\mathbf{Z})$, and outputting[4] $(g_0^r, g_1^r) \odot (1, g_1^{r'})$. The hardness of $\mathcal{SM}_{\mathsf{DDH}}$ is implied by that of DDH for $\mathsf{Gen}_{\mathsf{DDH}}$: if the DDH problem is hard, $\mathcal{SM}_{\mathsf{DDH}}$ is δ_{DDH}-hard where $\delta_{\mathsf{DDH}} = \mathsf{negl}(\lambda)$.

We can now define projective hash functions.

Definition 6. *Let $\mathcal{SM} := (\widehat{\mathcal{X}}, \mathcal{X}, \widehat{\mathcal{L}}, \mathcal{W}, \mathsf{R})$ be a subgroup membership problem. A projective hash function (PHF) for $\mathcal{SM}$ is a tuple of algorithms $\mathsf{H} := (\mathsf{hashkg}, \widehat{\mathsf{projkg}}, \mathsf{projkg}, \mathsf{hash}, \widehat{\mathsf{projhash}}, \mathsf{projhash})$, where:*

- hashkg *is a PPTA which on input the description of $\mathcal{SM}$, outputs a hashing key* hk *in some set K_{hk};*
- $\widehat{\mathsf{projkg}}$ *is a deterministic algorithm which on input $\mathsf{hk} \in K_{\mathsf{hk}}$ outputs a projection key $\widehat{\mathsf{hp}}$. The image of K_{hk} through $\widehat{\mathsf{projkg}}$ is denoted $K_{\widehat{\mathsf{hp}}}$;*
- projkg *is a DPTA which on input $\mathsf{hk} \in K_{\mathsf{hk}}$ outputs a public projection key hp, such that for $\mathsf{hk} \in K_{\mathsf{hk}}$, hp is a fixed deterministic function of the output of $\widehat{\mathsf{projkg}}(\mathsf{hk})$. The image of K_{hk} through projkg is denoted K_{hp};*
- hash *is a DPTA which on input $\mathsf{hk} \in K_{\mathsf{hk}}$, $x \in \widehat{\mathcal{X}}$ outputs the hash value $\mathsf{hash}(\mathsf{hk}, x)$. The image of $\widehat{\mathcal{X}}$ through hash is a finite Abelian group called the* set of hash values *and is denoted Π;*
- $\widehat{\mathsf{projhash}}$ *is a deterministic algorithm which on input $\widehat{\mathsf{hp}} \in K_{\widehat{\mathsf{hp}}}$ and $x \in \widehat{\mathcal{L}}$, outputs the hash value $\widehat{\mathsf{projhash}}(\widehat{\mathsf{hp}}, x)$ in Π;*
- $\mathsf{projhash}$ *is a DPTA which on input $\mathsf{hp} \in K_{\mathsf{hp}}$, $x \in \mathcal{L}$ and the corresponding witness $w \in \mathcal{W}$, outputs the hash value $\mathsf{projhash}(\mathsf{hp}, x, w)$ in Π.*

One says H is correct if for any $x \in \widehat{\mathcal{L}}$, any $\mathsf{hk} \in K_{\mathsf{hk}}$, $\widehat{\mathsf{hp}} \leftarrow \widehat{\mathsf{projkg}}(\mathsf{hk})$, it holds that $\widehat{\mathsf{projhash}}(\widehat{\mathsf{hp}}, x) = \mathsf{hash}(\mathsf{hk}, x)$; and if for any $(x, w) \in \mathsf{R}$, and $\mathsf{hp} \leftarrow \mathsf{projkg}(\mathsf{hk})$, it holds that $\mathsf{projhash}(\mathsf{hp}, x, w) = \widehat{\mathsf{projhash}}(\widehat{\mathsf{hp}}, x)$.

If $\widehat{\mathcal{X}} = \mathcal{X}$, then $\widehat{\mathsf{projkg}} = \mathsf{projkg}$ and $\widehat{\mathsf{projhash}} = \mathsf{projhash}$, and we simply denote $\mathsf{H} := (\mathsf{hashkg}, \mathsf{projkg}, \mathsf{hash}, \mathsf{projhash})$.

Remark 4. Algorithms $\widehat{\mathsf{projkg}}$ and $\widehat{\mathsf{projhash}}$ of Def. 6 will only be used to prove security of our constructions, so as to quantify the maximum information an

[4] This choice, which may seem convoluted when one could output $(g_0^r, g_1^{r'})$, is for consistency with the notion of decomposability introduced in Section 3.2.

adversary can learn. As such these algorithms needn't be efficiently computable, and one does not need a witness to evaluate $\widehat{\mathsf{projhash}}$.

Linearly homomorphic PHF. If PHFs satisfy some homomorphic properties, they allow for the construction of advanced cryptographic primitives. In particular we will need the following two definitions for correctness of our constructions.

Definition 7 ([HO09]). *Recall that $\mathit{\Pi}$ is the set of hash values, and that $(\widehat{\mathcal{X}}, \cdot)$ and $(\mathit{\Pi}, \cdot)$ are Abelian groups. A PHF* H *is* homomorphic *if for all* $\mathsf{hk} \in K_{\mathsf{hk}}$*,* $\mathsf{hash}(\mathsf{hk}, \cdot)$ *is a group homomorphism from $\widehat{\mathcal{X}}$ to $\mathit{\Pi}$.*

Remark 5. If H is correct and homomorphic, for all $\widehat{\mathsf{hp}} \in K_{\widehat{\mathsf{hp}}}, \mathsf{hp} \in K_{\mathsf{hp}}$, the functions $\widehat{\mathsf{projhash}}(\widehat{\mathsf{hp}}, \cdot)$ and $\mathsf{projhash}(\mathsf{hp}, \cdot, \cdot)$ are group homomorphisms from respectively $\widehat{\mathcal{L}}$ and $\mathcal{L}$ to $\mathit{\Pi}$.

Definition 8 ([BBL17]). *A PHF is* key homomorphic *if $(K_{\mathsf{hk}}, +)$ and $(\mathit{\Pi}, \cdot)$ are Abelian groups; and $\forall x \in \widehat{\mathcal{X}}$,* $\mathsf{hash}(\cdot, x)$ *is a group homomorphism from K_{hk} to $\mathit{\Pi}$.*

Example 1 – $\mathsf{HSM}_{\mathsf{CL}}$ **.** We define the PHF H_{CL} from $\mathcal{SM}_{\mathsf{CL}}$ as follows. The hash key space is $K_{\mathsf{hk}} := \mathbf{Z}$. Algorithm hashkg samples $\mathsf{hk} \hookleftarrow \widehat{\mathcal{D}}$ s.t. $(\mathsf{hk} \bmod \widehat{n})$ follows a distribution δ-close to $\mathcal{U}(\mathbf{Z}/\widehat{n}\mathbf{Z})$ (*cf.* Remark 1, or [CLT18, Lemma 4] for details on the choice of $\widehat{\mathcal{D}}$). The co-domain of function hash is $\mathit{\Pi} := \widehat{G}$, and:

$$\begin{aligned} \mathsf{hash} : \mathbf{Z} \times \widehat{G} &\to \widehat{G}. \\ (\mathsf{hk}, x) &\mapsto x^{\mathsf{hk}} \end{aligned}$$

Recall that ϖ denotes the group exponent of $\widehat{G}^p$. Functions $\widehat{\mathsf{projkg}}$ and projkg, which output values in $K_{\widehat{\mathsf{hp}}} := \mathbf{Z}/\varpi\mathbf{Z}$ and $K_{\mathsf{hp}} := G^p$ are defined as:

$$\begin{aligned} \widehat{\mathsf{projkg}} : \mathbf{Z} &\to \mathbf{Z}/\varpi\mathbf{Z} \\ \mathsf{hk} &\mapsto \mathsf{hk} \bmod \varpi \end{aligned} \quad \text{and} \quad \begin{aligned} \mathsf{projkg} : \mathbf{Z} &\to G^p. \\ \mathsf{hk} &\mapsto g_p^{\mathsf{hk}} \end{aligned}$$

As $\forall \mathsf{hk} \in \mathbf{Z}$, $\mathsf{projkg}(\mathsf{hk}) = g_p^{\widehat{\mathsf{projkg}}(\mathsf{hk})}$, projkg is a deterministic function of $\widehat{\mathsf{projkg}}$. For $\widehat{\mathsf{hp}} \in \mathbf{Z}/\varpi\mathbf{Z}$, $\widehat{x} \in \widehat{G}^p$ we define $\widehat{\mathsf{projhash}}(\widehat{\mathsf{hp}}, \widehat{x}) := \widehat{x}^{\widehat{\mathsf{hp}}}$. For $\mathsf{hp} \in K_{\mathsf{hp}}$, $(x, w) \in \mathsf{R}_{\mathsf{CL}}$, $\mathsf{projhash}(\mathsf{hp}, x, w)$ outputs hp^w. Clearly H_{CL} is correct, homomorphic and key homomorphic.

Example 2 – DDH**.** We define the PHF $\mathsf{H}_{\mathsf{DDH}}$ from $\mathcal{SM}_{\mathsf{DDH}}$ as follows. The hash key space is $K_{\mathsf{hk}} := (\mathbf{Z}/q\mathbf{Z})^2$. The hashkg algorithm samples $\mathsf{hk} \hookleftarrow \mathcal{U}((\mathbf{Z}/q\mathbf{Z})^2)$. The hash function has co-domain $\mathit{\Pi} := G$; and maps $\mathsf{hk} := (\kappa_0, \kappa_1) \in (\mathbf{Z}/q\mathbf{Z})^2$ and $(x_0, x_1) \in G^2$ to $x_0^{\kappa_0} x_1^{\kappa_1}$. Algorithm projkg outputs keys in $K_{\mathsf{hp}} := G$, and maps (κ_0, κ_1) to $g_0^{\kappa_0} g_1^{\kappa_1}$. For $\mathsf{hp} := g_0^{\kappa_0} g_1^{\kappa_1} \in G$, and $((x_0, x_1), w) \in \mathsf{R}_{\mathsf{DDH}}$, $\mathsf{projhash}(\mathsf{hp}, (x_0, x_1), w)$ outputs hp^w. It is clear that $\mathsf{H}_{\mathsf{DDH}}$ is correct, homomorphic and key homomorphic.

Extended projective hash functions. We also use *extended* projective hash functions (EPHF) [CS02]. These are defined as PHFs, only the hashing and projective hashing algorithms take an additional input from a finite set E.

Definition 9. *Consider a subset membership problem $\mathcal{SM} := (\widehat{\mathcal{X}}, \mathcal{X}, \widehat{\mathcal{L}}, \mathcal{W}, \mathsf{R})$ and an efficiently recognisable finite set E. An extended projective hash function for $\mathcal{SM}$ is a tuple $\mathsf{eH} := (\mathsf{ehashkg}, \widehat{\mathsf{eprojkg}}, \mathsf{eprojkg}, \mathsf{ehash}, \widehat{\mathsf{eprojhash}}, \mathsf{eprojhash})$, where:*

- $\mathsf{ehashkg}$ *is a PPTA which on input the description of $\mathcal{SM}$, outputs a hashing key ehk in some set K_{ehk};*
- $\widehat{\mathsf{eprojkg}}$ *is a deterministic algorithm which on input $\mathsf{ehk} \in K_{\mathsf{ehk}}$ outputs a projection key $\widehat{\mathsf{ehp}}$. The image of K_{ehk} through $\widehat{\mathsf{eprojkg}}$ is denoted $K_{\widehat{\mathsf{ehp}}}$;*
- $\mathsf{eprojkg}$ *is a DPTA which on input $\mathsf{ehk} \in K_{\mathsf{ehk}}$ outputs a public projection key ehp, such that for $\mathsf{ehk} \in K_{\mathsf{ehk}}$, ehp is a fixed deterministic function of the output of $\widehat{\mathsf{eprojkg}}(\mathsf{ehk})$. The image of K_{ehk} through $\mathsf{eprojkg}$ is denoted K_{ehp}.*
- ehash *is a DPTA which on input $\mathsf{ehk} \in K_{\mathsf{ehk}}$, $(x, e) \in \widehat{\mathcal{X}} \times E$ outputs the hash value $\mathsf{ehash}(\mathsf{hk}, x, e)$. The image of $\widehat{\mathcal{X}} \times E$ through ehash is called the set of hash values and is denoted Σ;*
- $\widehat{\mathsf{eprojhash}}$ *is a deterministic algorithm which on input $\widehat{\mathsf{ehp}} \in K_{\widehat{\mathsf{ehp}}}$, $x \in \widehat{\mathcal{L}}$, and $e \in E$ outputs the hash value $\widehat{\mathsf{eprojhash}}(\widehat{\mathsf{ehp}}, x, e)$ in Σ;*
- $\mathsf{eprojhash}$ *is a DPTA which on input $\mathsf{ehp} \in K_{\mathsf{ehp}}$, $x \in \mathcal{L}$, the corresponding witness $w \in \mathcal{W}$ and $e \in E$, outputs the hash value $\mathsf{projhash}(\mathsf{hp}, x, w, e)$ in Σ.*

Correctness holds if for any $\mathsf{ehk} \leftarrow \mathsf{ehashkg}(\mathcal{SM})$, $\widehat{\mathsf{ehp}} \leftarrow \widehat{\mathsf{eprojkg}}(\mathsf{ehk})$ it holds that $\forall (x, e) \in \widehat{\mathcal{L}} \times E$, $\widehat{\mathsf{eprojhash}}(\widehat{\mathsf{ehp}}, x, e) = \mathsf{ehash}(\mathsf{ehk}, x, e)$. And for $\mathsf{ehp} \leftarrow \mathsf{eprojkg}(\mathsf{ehk})$, it holds that $\forall (x, e) \in \mathcal{L} \times E$ and $w \in \mathcal{W}$, s.t. $(x, w) \in \mathsf{R}$, $\mathsf{eprojhash}(\mathsf{ehp}, x, w, e) = \widehat{\mathsf{eprojhash}}(\widehat{\mathsf{ehp}}, x, e)$.

Remark 6. The definitions of homomorphism and key homomorphism can be adapted to EPHFs in a straightforward way and must hold for any $e \in E$.

For our running examples and for our constructions of Sections 4 and 5 we use the generic construction of [CS02, Sec. 7.2] to build an EPHF from a PHF H and a CRHF Γ. The CRHF is required in order to attain the vector universality property defined in Def. 14: it ensures that one cannot compute a hash value $\mathsf{ehash}(\mathsf{hk}, x, e)$ from $\mathsf{ehash}(\mathsf{hk}, x^*, e^*)$ if $(x, e) \neq (x^*, e^*)$. Their generic construction is provided in Appendix B. If the underlying PHF is key homomorphic then the resulting EPHF is so[5]; and using the notations of Def. 6 and 9, one obtains $\Sigma = \Pi$, $K_{\mathsf{ehk}} = K_{\mathsf{hk}}^2$, $K_{\mathsf{ehp}} = K_{\mathsf{hp}}^2$ et $K_{\widehat{\mathsf{ehp}}} = K_{\widehat{\mathsf{hp}}}^2$.

Example 1 – $\mathsf{HSM}_{\mathsf{CL}}$ **.** Here $E := \widehat{G}$ and $\Gamma : \widehat{G}^2 \mapsto \{0, \ldots, p-1\}$ is sampled from a family of CRHF. The EPHF $\mathsf{eH}_{\mathsf{CL}}$ has hash key space $K_{\mathsf{ehk}} := \mathbf{Z}^2$; set of projection keys $K_{\widehat{\mathsf{ehp}}} := (\mathbf{Z}/\varpi\mathbf{Z})^2$; set of public projection keys $K_{\mathsf{ehp}} := G^2$; set of hash values $\Sigma := \widehat{G}$; and is defined from H_{CL} as:

[5] This is not the case for homomorphism.

- $\mathsf{ehashkg}$: sample $\mathsf{hk}_0 \hookleftarrow \widehat{\mathscr{D}}$; $\mathsf{hk}_1 \hookleftarrow \widehat{\mathscr{D}}$; and output $\mathsf{ehk} := (\mathsf{hk}_0, \mathsf{hk}_1)$
- $\mathsf{ehash}(\mathsf{hk}_0, \mathsf{hk}_1, x, e)$: let $\gamma \leftarrow \varGamma(x, e)$ and output $x^{\mathsf{hk}_0+\gamma\mathsf{hk}_1}$
- $\widehat{\mathsf{eprojkg}}(\mathsf{ehk})$: let $\widehat{\mathsf{ehp}} := (\widehat{\mathsf{hp}}_0, \widehat{\mathsf{hp}}_1) = (\mathsf{hk}_0 \bmod \varpi, \mathsf{hk}_1 \bmod \varpi)$; output $\widehat{\mathsf{ehp}}$.
- $\mathsf{eprojkg}(\mathsf{ehk})$: let $\mathsf{hp}_0 := g_p^{\mathsf{hk}_0}$; $\mathsf{hp}_1 := g_p^{\mathsf{hk}_1}$; output $\mathsf{ehp} := (\mathsf{hp}_0, \mathsf{hp}_1)$
- $\widehat{\mathsf{eprojhash}}((\widehat{\mathsf{hp}}_0, \widehat{\mathsf{hp}}_1), \widehat{x}, e)$, where $\widehat{x} \in \widehat{G}^p$: let $\gamma \leftarrow \varGamma(\widehat{x}, e)$; output $\widehat{x}^{\widehat{\mathsf{hp}}_0} \cdot (\widehat{x}^{\widehat{\mathsf{hp}}_1})^\gamma$
- $\mathsf{eprojhash}((\mathsf{hp}_0, \mathsf{hp}_1), x, w, e)$: let $\gamma \leftarrow \varGamma(x, e)$; output $(\mathsf{hp}_0 \cdot \mathsf{hp}_1^\gamma)^w$.

The EPHF $\mathsf{eH}_{\mathsf{CL}}$ is key homomorphic.

Example 2 – DDH**.** Here $E := G$ and $\varGamma : G^3 \mapsto \{0, \ldots, q-1\}$ is sampled from a family of CRHF. The EPHF $\mathsf{eH}_{\mathsf{DDH}}$ is defined from $\mathsf{H}_{\mathsf{DDH}}$ as:

- $\mathsf{ehashkg}$: sample $\kappa_0, \kappa_1, \kappa_2, \kappa_3 \hookleftarrow \mathcal{U}(\mathbf{Z}/q\mathbf{Z})$, and output $\mathsf{ehk} := (\kappa_0, \kappa_1, \kappa_2, \kappa_3)$ s.t. $K_{\mathsf{ehk}} := (\mathbf{Z}/q\mathbf{Z})^4$.
- $\mathsf{ehash}(\mathsf{ehk}, (x_0, x_1), e)$: let $\gamma \leftarrow \varGamma(x_0, x_1, e)$; output $\mathsf{hash}((\kappa_0, \kappa_1), (x_0, x_1)) \cdot \mathsf{hash}((\kappa_2, \kappa_3), (x_0, x_1))^\gamma = x_0^{\kappa_0+\gamma\kappa_2} x_1^{\kappa_1+\gamma\kappa_3}$ s.t. $\varSigma := G$.
- $\mathsf{eprojkg}(\mathsf{ehk})$: let $\mathsf{hp}_0 := g_0^{\kappa_0} g_1^{\kappa_1}$; $\mathsf{hp}_1 := g_0^{\kappa_2} g_1^{\kappa_3}$; output $\mathsf{ehp} := (\mathsf{hp}_0, \mathsf{hp}_1)$; s.t. $K_{\mathsf{ehp}} := G^2$.
- $\mathsf{eprojhash}(\mathsf{hp}_0, \mathsf{hp}_1, (x_0, x_1), w, e)$ where $x = g_0^w$ and $x_1 = g_1^w$: compute $\gamma \leftarrow \varGamma(x_0,\, x_1, e)$; output $(\mathsf{hp}_0 \cdot \mathsf{hp}_1^\gamma)^w = x_0^{\kappa_0+\gamma\kappa_2} x_1^{\kappa_1+\gamma\kappa_3}$.

The EPHF $\mathsf{eH}_{\mathsf{DDH}}$ is key homomorphic.

3 Building IPFE from PHFs

For correctness of our constructions, we first introduce compatibility properties for the underlying PHFs. For security we define two new properties: *vector smoothness* and *vector universality.* If the PHF used for confidentiality is *vector smooth*, one can build ind-fe-cpa-secure IPFE schemes. To attain ind-fe-cca-security, the PHF used to ensure ciphertext integrity must be *vector universal.*

3.1 Compatibility properties

To build IPFE from a PHF, one needs the PHF to be *compatible* with the ring in which inner products are computed; one also needs to impose restrictions on the message space $\mathcal{M}$ and the space $\mathcal{K}$ from which decryption keys are derived.

Definition 10 (ipfe-compatibility). *Let $\mathcal{R}$ be a ring, either $\mathbf{Z}$ or $\mathbf{Z}/q\mathbf{Z}$ for some prime q. Let $\mathcal{SM}$ be a subgroup membership problem, and consider the associated PHF* H*. One says* H *is* $(\mathcal{R}, a, f, \aleph, \ell, \mathcal{M}, \mathcal{K})$-ipfe-*compatible if:*

- *the hash key space is $K_{\mathsf{hk}} := \mathcal{R}^a$ for some positive integer a;*
- H *is key homomorphic, where the (additive) group operation associated to K_{hk} is the addition of $\mathcal{R}$ performed point-wise;*
- *the co-domain $\varPi$ of* hash *is a finite Abelian group which contains a cyclic subgroup F, generated by f, of order $\aleph$;*
- *if $\mathcal{R} = \mathbf{Z}/q\mathbf{Z}$ then $F = \varPi$ is of prime order $\aleph = q$;*
- *$\mathcal{M}$ and $\mathcal{K}$ are efficiently recognisable subsets of $\mathcal{R}^\ell$, for a positive integer ℓ;*

- *there exists an efficient algorithm* $\log_f$ *which, for all* $\boldsymbol{m} \in \mathcal{M}$*,* $\boldsymbol{k} \in \mathcal{K}$*, computes* $\log_f(f^{\langle \boldsymbol{m},\boldsymbol{k}\rangle}) = \langle \boldsymbol{m},\boldsymbol{k}\rangle \in \mathcal{R}$*.*

Remark 7. An EPHF built from a $(\mathcal{R}, a, f, \aleph, \ell, \mathcal{M}, \mathcal{K})$-$\mathsf{ipfe}$-compatible PHF via the generic construction of [CS02, Sec. 7] is $(\mathcal{R}, 2a, f, \aleph, \ell, \mathcal{M}, \mathcal{K})$-$\mathsf{ipfe}$-compatible.

Remark 8. In our examples and instantiations (*cf.* Appx. D), we set $\mathcal{M} = \mathcal{K}$ to simplify presentation. One can choose different subsets of $\mathcal{R}^\ell$ to fit applications.

Notation. For a message space $\mathcal{M}$, we denote:

$$\varDelta\mathcal{M} := \{\boldsymbol{x}_0 - \boldsymbol{x}_1 \mid \boldsymbol{x}_0 \neq \boldsymbol{x}_1 \in \mathcal{M}\}.$$

Example 1 – $\mathsf{HSM}_{\mathsf{CL}}$**.** Here $\mathcal{R} := \mathbf{Z}$ and $a := 1$ since $K_{\mathsf{hk}} := \mathbf{Z}$. The co-domain of hash is $\widehat{G}$ which is a finite Abelian group, and F is a cyclic subgroup of $\widehat{G}$ of prime order $\aleph := p$, generated by f. We set $\mathcal{M} = \mathcal{K} = \{\boldsymbol{x} \in \mathbf{Z}^\ell : ||\boldsymbol{x}||_\infty < \sqrt{\frac{p}{2\ell}}\}$. For $\boldsymbol{m} \in \mathcal{M}$, $\boldsymbol{k} \in \mathcal{K}$, since $||\boldsymbol{m}||_\infty$ and $||\boldsymbol{k}||_\infty < \sqrt{\frac{p}{2\ell}}$, it holds that $-p/2 < \langle \boldsymbol{m},\boldsymbol{k}\rangle < p/2$. Algorithm $\log_f$ first uses the Solve algorithm of Sec. 2.2 to compute $\mathsf{sol} \leftarrow \mathsf{Solve}(f^{\langle \boldsymbol{m},\boldsymbol{k}\rangle})$, then if $\mathsf{sol} \geqslant p/2$, it returns $(\mathsf{sol} - p)$, otherwise it returns sol. With this implementation $\log_f(f^{\langle \boldsymbol{m},\boldsymbol{k}\rangle}) = \langle \boldsymbol{m},\boldsymbol{k}\rangle$ in $\mathbf{Z}$.

Example 2 – DDH**.** Here $\mathcal{R} := \mathbf{Z}/q\mathbf{Z}$ and $a := 2$ since $K_{\mathsf{hk}} := (\mathbf{Z}/q\mathbf{Z})^2$. The co-domain of hash is the cyclic group $G = \langle g \rangle$ of prime order q. Thus we set $f := g$, which generates $F := G$ and $\aleph := q$. This implies that the algorithm $\log_f$ is the discrete logarithm in G. Note that in a DDH group the DL problem is hard by assumption, so $\log_f$ is only efficient for small input values. Thus $\mathcal{M}$ and $\mathcal{K}$ are subsets of $(\mathbf{Z}/q\mathbf{Z})^\ell$ s.t. $\forall \boldsymbol{m} \in \mathcal{M}$, $\boldsymbol{k} \in \mathcal{K}$, $\log_g(g^{\langle \boldsymbol{m},\boldsymbol{k}\rangle}) = \langle \boldsymbol{m},\boldsymbol{k}\rangle \in \mathbf{Z}/q\mathbf{Z}$ is computable in time $\mathsf{poly}(\lambda)$.

3.2 Decomposability

We introduce the notion of a decomposable PHF, this property allows us to have a clear separation between the part of a given hash value which is predictable (whose pre-image is in $\widehat{\mathcal{L}}$), and the part which appears random. Though the definition is new, many well known PHFs arising from groups satisfy this property (e.g. the original DDH and DCR based PHFs of [CS02]).

Definition 11. *Let* $\mathcal{SM} := (\widehat{\mathcal{X}}, \mathcal{X}, \widehat{\mathcal{L}}, \mathcal{W}, \mathsf{R})$ *be an SMP, and* H *the associated PHF. We say that* H *is* $(\widehat{\varUpsilon}, \varUpsilon, F)$*-decomposable if the co-domain* $\varPi$ *of* hash *is a finite Abelian group containing a cyclic subgroup* F*, and there exist* $\widehat{\varUpsilon} \in \widehat{\mathcal{X}}$ *and* $\varUpsilon \in \mathcal{X}$ *s.t.:*
- $\mathcal{X} \simeq \mathcal{L} \times \langle \varUpsilon \rangle$*, and* $\widehat{\mathcal{X}} \simeq \widehat{\mathcal{L}} \times \langle \widehat{\varUpsilon} \rangle$*;*
- $\forall \mathsf{hk} \in K_{\mathsf{hk}}$*,* $\mathsf{hash}(\mathsf{hk}, \varUpsilon) \in F$*,* $\mathsf{hash}(\mathsf{hk}, \widehat{\varUpsilon}) \in F$*.*

If $\widehat{\mathcal{X}} = \mathcal{X}$ *or* $\widehat{\varUpsilon} = \varUpsilon$*, we simply say* H *is* $(\varUpsilon, F)$*-decomposable.*

Remark 9. If H is decomposable, we sample $\mathcal{X} \backslash \mathcal{L}$ by sampling $(\tilde{x}, \tilde{w}) \leftarrow \mathsf{R}$, $\upsilon \hookleftarrow \mathbf{Z}/\aleph\mathbf{Z}$, $\upsilon \neq 0$, and outputting $x := \varUpsilon^\upsilon \tilde{x}$. We denote this sampling $x \hookleftarrow \mathcal{X} \backslash \mathcal{L}$.

Remark 10. An EPHF built from a $(\widehat{\Upsilon}, \Upsilon, F)$-decomposable PHF via the generic construction of [CS02, Sec. 7.2] is also $(\widehat{\Upsilon}, \Upsilon, F)$-decomposable.

Example 1 – $\mathsf{HSM}_{\mathsf{CL}}$ **.** By definition $\widehat{G} \simeq \widehat{G}^p \times F$ and $G \simeq G^p \times F$, where $F = \langle f \rangle$. Moreover $\forall \mathsf{hk} \in \mathbf{Z}$, $\mathsf{hash}(\mathsf{hk}, f) = f^{\mathsf{hk}} \in F$, and for $(\mathsf{hk}_0, \mathsf{hk}_1) \in \mathbf{Z}^2$, $\mathsf{ehash}((\mathsf{hk}_0, \mathsf{hk}_1), f, e) = f^{\mathsf{hk}_0 + e\mathsf{hk}_1} \in F$. Thus we set $\widehat{\Upsilon} := \Upsilon := f$, s.t. H_{CL} and $\mathsf{eH}_{\mathsf{CL}}$ are (f, F)-decomposable.

Example 2 – DDH**.** The group G is cyclic, so we take $F := G$, and $\Upsilon := (1, g_1)$. It holds that $G^2 \simeq \langle (1, g_1) \rangle \times \langle (g_0, g_1) \rangle$; clearly $\forall \mathsf{hk} \in (\mathbf{Z}/q\mathbf{Z})^2$, $\mathsf{hash}(\mathsf{hk}, (1, g_1)) \in G$. Thus $\mathsf{H}_{\mathsf{DDH}}$ and $\mathsf{eH}_{\mathsf{DDH}}$ are $((1, g_1), G)$-decomposable.

3.3 Associated matrix

We here define the notion of a matrix $\mathbf{B}_{\boldsymbol{m}}$ *associated* to a vector $\boldsymbol{m}$. In our upcoming constructions, $\boldsymbol{m}$ will be the difference between the two challenge message vectors. As such, valid adversaries can request decryption keys associated to vectors $\boldsymbol{k} \in \mathcal{K}$ satisfying $\boldsymbol{k} \in \boldsymbol{m}^{\perp}$. The matrix $\mathbf{B}_{\boldsymbol{m}}$ is constructed in such a way that any such $\boldsymbol{k}$ can be written as a linear combination of the top $\ell - 1$ rows of $\mathbf{B}_{\boldsymbol{m}}$. Conversely, any $\boldsymbol{k} \notin \boldsymbol{m}^{\perp}$ – for which a decryption key trivially reveals which of the challenge messages was encrypted – has some contribution from the last row of $\mathbf{B}_{\boldsymbol{m}}$. Our protocols' secret values, when projected onto this last row, must conserve sufficient entropy for security to hold. Defining the exact properties required of this matrix for all our proofs to go through is an essential point to attaining genericity. Capturing these properties in the following definition significantly improves readability of our theorems and proofs thereof.

Definition 12. *Let $\mathcal{R}$ be either the ring $\mathbf{Z}$ or $\mathbf{Z}/q\mathbf{Z}$ for some prime q; $\ell \in \mathbf{N}$; and $\boldsymbol{m} \in \mathcal{R}^{\ell}$. We say $\mathbf{B}_{\boldsymbol{m}} \in \mathcal{R}^{\ell \times \ell}$ is* a matrix associated to $\boldsymbol{m}$ *if, denoting $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_\ell)$ the rows of $\mathbf{B}_{\boldsymbol{m}}$ it holds that: (1) $\mathbf{B}_{\boldsymbol{m}}$ is invertible mod $\aleph$; (2) $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{\ell-1})$ form a basis of $\boldsymbol{m}^{\perp}$; (3) $\boldsymbol{b}_\ell \notin \boldsymbol{m}^{\perp}$ and if $\mathcal{R} = \mathbf{Z}$ then $\boldsymbol{b}_\ell = \boldsymbol{m}$.*

Lemma 1 states conditions to efficiently build $\mathbf{B}_{\boldsymbol{m}}$. A detailed proof, which follows immediately from proofs in [ALS16, CLT18], is provided in Appx. C.

Lemma 1. *Let $\mathcal{R}$ be either the ring $\mathbf{Z}$ or $\mathbf{Z}/q\mathbf{Z}$ for a prime q; ℓ and a be positive integers; and H be an $(\mathcal{R}, a, f, \aleph, \ell, \mathcal{M}, \mathcal{K})$-$\mathsf{ipfe}$-compatible projective hash function, where $\aleph$ is either prime or hard to factor. From any $\boldsymbol{m} \in \Delta\mathcal{M}$ one can efficiently and deterministically construct a matrix $\mathbf{B}_{\boldsymbol{m}} \in \mathcal{R}^{\ell \times \ell}$ associated to $\boldsymbol{m}$.*

Notation. As Lemma 1 builds $\mathbf{B}_{\boldsymbol{m}}$ *deterministically* from $\boldsymbol{m}$, one can build – from a random variable M taking values in $\Delta\mathcal{M}$ – the matrix of random variables $\mathbf{B}_M$. We will use this notation in our definitions and proofs. We denote $\boldsymbol{b}_1^M, \ldots, \boldsymbol{b}_\ell^M$ the rows of $\mathbf{B}_M$.

3.4 Confidentiality

The notion of smoothness, defined in [CS02], ensures confidentiality given a PKE scheme's public parameters. In the context of IPFE, one must also deal with key derivation queries performed by the adversary $\mathcal{A}$. Here the master secret key msk is a vector of ℓ hash keys of which $\mathcal{A}$ can request linear combinations. The first property we introduce – vector smoothness – ensures confidentiality given this extra leakage of information. Precisely, vector smoothness ensures that given the projection of msk on a hyperplane $\mathcal{H}$, its projection onto a line orthogonal to $\mathcal{H}$ remains uniformly distributed. This latter projection masks the challenge bit in our constructions. This new property captures the techniques used to build $\mathsf{ind\text{-}fe\text{-}cpa}$-secure IPFE schemes in [ALS16] from DDH and DCR, and later in [CLT18] from class group based assumptions. Hence the proofs of Lemmas 3 and 4 resemble the security proofs of [CLT18, Thm. 7] and [ALS16, Thm. 1].

Definition 13 (δ_{vs}-vector smooth over $\mathcal{X}$ on F). *Let $\mathcal{R}$ be a ring, either $\mathbf{Z}$ or $\mathbf{Z}/q\mathbf{Z}$ for some prime q; ℓ and a be positive integers; $\mathcal{SM} := (\widehat{\mathcal{X}}, \mathcal{X}, \widehat{\mathcal{L}}, \mathcal{W}, \mathsf{R})$ an SMP; and H the associated PHF which we assume to be $(\mathcal{R}, a, f, \aleph, \ell, \mathcal{M}, \mathcal{K})$-$\mathsf{ipfe}$-compatible. For $i \in [\ell]$, let $\mathsf{hk}_i \leftarrow \mathsf{hashkg}(\mathcal{SM})$, and $\mathbf{hk} := (\mathsf{hk}_1, \ldots, \mathsf{hk}_\ell)$. Consider a random variable M taking values in $\Delta\mathcal{M}$ and the associated matrix $\mathbf{B}_M \in \mathcal{R}^{\ell\times\ell}$. Let $X \hookleftarrow \mathcal{X}\backslash\mathcal{L}$, and $Y \hookleftarrow \mathcal{U}(F)$. Then H is $\delta_{vs}(\ell)$-vector smooth over $\mathcal{X}$ on F if the following tuples of random variables are $\delta_{vs}(\ell)$-close:*

$$\big\{M, X, \{\widehat{\mathsf{projkg}}(\mathsf{hk}_i)\}_{i\in[\ell]}, \{\lessdot \mathbf{hk}, \boldsymbol{b}_j^M \gtrdot_a\}_{j\in[\ell-1]}, \mathsf{hash}(\lessdot \mathbf{hk}, \boldsymbol{b}_\ell^M \gtrdot_a, X)\cdot Y\big\}$$
$$\text{and } \big\{M, X, \{\widehat{\mathsf{projkg}}(\mathsf{hk}_i)\}_{i\in[\ell]}, \{\lessdot \mathbf{hk}, \boldsymbol{b}_j^M \gtrdot_a\}_{j\in[\ell-1]}, \mathsf{hash}(\lessdot \mathbf{hk}, \boldsymbol{b}_\ell^M \gtrdot_a, X)\big\}.$$

Lemma 2 is a convenient reformulation of vector smoothness for PHFs which have homomorphic properties and are decomposable.

Lemma 2. *Assume H is further homomorphic, key homomorphic and $(\widehat{\Upsilon}, \Upsilon, F)$-decomposable. Since $X \in \mathcal{X}\backslash\mathcal{L}$, there exist unique $(X_{\mathcal{L}}, W) \in \mathsf{R}$ and $X_\Upsilon \in \langle\Upsilon\rangle$ s.t. $X = X_{\mathcal{L}} \cdot X_\Upsilon$. Then H is δ_{vs}-vector smooth over $\mathcal{X}$ on F if and only if the following tuples of random variables are δ_{vs}-close:*

$$\big\{M, X, \{\widehat{\mathsf{projkg}}(\mathsf{hk}_i)\}_{i\in[\ell]}, \{\lessdot \mathbf{hk}, \boldsymbol{b}_i^M \gtrdot_a\}_{i\in[\ell-1]}, Y\big\} \quad \text{and}$$
$$\big\{M, X, \{\widehat{\mathsf{projkg}}(\mathsf{hk}_i)\}_{i\in[\ell]}, \{\lessdot \mathbf{hk}, \boldsymbol{b}_i^M \gtrdot_a\}_{i\in[\ell-1]}, \mathsf{hash}(\lessdot \mathbf{hk}, \boldsymbol{b}_\ell^M \gtrdot_a, X_\Upsilon)\big\}.$$

Proof. For fixed $\boldsymbol{m} \in \Delta\mathcal{M}$, $(x, w) \in \mathsf{R}$, $y \in \langle\Upsilon\rangle$, $\widehat{\mathsf{hp}}_i \in K_{\widehat{\mathsf{hp}}}$ (which in turn fixes $\mathsf{hp}_i \in K_{\mathsf{hp}}$) for $i \in [\ell]$, and $v_j \in \mathcal{R}^a$ for $j \in [\ell-1]$, the first three coordinates of the considered tuples fix $M = \boldsymbol{m}$; $X_{\mathcal{L}} = x$; $X_\Upsilon = y$; $\widehat{\mathsf{hp}}_i = \widehat{\mathsf{projkg}}(\mathsf{hk}_i)$; $\mathsf{hp}_i := \mathsf{projkg}(\mathsf{hk}_i)$; and $\lessdot \mathbf{hk}, \boldsymbol{b}_j^M \gtrdot_a = v_j$. Hence they fix the value of $\mathsf{hash}(\mathsf{hk}_i, X_{\mathcal{L}}) = \mathsf{projhash}(\mathsf{hp}_i, x, w)$ for $i \in [\ell]$.

Let us denote $\mathbf{B}_{\boldsymbol{m}}$ the matrix associated to $\boldsymbol{m}$ built as per Lemma 1, and its rows $\boldsymbol{b}_1^{\boldsymbol{m}}, \ldots, \boldsymbol{b}_\ell^{\boldsymbol{m}}$. Since $M = \boldsymbol{m}$, it holds that $\mathbf{B}_M = \mathbf{B}_{\boldsymbol{m}}$. Now using the key homomorphism of H we see that $\mathsf{hash}(\lessdot \mathbf{hk}, \boldsymbol{b}_\ell^M \gtrdot_a, X_{\mathcal{L}}) = \mathsf{hash}(\lessdot \mathbf{hk}, \boldsymbol{b}_\ell^{\boldsymbol{m}} \gtrdot_a, x)$. Furthermore, from the homomorphism of H it holds that $\mathsf{hash}(\lessdot \mathbf{hk}, \boldsymbol{b}_\ell^M \gtrdot_a, X) = \mathsf{hash}(\lessdot \mathbf{hk}, \boldsymbol{b}_\ell^{\boldsymbol{m}} \gtrdot_a, x) \cdot \mathsf{hash}(\lessdot \mathbf{hk}, \boldsymbol{b}_\ell^M \gtrdot_a, X_\Upsilon)$. It is now clear that the statistical distance between the two tuples of random variables in Lemma 2 is equal to the statistical distance between those of Definition 13. □

Example 1 – $\mathsf{HSM}_{\mathsf{CL}}$ **.** Lemma 3 is inspired by [CLT18, Thm. 7] and states sufficient conditions for H_{CL} to be vector smooth.

Lemma 3. *If the* hashkg *algorithm of* H_{CL} *samples hashing keys from the Gaussian distribution* $\mathcal{D}_{\mathbf{Z},\sigma}$ *for* $\sigma > \tilde{s}p^{3/2}\sqrt{|\log_2(\delta_{vs})|}$*, then* H_{CL} *is* δ_{vs}*-vector-smooth over* G *on* F*.*

Proof. Recall that $\mathcal{R} = \mathbf{Z}$ and for some $\ell \in \mathbf{N}$, $\mathcal{M} = \mathcal{K} = \{\boldsymbol{x} \in \mathbf{Z}^\ell : ||\boldsymbol{x}||_\infty < \sqrt{\frac{p}{2\ell}}\}$. For $i \in [\ell]$, let hk_i denote independent random variables following the distribution $\mathcal{D}_{\mathbf{Z},\sigma}$; let $\mathbf{hk} := (\mathsf{hk}_1, \ldots, \mathsf{hk}_\ell) \in \mathbf{Z}^\ell$. Consider a random variable M taking values in $\Delta\mathcal{M}$ and the associated matrix $\mathbf{B}_M \in \mathbf{Z}^{\ell\times\ell}$.

For $X \hookleftarrow G\backslash G^p$, there exist unique $\alpha \in \mathbf{Z}/s\mathbf{Z}$ and $\beta \in (\mathbf{Z}/p\mathbf{Z})^*$ s.t. $X = g_p^\alpha f^\beta$. As noted in Lemma 2, for $\gamma \hookleftarrow \mathcal{U}(\mathbf{Z}/p\mathbf{Z})$, we need to evaluate the statistical distance between:

$$\mathcal{U} = \Big\{M, g_p^\alpha f^\beta, \{\widehat{\mathsf{projkg}}(\mathsf{hk}_i)\}_{i\in[\ell]}, \{\langle \mathbf{hk}, \boldsymbol{b}_j^M\rangle\}_{j\in[\ell-1]}, f^\gamma\Big\}$$
$$\text{and } \ \mathcal{V} = \Big\{M, g_p^\alpha f^\beta, \{\widehat{\mathsf{projkg}}(\mathsf{hk}_i)\}_{i\in[\ell]}, \{\langle \mathbf{hk}, \boldsymbol{b}_j^M\rangle\}_{j\in[\ell-1]}, f^{\beta\langle \mathbf{hk}, \boldsymbol{b}_\ell^M\rangle}\Big\}.$$

Consider $\boldsymbol{m} \in \Delta\mathcal{M}$, $\alpha_0 \in \mathbf{Z}/s\mathbf{Z}$, $\beta_0 \in \mathbf{Z}/p\mathbf{Z}$, $\widehat{\mathbf{hp}} \in (\mathbf{Z}/\varpi\mathbf{Z})^\ell$, and $\boldsymbol{v} \in \mathbf{Z}^{\ell-1}$. It suffices to study the distance between the random variables $Y := \beta\langle \mathbf{hk}, \boldsymbol{b}_\ell^M\rangle \bmod p$ and γ conditioned on the conjunction of the following events: $M = \boldsymbol{m}$; $\alpha \bmod s = \alpha_0 \bmod s$; $\beta \bmod p = \beta_0 \bmod p$; $\mathbf{hk} \bmod \varpi = \widehat{\mathbf{hp}} \bmod \varpi$ and for $j \in [\ell-1]$, $\langle \mathbf{hk}, \boldsymbol{b}_j^M\rangle = v_j$.

In the following, we evaluate the distribution followed by $\mathbf{hk}$ in $\mathbf{Z}$, conditioned on these events. Let $\mathbf{hk}_0$ denote an arbitrary vector satisfying the same equations as $\mathbf{hk}$, *i.e.* for $j \in [\ell-1]$, $\langle \mathbf{hk}_0, \boldsymbol{b}_j^{\boldsymbol{m}}\rangle = v_j$ in $\mathbf{Z}$, and $\mathbf{hk}_0 \bmod \varpi = \widehat{\mathbf{hp}} \bmod \varpi$. We define $\Lambda := \{\boldsymbol{t} \in \mathbf{Z}^\ell \mid \langle \boldsymbol{t}, \boldsymbol{b}_j^{\boldsymbol{m}}\rangle = 0 \text{ for } j \in [\ell-1]; \boldsymbol{t} = \mathbf{0} \bmod \varpi\} \subset \mathbf{Z}^\ell$. Since $\mathbf{hk}$ is sampled from $\mathcal{D}_{\mathbf{Z}^\ell,\sigma}$, given the fixed information, $\mathbf{hk}$ is of the form $\mathbf{hk}_0 + T$ where T is a random variable with values in Λ, and which follows the same probability distribution as $\mathbf{hk} - \mathbf{hk}_0$ but taken over Λ, *i.e.* $\forall \boldsymbol{t} \in \Lambda$:

$$\Pr[T = \boldsymbol{t}] = \frac{\mathcal{D}_{Z^\ell,\sigma,-\mathbf{hk}_0}(\boldsymbol{t})}{\mathcal{D}_{Z^\ell,\sigma,-\mathbf{hk}_0}(\Lambda)} = \frac{\rho_{\sigma,-\mathbf{hk}_0}(\boldsymbol{t})}{\rho_{\sigma,-\mathbf{hk}_0}(\mathbf{Z}^\ell)} \times \frac{\rho_{\sigma,-\mathbf{hk}_0}(\mathbf{Z}^\ell)}{\rho_{\sigma,-\mathbf{hk}_0}(\Lambda)} = \mathcal{D}_{\Lambda,\sigma,-\mathbf{hk}_0}(\boldsymbol{t}).$$

So the conditional distribution followed by $\mathbf{hk}$ is $\mathbf{hk}_0 + \mathcal{D}_{\Lambda,\sigma,-\mathbf{hk}_0}$.

Now denoting $d \neq 0$ the gcd of the coefficients of $\boldsymbol{b}_\ell^{\boldsymbol{m}}$, and $\tilde{\boldsymbol{b}} = 1/d \cdot \boldsymbol{b}_\ell^{\boldsymbol{m}} \in \mathbf{Z}^\ell$, it holds that for $1 \leqslant j \leqslant \ell - 1$, $\boldsymbol{b}_j^{\boldsymbol{m}} \in \tilde{\boldsymbol{b}}^\perp$. We consider the 1-dimensional lattice $\Lambda' := \{\boldsymbol{t} \in \mathbf{Z}^\ell | \langle \boldsymbol{t}, \boldsymbol{b}_i^{\boldsymbol{m}}\rangle = 0 \text{ for } i \in [\ell-1]\}$ which contains $\tilde{\boldsymbol{b}}\mathbf{Z}$. In fact as $\gcd(\tilde{b}_1, \ldots, \tilde{b}_\ell) = 1$, one has $\Lambda' = \tilde{\boldsymbol{b}}\mathbf{Z}$ (there exists $\boldsymbol{y} \in \mathbf{Z}^\ell$ s.t. $\Lambda' = \boldsymbol{y} \cdot \mathbf{Z}$, and $\tilde{\boldsymbol{b}} = \alpha\boldsymbol{y}$, so α must divide $\gcd(\tilde{b}_1, \ldots, \tilde{b}_\ell) = 1$). Moreover $\Lambda = \Lambda' \cap \varpi \cdot \mathbf{Z}^\ell = (\tilde{\boldsymbol{b}} \cdot \mathbf{Z} \cap \varpi \cdot \mathbf{Z}^\ell) = \varpi \cdot \tilde{\boldsymbol{b}} \cdot \mathbf{Z}$, since $\forall \mu \in \mathbf{Z}$, in order for ϖ to divide each $\mu\tilde{b}_i$, ϖ must divide $\mu \cdot \gcd(\tilde{b}_1, \ldots, \tilde{b}_\ell) = \mu$. We now consider the distribution of $\langle \mathbf{hk}, \tilde{\boldsymbol{b}}\rangle$, and reduce it mod p, so as to prove that the random variable $\tilde{Y} := \langle \mathbf{hk}, \tilde{\boldsymbol{b}}\rangle \bmod p$

follows a distribution close to $\mathcal{U}(\mathbf{Z}/p\mathbf{Z})$. Let us denote $\Lambda_0 := \varpi \cdot ||\tilde{\boldsymbol{b}}||_2^2 \cdot \mathbf{Z}$. It follows from Lemma 10 that the distribution followed by $\langle \mathsf{\mathbf{hk}}, \tilde{\boldsymbol{b}} \rangle$ is:

$$\langle \mathsf{\mathbf{hk}}_0, \tilde{\boldsymbol{b}} \rangle + \mathcal{D}_{\Lambda_0, ||\tilde{\boldsymbol{b}}||_2 \cdot \sigma, -c} \text{ where } c := \langle \mathsf{\mathbf{hk}}_0, \tilde{\boldsymbol{b}} \rangle \text{ in } \mathbf{Z}.$$

In order to prove that the above distribution, taken mod p, is statistically close to $\mathcal{U}(\mathbf{Z}/p\mathbf{Z})$, we consider the distribution obtained by reducing the distribution $\mathcal{D}_{\Lambda_0, ||\tilde{\boldsymbol{b}}||_2 \cdot \sigma, -c}$ over Λ_0 modulo the sub-lattice $\Lambda_0' := p\Lambda_0$. Since ϖ and p are co-prime, it holds that $\Lambda_0/\Lambda_0' \simeq \mathbf{Z}/p\mathbf{Z}$, so demonstrating that $\langle \mathsf{\mathbf{hk}}, \tilde{\boldsymbol{b}} \rangle \bmod p$ follows a distribution statistically close to $\mathcal{U}(\Lambda_0/\Lambda_0')$ will allow us to conclude. From Lemma 11 it follows that to achieve the required smoothing parameter $\eta_\epsilon(\Lambda_0')$ one must impose a lower bound on the standard deviation σ: we need $||\tilde{\boldsymbol{b}}||_2 \cdot \sigma > \eta_\epsilon(\Lambda_0')$. From [MR07] we know that, for $0 < \epsilon < 1/2$, and setting $\delta_{vs} := 2\epsilon$, we have:

$$\eta_\epsilon(\Lambda_0') \leqslant (\ln(2(1+1/\epsilon))\pi^{-1})^{1/2} \cdot \lambda_1(\Lambda_0') < (2^{-1}|\log_2(\delta_{vs})|)^{1/2} \cdot \lambda_1(\Lambda_0')$$

Since $\lambda_1(\Lambda_0') = p \cdot \varpi \cdot ||\tilde{\boldsymbol{b}}||_2^2 < p \cdot \widehat{s} \cdot ||\tilde{\boldsymbol{b}}||_2^2$, we require $\sigma > p \cdot \widehat{s} \cdot ||\tilde{\boldsymbol{b}}||_2 \sqrt{2^{-1}|\log_2(\delta^{1/p})|}$. Moreover, as $||\tilde{\boldsymbol{b}}||_2 < \sqrt{2p}$ (due to the norm bounds on vectors in $\mathcal{M}$, one has $||\tilde{\boldsymbol{b}}||_\infty < 2\sqrt{p/(2\ell)}$), choosing $\sigma > \widehat{s} \cdot p^{3/2} \ \sqrt{|\log_2(\delta^{1/p})|}$ suffices to ensure that the distribution of $\langle \mathsf{\mathbf{hk}}, \tilde{\boldsymbol{b}} \rangle \bmod p$ is $\delta^{1/p}$-close to the uniform distribution over $\Lambda_0/\Lambda_0' \simeq \mathbf{Z}/p\mathbf{Z}$. We denote $\delta_{vs} := \delta^{1/p}$.

Finally $Y = \beta \cdot \langle \mathsf{\mathbf{hk}}, \boldsymbol{b}_\ell^{\boldsymbol{m}} \rangle \bmod p = \beta \cdot d \cdot \langle \mathsf{\mathbf{hk}}, \tilde{\boldsymbol{b}} \rangle \bmod p$ where $\langle \mathsf{\mathbf{hk}}, \tilde{\boldsymbol{b}} \rangle \bmod p$ is δ_{vs}-close to $\mathcal{U}(\mathbf{Z}/p\mathbf{Z})$, $\beta \neq 0 \bmod p$ and $d \neq 0 \bmod p$. This implies that Y also follows a distribution δ_{vs}-close to $\mathcal{U}(\mathbf{Z}/p\mathbf{Z})$. Thus the statistical distance between the last coordinates of $\mathcal{U}$ and $\mathcal{V}$, given the first four, is at most δ_{vs}, which concludes the proof. □

Example 2 – DDH**.** Lemma 4 is inspired by [ALS16, Thm. 1] and states sufficient conditions for $\mathsf{H}_{\mathsf{DDH}}$ to be vector smooth.

Lemma 4. *The* $\mathsf{H}_{\mathsf{DDH}}$ *projective hash function is* 0*-vector smooth over* G *on* G.

Proof. Recall that $\mathcal{R} = \mathbf{Z}/q\mathbf{Z}$, $a = 2$ and $\mathcal{M} = \mathcal{K} = \mathbf{Z}/q\mathbf{Z}$. For $i \in [\ell]$, let $\mathsf{hk}_i := (\kappa_{0,i}, \kappa_{1,i})$ denote independent random variables sampled from $\mathcal{U}((\mathbf{Z}/q\mathbf{Z})^2)$; let $\boldsymbol{\kappa}_0 := (\kappa_{0,1}, \ldots, \kappa_{0,\ell})$, $\boldsymbol{\kappa}_1 := (\kappa_{1,1}, \ldots, \kappa_{1,\ell})$, and $\mathsf{\mathbf{hk}} := (\mathsf{hk}_1, \ldots, \mathsf{hk}_\ell)$. Consider a random variable M taking values in $\Delta\mathcal{M}$ and the associated matrix $\mathbf{B}_M$. For $X \hookleftarrow \mathcal{X} \backslash \mathcal{L}$, there exist unique $\alpha \in \mathbf{Z}/q\mathbf{Z}$ and $\beta \in (\mathbf{Z}/q\mathbf{Z})^*$ s.t. $X = (g_0, g_1)^\alpha \odot (1, g_1)^\beta$. As noted in Lemma 2, for $\gamma \hookleftarrow \mathcal{U}(\mathbf{Z}/q\mathbf{Z})$, we need to evaluate the distance between:

$$\mathcal{U} = \Big\{ M, (g_0^\alpha, g_1^{\alpha+\beta}), \{g_0^{\kappa_{0,i}} g_1^{\kappa_{1,i}}\}_{i \in [\ell]}, \{\langle \boldsymbol{\kappa}_0, \boldsymbol{b}_j^M \rangle, \langle \boldsymbol{\kappa}_1, \boldsymbol{b}_j^M \rangle\}_{j \in [\ell-1]}, g_1^\gamma \Big\}, \text{ and}$$

$$\mathcal{V} = \Big\{ M, (g_0^\alpha, g_1^{\alpha+\beta}), \{g_0^{\kappa_{0,i}} g_1^{\kappa_{1,i}}\}_{i \in [\ell]}, \{\langle \boldsymbol{\kappa}_0, \boldsymbol{b}_j^M \rangle, \langle \boldsymbol{\kappa}_1, \boldsymbol{b}_j^M \rangle\}_{j \in [\ell-1]}, g_1^{\beta \langle \boldsymbol{\kappa}_1, \boldsymbol{b}_\ell^M \rangle} \Big\}.$$

Consider $\boldsymbol{m} \in \Delta\mathcal{M}$, $\alpha_0, \beta_0 \in \mathbf{Z}/q\mathbf{Z}$, $\boldsymbol{h} \in (\mathbf{Z}/q\mathbf{Z})^\ell$, $\boldsymbol{v}_0, \boldsymbol{v}_1 \in (\mathbf{Z}/q\mathbf{Z})^{\ell-1}$, and let us denote $a := \log_{g_0}(g_1)$. It suffices to study the distance between the random variables $Y := \beta \langle \boldsymbol{\kappa}_1, \boldsymbol{b}_\ell^M \rangle \bmod q$ and γ conditioned on the conjunction of the following events: $M = \boldsymbol{m}$; $(\alpha \bmod q, \beta \bmod q) = (\alpha_0 \bmod q, \beta_0 \bmod q)$; $\boldsymbol{\kappa}_0 + a \cdot \boldsymbol{\kappa}_1 \bmod q = \boldsymbol{h} \bmod q$; and for $j \in [\ell - 1]$, $\langle \boldsymbol{\kappa}_0, \boldsymbol{b}_j^M \rangle = v_{0,j}, \langle \boldsymbol{\kappa}_1, \boldsymbol{b}_j^M \rangle = v_{1,j}$.

Let $(\boldsymbol{\kappa}_0^*, \boldsymbol{\kappa}_1^*)$ denote an arbitrary pair of vectors satisfying the same equations as $(\boldsymbol{\kappa}_0, \boldsymbol{\kappa}_1)$, i.e. those fixed by the events above. Then $\boldsymbol{\kappa}_0^* + a\boldsymbol{\kappa}_1^* = \boldsymbol{h} \bmod q$; $\langle \boldsymbol{\kappa}_0^*, \boldsymbol{b}_j^{\boldsymbol{m}} \rangle = v_{0,j}$ and $\langle \boldsymbol{\kappa}_1^*; \boldsymbol{b}_j^{\boldsymbol{m}} \rangle = v_{1,j}$ for $j \in [\ell - 1]$. Since for $i \in [\ell - 1]$, $\boldsymbol{b}_i^{\boldsymbol{m}} \in \boldsymbol{m}^\perp$, given the fixed information, the joint distribution of vectors $(\boldsymbol{\kappa}_0, \boldsymbol{\kappa}_1) \in (\mathbf{Z}/q\mathbf{Z})^2$ is:

$$\{(\boldsymbol{\kappa}_0^* - a \cdot \mu \cdot \boldsymbol{m} \bmod q, \boldsymbol{\kappa}_1^* + \mu \cdot \boldsymbol{m} \bmod q) \mid \mu \hookleftarrow \mathcal{U}(\mathbf{Z}/q\mathbf{Z})\}.$$

The conditional distribution of $\beta \langle \boldsymbol{\kappa}_1, \boldsymbol{b}_\ell^M \rangle$ is thus: $\{\beta(\langle \boldsymbol{\kappa}_1^*, \boldsymbol{b}_\ell^{\boldsymbol{m}} \rangle + \mu \langle \boldsymbol{m}, \boldsymbol{b}_\ell^{\boldsymbol{m}} \rangle) \bmod q \mid \mu \hookleftarrow \mathcal{U}(\mathbf{Z}/q\mathbf{Z})\}$ which is exactly $\mathcal{U}(\mathbf{Z}/q\mathbf{Z})$ since by construction $\boldsymbol{b}_\ell^{\boldsymbol{m}} \notin \boldsymbol{m}^\perp$, so $\langle \boldsymbol{m}, \boldsymbol{b}_\ell^{\boldsymbol{m}} \rangle \neq 0 \bmod q$, and $\beta \in (\mathbf{Z}/q\mathbf{Z})^*$. Thus $\mathcal{U} = \mathcal{V}$ and $\mathsf{H}_{\mathsf{DDH}}$ is 0-vector-smooth. □

3.5 Integrity

The high level idea to ensure $\mathsf{ind\text{-}fe\text{-}cca}$-security is to have the decryption algorithm reject any ciphertext whose decryption could leak more information than that revealed in an $\mathsf{ind\text{-}fe\text{-}cpa}$ attack. These harmful ciphertexts, dubbed *invalid ciphertexts*, are exactly those whose first component lives in $\widehat{\mathcal{X}} \backslash \widehat{\mathcal{L}}$. Hence, denoting this first ciphertext component x, one needs to ensure the decryption algorithm never decrypts a ciphertext with $x \notin \widehat{\mathcal{L}}$. To this end, upon encryption of a message vector of length ℓ, one computes ℓ evaluations of an extended projective hash function ehash over x, using independently sampled hashing keys $\mathsf{ehk}_1, \ldots, \mathsf{ehk}_\ell$. The resulting ciphertext contains x, the masked message components, and all the evaluations of ehash.
In our $\mathsf{ind\text{-}fe\text{-}cca}$-secure IPFE schemes a decryption key for $\boldsymbol{k} \in \mathcal{K}$ contains the linear combination $\overline{\mathsf{sk}}_{\boldsymbol{k}} := \sum_{i=1}^{\ell} k_i \mathsf{ehk}_i$. Using the key homomorphic property of the EPHF, a ciphertext will only be decrypted if $\mathsf{ehash}(\overline{\mathsf{sk}}_{\boldsymbol{k}}, x)$ yields the expected combination of the received ciphertext components. Now if the ciphertext is invalid, *i.e.* if $x \notin \widehat{\mathcal{L}}$, it must be infeasible for an adversary to compute a ciphertext which will not be rejected, even given all the auxiliary information it gets from the scheme's public values and from its key derivation queries. We thus introduce a new property for EPHFs: *vector universality*, which ensures that conditioned on the publicly available information (*i.e.* $\widehat{\mathsf{eprojkg}}(\mathbf{ehk})$); the adaptively chosen difference between challenge messages $\boldsymbol{m}$; the evaluation of ehash given by the challenge ciphertext; and all the information available on $\mathbf{ehk}$ from key derivation queries (*i.e.* the evaluations of $\lessdot \mathbf{ehk}, \boldsymbol{b} \gtrdot_{2a}$ for any $\boldsymbol{b} \in \boldsymbol{m}^\perp$), no adversary can predict an extended hash value π over an element $x \notin \widehat{\mathcal{L}}$ which would authorise decryption. Hence vector universality ensures ciphertext integrity in our upcoming constructions. The definition of vector universality, and proofs that our running examples possess it (Lemmas 5 and 6) are key to our achievements regarding IPFE schemes secure against active adversaries.

Definition 14 (δ_{vu}-vector universal). *Let $\mathcal{R}$ be a ring, either $\mathbf{Z}$ or $\mathbf{Z}/q\mathbf{Z}$ for some prime q; ℓ and a be positive integers; $\mathcal{SM} := (\widehat{\mathcal{X}}, \mathcal{X}, \widehat{\mathcal{L}}, \mathcal{W}, \mathsf{R})$ an SMP;*

and eH *the associated EPHF which we assume to be* $(\mathcal{R}, 2a, f, \aleph, \ell, \mathcal{M}, \mathcal{K})$-ipfe-*compatible. For* $i \in [\ell]$*, let* $\mathsf{ehk}_i \leftarrow \mathsf{ehashkg}(\mathcal{SM})$*, and* $\mathbf{ehk} := (\mathsf{ehk}_1, \dots, \mathsf{ehk}_\ell)$*. Consider a random variable* M *taking values in* $\Delta\mathcal{M}$ *and the associated matrix* $\mathbf{B}_M \in \mathcal{R}^{\ell \times \ell}$*. We say* eH *is* $\delta_{vu}(\ell)$*-vector universal if for any* $\widehat{\mathbf{ehp}} \in (K_{\widehat{\mathsf{ehp}}})^\ell$*; any* $\boldsymbol{m} \in \Delta\mathcal{M}$*; any* $\boldsymbol{k} \in \mathcal{K}$ *s.t.* $\boldsymbol{k} \notin \boldsymbol{m}^\perp$*; any* $(x^*, e^*) \in \widehat{\mathcal{X}} \times E$*,* $(x, e) \in \widehat{\mathcal{X}} \backslash \widehat{\mathcal{L}} \times E$*, s.t.* $(x, e) \neq (x^*, e^*)$*, and for any* $(\boldsymbol{v}_1, \dots, \boldsymbol{v}_{\ell-1}) \in (K_{\mathsf{ehk}})^{\ell-1}$*;* $\boldsymbol{\pi^*} \in \Pi^\ell$ *and* $\pi \in \Pi$ *it holds that:*

$$\begin{aligned}\Pr\big[\ &\mathsf{ehash}(\lessdot \mathbf{ehk}, \boldsymbol{k} \gtrdot_{2a}, x, e) = \pi \ \big|\ \widehat{\mathsf{eprojkg}}(\mathbf{ehk}) = \widehat{\mathbf{ehp}} \wedge M = \boldsymbol{m} \wedge \\ &\mathsf{ehash}(\mathbf{ehk}, x^*, e^*) = \boldsymbol{\pi^*} \wedge (\lessdot \mathbf{ehk}, \boldsymbol{b}_j^M \gtrdot_{2a} = \boldsymbol{v}_j \textit{ for } j \in [\ell - 1])\big] \leqslant \delta_{vu}(\ell).\end{aligned}$$

Example 1 – $\mathsf{HSM}_{\mathsf{CL}}$**.** Recall that for the $\mathsf{HSM}_{\mathsf{CL}}$ based EPHF, denoted $\mathsf{eH}_{\mathsf{CL}}$, the inner product ring is $\mathcal{R} = \mathbf{Z}$ and $K_{\mathsf{ehk}} = \mathbf{Z}^2$. In Lemma 5 we provide sufficient conditions for $\mathsf{eH}_{\mathsf{CL}}$ to be vector universal.

Lemma 5. *If algorithm* $\mathsf{ehashkg}$ *of* $\mathsf{eH}_{\mathsf{CL}}$ *samples hashing keys from* $\mathcal{D}_{\mathbf{Z},\sigma}$ *for* $\sigma > \sqrt{|\log_2(\delta)|}\tilde{s}p^{3/2}$*, and* $\Gamma : \widehat{G}^2 \mapsto \{0, \dots, p-1\}$ *is sampled from a family of* δ_Γ*-collision resistant hash functions, then* $\mathsf{eH}_{\mathsf{CL}}$ *is* δ_{vu}*-vector universal, where* $\delta_{vu} = 1/p + \delta_\Gamma + \delta$*.*

Proof. For $i \in [\ell]$, $\beta \in \{0, 1\}$, let $\mathsf{hk}_{\beta,i}$ be independent random variables following distribution $\mathcal{D}_{\mathbf{Z},\sigma}$; let $\mathbf{hk}_\beta := (\mathsf{hk}_{\beta,1}, \dots, \mathsf{hk}_{\beta,\ell}) \in \mathbf{Z}^\ell$. Let M be a random variable taking values in $\Delta\mathcal{M}$ and $\mathbf{B}_M \in \mathcal{R}^{\ell \times \ell}$ be the associated matrix. The lemma holds if for any $\widehat{\mathbf{ehp}} \in (\mathbf{Z}/\varpi\mathbf{Z})^{2\ell}$; any $\boldsymbol{m} \in \Delta\mathcal{M}$; any $\boldsymbol{k} \in \mathcal{K}$ s.t. $\boldsymbol{k} \notin \boldsymbol{m}^\perp$; any $(x^*, e^*) \in \widehat{G}^2$, $(x, e) \in \widehat{G} \backslash \widehat{G}^p \times \widehat{G}$, s.t. $(x, e) \neq (x^*, e^*)$; any $v_{\beta,j} \in \mathbf{Z}$ for $\beta \in \{0, 1\}$ and $j \in [\ell - 1]$; any $\boldsymbol{\pi^*} \in \widehat{G}^\ell$; any $\pi \in \widehat{G}$; and denoting $\gamma^* \leftarrow \Gamma(x^*, e^*)$ and $\gamma \leftarrow \Gamma(x, e)$ it holds that:

$$\begin{aligned}\Pr\big[&x^{\langle \mathbf{hk}_0 + \gamma \mathbf{hk}_1, \boldsymbol{k} \rangle} = \pi \mid \widehat{\mathsf{eprojkg}}(\mathbf{hk}_0, \mathbf{hk}_1) = \widehat{\mathbf{ehp}} \wedge M = \boldsymbol{m} \wedge \\ &(x^*)^{\mathbf{hk}_0 + \gamma^* \mathbf{hk}_1} = \boldsymbol{\pi}^* \wedge \langle \mathbf{hk}_\beta, \boldsymbol{b}_j^M \rangle = v_{\beta,j} \text{ for } j \in [\ell - 1], \beta \in \{0, 1\}\big] \leqslant \delta_{vu}.\end{aligned}$$

We consider the information on vector $\mathbf{ehk}$ fixed by the following events:

1. E_1 is the event "$\widehat{\mathsf{eprojkg}}(\mathbf{hk}_0, \mathbf{hk}_1) = \widehat{\mathbf{ehp}}$". Denoting $\widehat{\mathbf{ehp}} = (\widehat{\mathbf{ehp}}_0, \widehat{\mathbf{ehp}}_1) \in (\mathbf{Z}/\varpi\mathbf{Z})^{2\ell}$, given E_1 it holds that:
$$\mathbf{hk}_0 \bmod \varpi = \widehat{\mathbf{ehp}}_0 \bmod \varpi \qquad \text{and} \qquad \mathbf{hk}_1 \bmod \varpi = \widehat{\mathbf{ehp}}_1 \bmod \varpi.$$
2. E_2 is the event "$M = \boldsymbol{m}$". Denoting $\mathbf{B}_{\boldsymbol{m}}$ the matrix associated to $\boldsymbol{m}$, built as per Lemma 1, it holds that $\mathbf{B}_M = \mathbf{B}_{\boldsymbol{m}}$. Let $\boldsymbol{b}_1^{\boldsymbol{m}}, \dots, \boldsymbol{b}_\ell^{\boldsymbol{m}}$ denote the rows of $\mathbf{B}_{\boldsymbol{m}}$. From Def. 12 it hold that $\boldsymbol{b}_\ell^{\boldsymbol{m}} = \boldsymbol{m}$ since $\mathcal{R} = \mathbf{Z}$.
3. E_3 is the event "$(x^*)^{\mathbf{hk}_0 + \gamma^* \mathbf{hk}_1} = \boldsymbol{\pi^*}$". Since $\widehat{G}$ is of order $\widehat{n}$, we upper bound the information provided by E_3 (hence upper bounding any probability conditioned on E_3) by considering a vector $\boldsymbol{h} \in (\mathbf{Z}/\widehat{n}\mathbf{Z})^\ell$ and conditioning on the event "$\mathbf{hk}_0 + \gamma^* \mathbf{hk}_1 \bmod \widehat{n} = \boldsymbol{h} \bmod \widehat{n}$". The conditional joint distribution of $(\mathbf{hk}_0 \bmod \widehat{n}, \mathbf{hk}_1 \bmod \widehat{n})$ is thus:
$$\{(\boldsymbol{h} - \gamma^* \mathbf{hk}_1 \bmod \widehat{n}, \mathbf{hk}_1 \bmod \widehat{n}) \mid \mathbf{hk}_1 \hookleftarrow \mathcal{D}_{\mathbf{Z}^\ell,\sigma}\}. \tag{1}$$
4. E_4 is the event "$\langle \mathbf{hk}_\beta, \boldsymbol{b}_j^M \rangle = v_{\beta,j}$ in $\mathbf{Z}$ for $j \in [\ell - 1], \beta \in \{0, 1\}$".

We first evaluate the distribution followed by $\boldsymbol{\mathsf{hk}}_1$ conditioned on these events. Let $\boldsymbol{\mathsf{hk}}_1^* \in \mathbf{Z}^\ell$ denote an arbitrary vector satisfying the same equations as $\boldsymbol{\mathsf{hk}}_1$, i.e. for $j \in [\ell - 1]$, $\langle \boldsymbol{\mathsf{hk}}_1^*, \boldsymbol{b}_j^{\boldsymbol{m}} \rangle = \upsilon_{1,j}$ and $\boldsymbol{\mathsf{hk}}_1^* \bmod \varpi = \widehat{\boldsymbol{\mathsf{ehp}}}_1 \bmod \varpi$. We define $\mathit{\Lambda} := \{\boldsymbol{t} \in \mathbf{Z}^\ell \mid \langle \boldsymbol{t}, \boldsymbol{b}_j^{\boldsymbol{m}} \rangle = 0 \text{ for } j \in [\ell - 1]; \boldsymbol{t} = \mathbf{0} \bmod \varpi\} \subset \mathbf{Z}^\ell$, so that, as in proof of Lemma 3, the conditional distribution followed by $\boldsymbol{\mathsf{hk}}_1$ is $\{\boldsymbol{\mathsf{hk}}_1^* + \mathcal{D}_{\mathit{\Lambda},\sigma,-\boldsymbol{\mathsf{hk}}_1^*}\}$.

Denoting $d \neq 0$ the gcd of the coefficients of $\boldsymbol{b}_\ell^{\boldsymbol{m}}$ and $\tilde{\boldsymbol{b}} = 1/d \cdot \boldsymbol{b}_\ell^{\boldsymbol{m}} \in \mathbf{Z}^\ell$, it holds that for $j \in [\ell - 1]$, $\boldsymbol{b}_j^{\boldsymbol{m}} \in \tilde{\boldsymbol{b}}^\perp$. Now consider the 1-dimensional lattice $\mathit{\Lambda}' := \{\boldsymbol{t} \in \mathbf{Z}^\ell | \langle \boldsymbol{t}, \boldsymbol{b}_j^{\boldsymbol{m}} \rangle = 0 \text{ for } j \in [\ell - 1]\}$ which contains $\tilde{\boldsymbol{b}}\mathbf{Z}$. In fact as $\gcd(\tilde{b}_1, \ldots, \tilde{b}_\ell) = 1$, one has $\mathit{\Lambda}' = \tilde{\boldsymbol{b}}\mathbf{Z}$ (indeed there exists $\boldsymbol{y} \in \mathbf{Z}^\ell$ s.t. $\mathit{\Lambda}' = \boldsymbol{y} \cdot \mathbf{Z}$, and $\tilde{\boldsymbol{b}} = \alpha \boldsymbol{y}$, so α must divide $\gcd(\tilde{b}_1, \ldots, \tilde{b}_\ell) = 1$). Moreover $\mathit{\Lambda} = \mathit{\Lambda}' \cap \varpi \cdot \mathbf{Z}^\ell = (\tilde{\boldsymbol{b}} \cdot \mathbf{Z} \cap \varpi \cdot \mathbf{Z}^\ell) = \varpi \cdot \tilde{\boldsymbol{b}} \cdot \mathbf{Z}$, since $\forall \mu \in \mathbf{Z}$, in order for ϖ to divide each $\mu \tilde{b}_i$, ϖ must divide $\mu \cdot \gcd(\tilde{b}_1, \ldots, \tilde{b}_\ell) = \mu$. We now consider the distribution of $\langle \boldsymbol{\mathsf{hk}}_1, \tilde{\boldsymbol{b}} \rangle$, and then reduce it mod p, so as to prove that the random variable $\tilde{Y} := \langle \boldsymbol{\mathsf{hk}}_1, \tilde{\boldsymbol{b}} \rangle \bmod p$ follows a distribution close to $\mathcal{U}(\mathbf{Z}/p\mathbf{Z})$. Let us denote $\mathit{\Lambda}_0 := \varpi \cdot ||\tilde{\boldsymbol{b}}||_2^2 \cdot \mathbf{Z}$. It follows from Lemma 10 that the distribution followed by $\langle \boldsymbol{\mathsf{hk}}_1, \tilde{\boldsymbol{b}} \rangle$ is $\langle \boldsymbol{\mathsf{hk}}_1^*, \tilde{\boldsymbol{b}} \rangle + \mathcal{D}_{\mathit{\Lambda}_0, ||\tilde{\boldsymbol{b}}||_2 \cdot \sigma, -c}$ where $c = \langle \boldsymbol{\mathsf{hk}}_1^*, \tilde{\boldsymbol{b}} \rangle$ in $\mathbf{Z}$. As in proof of Lemma 3, we reduce the distribution $\mathcal{D}_{\mathit{\Lambda}_0, ||\tilde{\boldsymbol{b}}||_2 \cdot \sigma, -c}$ over $\mathit{\Lambda}_0$ modulo $\mathit{\Lambda}_0' := p\mathit{\Lambda}_0$. Since $\sigma > \sqrt{|\log_2(\delta)|} \cdot \tilde{s} \cdot p^{3/2}$ it holds that $\langle \boldsymbol{\mathsf{hk}}_1, \tilde{\boldsymbol{b}} \rangle \bmod p$ is δ-close to the uniform distribution over $\mathit{\Lambda}_0/\mathit{\Lambda}_0'$; and since ϖ and p are co-prime, $\mathit{\Lambda}_0/\mathit{\Lambda}_0' \simeq \mathbf{Z}/p\mathbf{Z}$. Using (1), the distribution of $\langle \boldsymbol{\mathsf{hk}}_0 + \gamma \boldsymbol{\mathsf{hk}}_1, \tilde{\boldsymbol{b}} \rangle \bmod p$ is thus:

$$\{\langle \boldsymbol{h}, \tilde{\boldsymbol{b}} \rangle + (\gamma - \gamma^*)(\langle \boldsymbol{\mathsf{hk}}_1^*, \tilde{\boldsymbol{b}} \rangle + \nu) \bmod p | \ \nu \hookleftarrow (\mathcal{D}_{\mathit{\Lambda}_0, ||\tilde{\boldsymbol{b}}||_2 \sigma, -c} \bmod \mathit{\Lambda}_0')\} \tag{2}$$

Since $(\mathcal{D}_{\mathit{\Lambda}_0, ||\tilde{\boldsymbol{b}}||_2 \sigma, -c} \bmod \mathit{\Lambda}_0')$ is δ-close to $\mathcal{U}(\mathbf{Z}/p\mathbf{Z})$, and $\boldsymbol{h}$ is given, if $\gamma \neq \gamma^* \bmod p$ then distribution (2) is δ-close to $\mathcal{U}(\mathbf{Z}/p\mathbf{Z})$.

We now estimate the conditional probability that $x^{\langle \boldsymbol{\mathsf{hk}}_0 + \gamma \boldsymbol{\mathsf{hk}}_1, \boldsymbol{k} \rangle} = \pi$. Since $x \in \widehat{G} \backslash \widehat{G}^p$, there exist unique $x_0 \in \widehat{G}^p$, $\upsilon \in (\mathbf{Z}/p\mathbf{Z})^*$ satisfying $x = x_0 f^\upsilon$. Thus $x^{\langle \boldsymbol{\mathsf{hk}}_0 + \gamma \boldsymbol{\mathsf{hk}}_1, \boldsymbol{k} \rangle} = x_0^{\langle \boldsymbol{\mathsf{hk}}_0 + \gamma \boldsymbol{\mathsf{hk}}_1, \boldsymbol{k} \rangle} f^{\upsilon \langle \boldsymbol{\mathsf{hk}}_0 + \gamma \boldsymbol{\mathsf{hk}}_1, \boldsymbol{k} \rangle}$. From the knowledge of x, $\boldsymbol{k}$ and E_1 the value of $x_0^{\langle \boldsymbol{\mathsf{hk}}_0 + \gamma \boldsymbol{\mathsf{hk}}_1, \boldsymbol{k} \rangle} = x_0^{\langle \widehat{\boldsymbol{\mathsf{ehp}}}_0 + \gamma \widehat{\boldsymbol{\mathsf{ehp}}}_1, \boldsymbol{k} \rangle}$ is fixed. Moreover since υ is information theoretically theoretically fixed by x, and f is of prime order p, it suffices to bound the probability $\langle \boldsymbol{\mathsf{hk}}_0 + \gamma \boldsymbol{\mathsf{hk}}_1, \boldsymbol{k} \rangle \bmod p$ takes a given fixed value. Now since the matrix $\mathbf{B}_{\boldsymbol{m}}$ is invertible mod p, $\boldsymbol{k} \bmod p$ can be uniquely expressed as a linear combination of the rows of $\mathbf{B}_{\boldsymbol{m}}$. We denote this decomposition:

$$\boldsymbol{k} = \sum_{i \in [\ell]} \alpha_i \boldsymbol{b}_i^{\boldsymbol{m}} \bmod p \text{ with } \alpha_\ell \in (\mathbf{Z}/p\mathbf{Z})^* \text{ and } \alpha_i \in \mathbf{Z}/p\mathbf{Z} \text{ for } i \in [\ell - 1].$$

From $\boldsymbol{k}$, E_2 and E_4, for $i \in [\ell - 1]$ the values $\langle \boldsymbol{\mathsf{hk}}_0 + \gamma \boldsymbol{\mathsf{hk}}_1, \alpha_i \boldsymbol{b}_i^{\boldsymbol{m}} \rangle$ are fixed. And so we need only consider the probability $\alpha_\ell \langle \boldsymbol{\mathsf{hk}}_0 + \gamma \boldsymbol{\mathsf{hk}}_1, \boldsymbol{b}_\ell^{\boldsymbol{m}} \rangle = d \cdot \alpha_\ell \langle \boldsymbol{\mathsf{hk}}_0 + \gamma \boldsymbol{\mathsf{hk}}_1, \tilde{\boldsymbol{b}} \rangle$ takes a fixed value modulo p. But from (2) we know that, if $\gamma \neq \gamma^* \bmod p$ then $\langle \boldsymbol{\mathsf{hk}}_0 + \gamma \boldsymbol{\mathsf{hk}}_1, \tilde{\boldsymbol{b}} \rangle \bmod p$ follows a distribution δ-close to $\mathcal{U}(\mathbf{Z}/p\mathbf{Z})$. Note that, since $\gamma^* = \mathit{\Gamma}(x^*, e^*)$ and $\gamma = \mathit{\Gamma}(x, e)$, the event $\gamma = \gamma^* \bmod p$ occurs with probability $\leqslant \delta_{\mathit{\Gamma}}$. We conclude that the conditional probability $\langle \boldsymbol{\mathsf{hk}}_0 + \gamma \boldsymbol{\mathsf{hk}}_1, \tilde{\boldsymbol{b}} \rangle$ takes a given value mod p is $\leqslant 1/p + \delta + \delta_{\mathit{\Gamma}}$, which concludes the proof. □

Example 2 – DDH. Recall that for $\mathsf{eH}_{\mathsf{DDH}}$ the inner product ring is $\mathscr{R} = \mathbf{Z}/q\mathbf{Z}$ and $K_{\mathsf{ehk}} = (\mathbf{Z}/q\mathbf{Z})^4$. In Lemma 6 we provide sufficient conditions for $\mathsf{H}_{\mathsf{DDH}}$ to be vector universal.

Lemma 6. *If $\Gamma : G^3 \mapsto \{0, \ldots, q-1\}$ is sampled from a family of δ_Γ-collision resistant hash functions, then $\mathsf{eH}_{\mathsf{DDH}}$ is δ_{vu}-vector universal, for $\delta_{vu} = 1/q + \delta_\Gamma$.*

Proof. For $i \in [\ell]$ let $\mathsf{ehk}_i := (\kappa_{0,i}, \kappa_{1,i}, \kappa_{2,i}, \kappa_{3,i})$ denote independent random variables following the distribution $\mathcal{U}((\mathbf{Z}/q\mathbf{Z})^4)$; let $\textbf{\textsf{ehk}} := (\mathsf{ehk}_1, \ldots, \mathsf{ehk}_\ell)$ and $\boldsymbol{\kappa}_\mu := (\kappa_{\mu,1}, \ldots, \kappa_{\mu,\ell}) \in (\mathbf{Z}/q\mathbf{Z})^\ell$ for $\mu \in \{0,1,2,3\}$. Consider a random variable M taking values in $\Delta\mathcal{M}$ and the associated matrix $\mathbf{B}_M \in \mathbf{Z}/q\mathbf{Z}^{\ell \times \ell}$.

Lemma 6 holds if for any $\textbf{\textsf{ehp}} \in G^{2\ell}$; any $\boldsymbol{m} \in \Delta\mathcal{M}$; any $\boldsymbol{k} \in \mathcal{K}$ s.t. $\boldsymbol{k} \notin \boldsymbol{m}^\perp$; any $((x_0^*, x_1^*), e^*) \in G^2 \times G$, $((x_0, x_1), e) \in (G^2 \backslash \langle (g_0, g_1) \rangle) \times G$, s.t. $((x_0, x_1), e) \neq ((x_0^*, x_1^*), e^*)$; any $v_{\mu,j} \in \mathbf{Z}/q\mathbf{Z}$ for $\mu \in \{0,1,2,3\}$ and $j \in [\ell-1]$; any $\pi_i^* \in G$ for $i \in [\ell]$; and for any $\pi \in G$, denoting $\gamma^* \leftarrow \Gamma((x_0^*, x_1^*), e^*)$ and $\gamma \leftarrow \Gamma((x_0, x_1), e)$, it holds that:

$$\begin{aligned}\Pr[x_0^{\langle \boldsymbol{\kappa}_0 + \gamma \boldsymbol{\kappa}_2, \boldsymbol{k} \rangle} x_1^{\langle \boldsymbol{\kappa}_1 + \gamma \boldsymbol{\kappa}_3, \boldsymbol{k} \rangle} = \pi \,\big|\, &(x_0^*)^{\kappa_{0,i} + \gamma^* \kappa_{2,i}} (x_1^*)^{\kappa_{1,i} + \gamma^* \kappa_{3,i}} = \pi_i^* \text{ for } i \in [\ell] \\ &\wedge \langle \boldsymbol{\kappa}_\mu, \boldsymbol{b}_j^M \rangle = v_{\mu,j} \text{ for } j \in [\ell-1], \mu \in \{0, \ldots, 3\} \\ &\wedge (g_0^{\boldsymbol{\kappa}_0} g_1^{\boldsymbol{\kappa}_1}, g_0^{\boldsymbol{\kappa}_2} g_1^{\boldsymbol{\kappa}_3}) = \textbf{\textsf{ehp}} \wedge M = \boldsymbol{m}] \leqslant \delta_{vu},\end{aligned}$$

Let $a := \log_{g_0}(g_1)$, $\alpha^* := \log_{x_0^*}(x_1^*)$, and $\alpha := \log_{x_0}(x_1)$. Since $(x_0, x_1) \notin \langle (g_0, g_1) \rangle$ we have $\alpha \neq a \bmod q$. We consider the information on $\textbf{\textsf{ehk}}$ fixed by the following events:

1. E_1 is the event "$(x_0^*)^{\kappa_{0,i} + \gamma^* \kappa_{2,i}} (x_1^*)^{\kappa_{1,i} + \gamma^* \kappa_{3,i}} = \pi_i^*$ for $i \in [\ell]$". Equivalently, for a vector $\boldsymbol{h} \in (\mathbf{Z}/q\mathbf{Z})^\ell$, E_1 is the event "$(\boldsymbol{\kappa}_0 + \gamma^* \boldsymbol{\kappa}_2) + \alpha^* (\boldsymbol{\kappa}_1 + \gamma^* \boldsymbol{\kappa}_3) \bmod q = \boldsymbol{h}$".
2. E_2 is the event "$(g_0^{\boldsymbol{\kappa}_0} g_1^{\boldsymbol{\kappa}_1},\ g_0^{\boldsymbol{\kappa}_2} g_1^{\boldsymbol{\kappa}_3}) = \textbf{\textsf{ehp}}$". Equivalently, for some $\boldsymbol{s}_0$ and $\boldsymbol{s}_1$ in $(\mathbf{Z}/q\mathbf{Z})^\ell$, E_2 is the event "$\boldsymbol{\kappa}_0 + a\boldsymbol{\kappa}_1 \bmod q = \boldsymbol{s}_0 \bmod q$ and $\boldsymbol{\kappa}_2 + a\boldsymbol{\kappa}_3 \bmod q = \boldsymbol{s}_1 \bmod q$". Observe that if $\alpha^* = a \bmod q$ then E_1 provides no more information than E_2 alone. We hereafter assume $\alpha^* \neq a \bmod q$. Conditioned on E_1 and E_2, the joint distribution of $(\boldsymbol{\kappa}_0, \boldsymbol{\kappa}_1,\ \boldsymbol{\kappa}_2, \boldsymbol{\kappa}_3)$ is thus:
$$\begin{aligned}\{(&\boldsymbol{s}_0 - a((\alpha^* - a)^{-1}(\boldsymbol{h} - \boldsymbol{s}_0 - \gamma^* \boldsymbol{s}_1) - \gamma^* \boldsymbol{\mu}), \\ &\quad (\alpha^* - a)^{-1}(\boldsymbol{h} - \boldsymbol{s}_0 - \gamma^* \boldsymbol{s}_1) - \gamma^* \boldsymbol{\mu}, \boldsymbol{s}_1 - a\boldsymbol{\mu}, \boldsymbol{\mu}) \mid \boldsymbol{\mu} \in (\mathbf{Z}/q\mathbf{Z})^\ell\}. \quad (3)\end{aligned}$$
3. E_3 is the event "$\langle \boldsymbol{\kappa}_\mu, \boldsymbol{b}_j^M \rangle = v_{\mu,j}$ for $j \in [\ell-1]$, $\mu \in \{0,1,2,3\}$".
4. E_4 is the event "$M = \boldsymbol{m}$". Denoting $\mathbf{B}_{\boldsymbol{m}}$ the matrix associated to $\boldsymbol{m}$ built as per Lemma 1, it holds that $\mathbf{B}_M = \mathbf{B}_{\boldsymbol{m}}$. Let $\boldsymbol{b}_1^{\boldsymbol{m}}, \ldots, \boldsymbol{b}_\ell^{\boldsymbol{m}}$ denote the rows of $\mathbf{B}_{\boldsymbol{m}}$.

We first evaluate the distribution followed by $(\langle \boldsymbol{\kappa}_0, \boldsymbol{k} \rangle,\ \langle \boldsymbol{\kappa}_1, \boldsymbol{k} \rangle, \langle \boldsymbol{\kappa}_2, \boldsymbol{k} \rangle, \langle \boldsymbol{\kappa}_3, \boldsymbol{k} \rangle)$ conditioned on the conjunction of E_1, E_2, E_3, E_4. Let $(\boldsymbol{\kappa}_0^*, \boldsymbol{\kappa}_1^*, \boldsymbol{\kappa}_2^*, \boldsymbol{\kappa}_3^*)$ denote an arbitrary quadruple of vectors satisfying the same equations as $(\boldsymbol{\kappa}_0, \boldsymbol{\kappa}_1, \boldsymbol{\kappa}_2, \boldsymbol{\kappa}_3)$, i.e. those fixed by E_1, E_2, E_3, E_4. Then

$$\begin{cases} \boldsymbol{\kappa}_0^* = (\boldsymbol{s}_0 - a((\alpha^* - a)^{-1}(\boldsymbol{h} - \boldsymbol{s}_0 - \gamma^* \boldsymbol{s}_1) - \gamma^* \boldsymbol{\kappa}_3^*) \bmod q \\ \boldsymbol{\kappa}_1^* = (\alpha^* - a)^{-1}(\boldsymbol{h} - \boldsymbol{s}_0 - \gamma^* \boldsymbol{s}_1) - \gamma^* \boldsymbol{\kappa}_3^* \bmod q \\ \boldsymbol{\kappa}_2^* = \boldsymbol{s}_1 - a\boldsymbol{\kappa}_3^* \bmod q \\ \langle \boldsymbol{\kappa}_\mu^*, \boldsymbol{b}_j^{\boldsymbol{m}} \rangle = v_{\mu,j} \text{ for } j \in [\ell - 1], \mu \in \{0, 1, 2, 3\} \end{cases}$$

Since for $i \in [\ell - 1]$, all vectors $\boldsymbol{b}_i^{\boldsymbol{m}}$ are in $\boldsymbol{m}^\perp$ the joint conditional distribution of vectors $(\boldsymbol{\kappa}_0, \boldsymbol{\kappa}_1, \boldsymbol{\kappa}_2, \boldsymbol{\kappa}_3)$ is:

$$\{(\boldsymbol{\kappa}_0^* + \gamma^* \cdot a \cdot \mu \cdot \boldsymbol{m}, \boldsymbol{\kappa}_1^* - \gamma^* \cdot \mu \cdot \boldsymbol{m}, \boldsymbol{\kappa}_2^* - a \cdot \mu \cdot \boldsymbol{m}, \boldsymbol{\kappa}_3^* + \mu \cdot \boldsymbol{m}) \mid \mu \in \mathbf{Z}/q\mathbf{Z}\}. \quad (4)$$

The conditional distribution of $\langle \boldsymbol{\kappa}_3, \boldsymbol{k} \rangle$ is thus: $\{\langle \boldsymbol{\kappa}_3^*, \boldsymbol{k} \rangle + \mu \langle \boldsymbol{m}, \boldsymbol{k} \rangle \mid \mu \in \mathbf{Z}/q\mathbf{Z}\}$ which is exactly $\mathcal{U}(\mathbf{Z}/q\mathbf{Z})$ since by definition $\boldsymbol{k} \notin \boldsymbol{m}^\perp$, so $\langle \boldsymbol{m}, \boldsymbol{k} \rangle \neq 0 \bmod q$.

We now consider the probability $\mathfrak{p}$ that event $E_0 :=$"$x_0^{\langle \boldsymbol{\kappa}_0 + \gamma \boldsymbol{\kappa}_2, \boldsymbol{k} \rangle} x_1^{\langle \boldsymbol{\kappa}_1 + \gamma \boldsymbol{\kappa}_3, \boldsymbol{k} \rangle} = \pi$" occurs, i.e. that the random variable $X_1 := \langle \boldsymbol{\kappa}_0 + \alpha \boldsymbol{\kappa}_1 + \gamma(\boldsymbol{\kappa}_2 + \alpha \boldsymbol{\kappa}_3), \boldsymbol{k} \rangle \bmod q$ takes a fixed value mod q. From Eq. (3) we have:

$$\begin{aligned} X_1 = \langle & \boldsymbol{s}_0 - a((\alpha^* - a)^{-1}(\boldsymbol{h} - \boldsymbol{s}_0 - \gamma^* \boldsymbol{s}_1) - \gamma^* \boldsymbol{\kappa}_3) \\ & + \alpha((\alpha^* - a)^{-1}(\boldsymbol{h} - \boldsymbol{s}_0 - \gamma^* \boldsymbol{s}_1) - \gamma^* \boldsymbol{\kappa}_3) + \gamma((\boldsymbol{s}_1 - a\boldsymbol{\kappa}_3) + \alpha \boldsymbol{\kappa}_3), \boldsymbol{k} \rangle \\ = \langle & \boldsymbol{s}_0 + \gamma \boldsymbol{s}_1 + (\alpha - a)(\alpha^* - a)^{-1}(\boldsymbol{h} - \boldsymbol{s}_0 - \gamma^* \boldsymbol{s}_1), \boldsymbol{k} \rangle \\ & + (a - \alpha)(\gamma^* - \gamma) \langle \boldsymbol{\kappa}_3, \boldsymbol{k} \rangle. \end{aligned}$$

Where $a \neq \alpha^* \bmod q$ and $(\langle \boldsymbol{s}_0 + \gamma \boldsymbol{s}_1 + (\alpha^* - a)^{-1}(\alpha - a)(\boldsymbol{h} - \boldsymbol{s}_0 - \gamma^* \boldsymbol{s}_1), \boldsymbol{k} \rangle \bmod q)$ is fixed by E_1, E_2, E_3, E_4 and the value of $\boldsymbol{k}$. Thus if $\gamma \neq \gamma^* \bmod q$ then X_1 follows the uniform distribution modulo q and so the conditional probability E_0 occurs is $1/q$. Now since $\gamma^* = \mathit{\Gamma}((x_0^*, x_1^*), e^*)$ and $\gamma = \mathit{\Gamma}((x_0, x_1), e)$, the event "$\gamma = \gamma^* \bmod q$" occurs with probability $\leqslant \delta_{\mathit{\Gamma}}$. We can conclude that $\mathfrak{p} \leqslant 1/p + \delta_{\mathit{\Gamma}}$, and denoting $\delta_{vu} := 1/p + \delta_{\mathit{\Gamma}}$, it holds that $\mathsf{H}_{\mathsf{DDH}}$ is δ_{vu}-vector-universal. □

3.6 Inner product safe projective hash function

The notions of active and passive inner product safe PHFs summarise the properties required to build ind-fe-cca and ind-fe-cpa secure IPFE schemes.

Definition 15 (pip-safe)**.** *Let $\mathcal{R}$ be a ring, either $\mathbf{Z}$ or $\mathbf{Z}/q\mathbf{Z}$ for some prime q; $\ell, a \in \mathbf{N}$; $\mathcal{SM} := (\widehat{\mathcal{X}}, \mathcal{X}, \widehat{\mathcal{L}}, \mathcal{W}, \mathsf{R})$ an SMP;* H *the associated PHF which we assume to be $(\mathcal{R}, a, f, \aleph, \ell, \mathcal{M}, \mathcal{K})$-*ipfe*-compatible; and let $F := \langle f \rangle$. Then* H *is $(\mathcal{R}, a, f, \aleph, \ell, \mathcal{M}, \mathcal{K}, \widehat{\mathit{\Upsilon}}, \mathit{\Upsilon}, \delta_{\mathcal{L}}, \delta_{vs})$-passive inner product safe* (pip-safe) *if: (1) the order $\aleph$ of F is prime or hard to factor; (2)* H *is $(\widehat{\mathit{\Upsilon}}, \mathit{\Upsilon}, F)$-decomposable, where $\widehat{\mathit{\Upsilon}} \in \widehat{\mathcal{X}}$, $\mathit{\Upsilon} \in \mathcal{X}$; (3)* H *is homomorphic; (4) $\mathcal{SM}$ is $\delta_{\mathcal{L}}$-hard; and (5)* H *is δ_{vs}-vector smooth over $\mathcal{X}$ on F.*

Definition 16 (aip-safe)**.** *Let $\mathcal{R}$ be a ring, either $\mathbf{Z}$ or $\mathbf{Z}/q\mathbf{Z}$ for some prime q; $\ell, a \in \mathbf{N}$; $\mathcal{SM} := (\widehat{\mathcal{X}}, \mathcal{X}, \widehat{\mathcal{L}}, \mathcal{W}, \mathsf{R})$ an SMP;* H *the associated PHF which we assume to be $(\mathcal{R}, a, f, \aleph, \ell, \mathcal{M}, \mathcal{K})$-*ipfe*-compatible;* eH *the resulting EPHF (as per the generic construction of [CS02], cf. Appx. B). The pair* (H, eH) *is $(\mathcal{R}, a, f, \aleph, \ell, \mathcal{M}, \mathcal{K}, \widehat{\mathit{\Upsilon}}, \mathit{\Upsilon}, \delta_{\mathcal{L}}, \delta_{vs}, \delta_{vu})$-active inner product safe* (aip-safe) *if* H *is $(\mathcal{R}, a, f, \aleph, \ell, \mathcal{M}, \mathcal{K}, \widehat{\mathit{\Upsilon}}, \mathit{\Upsilon}, \delta_{\mathcal{L}}, \delta_{vs})$-*pip-safe *and* eH *is δ_{vu}-vector universal.*

4 IPFE from PHFs secure against passive adversaries

Let $\mathcal{R}$ be either the ring $\mathbf{Z}$ or $\mathbf{Z}/q\mathbf{Z}$ for some prime q; $\mathsf{Gen}_{\mathcal{SM}}$ be a subgroup membership problem generator outputting the description $\mathcal{SM} := (\widehat{\mathcal{X}}, \mathcal{X}, \widehat{\mathcal{L}}, \mathcal{W}, \mathsf{R})$ of an SMP; ℓ and a be positive integers; $\mathcal{M} \subseteq \mathcal{R}^\ell$ be the plaintext space; and $\mathcal{K} \subseteq \mathcal{R}^\ell$ be the space from which keys are derived. Building upon a $\mathsf{pip\text{-}safe}$ projective hash family H, the scheme recovers $\langle \boldsymbol{m}, \boldsymbol{k} \rangle \in \mathcal{R}$ for $\boldsymbol{m} \in \mathcal{M}$, $\boldsymbol{k} \in \mathcal{K}$. The resulting $\mathsf{ind\text{-}fe\text{-}cpa}$-secure IPFE scheme is depicted in Fig. 1.

```
Setup($1^\lambda, 1^\ell$):
  1. $\mathcal{SM} \leftarrow \mathsf{Gen}_{\mathcal{SM}}(1^\lambda)$
  2. For $1 \leqslant i \leqslant \ell$ :
  3.     Sample $\mathsf{hk}_i \leftarrow \mathsf{hashkg}(\mathcal{SM})$
  4.     $\mathsf{hp}_i \leftarrow \mathsf{projkg}(\mathsf{hk}_i)$
  5. Return $\mathsf{mpk} := \mathbf{hp}$; $\mathsf{msk} := \mathbf{hk}$

Enc($\mathsf{mpk}, \boldsymbol{m}$):
  1. If $\boldsymbol{m} \notin \mathcal{M}$ return $\perp$
  2. Sample $(c_0, w) \leftarrow \mathsf{R}$
  3. For $1 \leqslant i \leqslant \ell$ :
  4.     $c_i \leftarrow \mathsf{projhash}(\mathsf{hp}_i, c_0, w) \cdot f^{m_i}$
  5. Return $\boldsymbol{ct} := (c_0, \boldsymbol{c})$
```

```
KeyDer($\mathsf{msk}, \boldsymbol{k}$):
  1. If $\boldsymbol{k} \notin \mathcal{K}$ return $\perp$
  2. $\mathsf{sk}_{\boldsymbol{k}} \leftarrow \lessdot \mathbf{hk}, \boldsymbol{k} \gtrdot_a$
  3. Return $(\mathsf{sk}_{\boldsymbol{k}}, \boldsymbol{k})$

Dec($\mathsf{mpk}, (\mathsf{sk}_{\boldsymbol{k}}, \boldsymbol{k}), \boldsymbol{ct}$):
  1. If $\boldsymbol{ct} \notin \widehat{\mathcal{X}} \times \Pi^\ell$ then return $\perp$
  2. Parse $(c_0, \boldsymbol{c}) = \boldsymbol{ct}$
  3. $M \leftarrow (\prod_{i \in [\ell]} c_i^{k_i}) \cdot \mathsf{hash}(\mathsf{sk}_{\boldsymbol{k}}, c_0)^{-1}$
  4. If $M \notin F$ then return $\perp$
  5. Return $\mathsf{sol} \leftarrow \log_f(M)$
```

Fig. 1: IPFE that is $\mathsf{ind\text{-}fe\text{-}cpa}$-secure from projective hash functions

Correctness. As $K_{\mathsf{hk}} = \mathcal{R}^a$ one has $\mathbf{hk} \in (\mathcal{R}^\ell)^a$, so $\lessdot \mathbf{hk}, \boldsymbol{k} \gtrdot_a \in \mathcal{R}^a$. Next, as H is key homomorphic, $\prod_{i \in [\ell]} c_i^{k_i} = \prod_{i \in [\ell]} (\mathsf{hash}(\mathsf{hk}_i, c_0) f^{m_i})^{k_i} = f^{\langle \boldsymbol{k}, \boldsymbol{m} \rangle} \mathsf{hash}(\mathsf{sk}_{\boldsymbol{k}}, c_0)$ so $\prod_{i \in [\ell]} c_i^{k_i} \cdot \mathsf{hash}(\mathsf{sk}_{\boldsymbol{k}}, c_0)^{-1} = f^{\langle \boldsymbol{k}, \boldsymbol{m} \rangle} \in F$. Since H is $(\mathcal{R}, a, f, \aleph, \ell, \mathcal{M}, \mathcal{K})$-$\mathsf{ipfe}$-compatible, $\log_f(f^{\langle \boldsymbol{k}, \boldsymbol{m} \rangle}) = \langle \boldsymbol{k}, \boldsymbol{m} \rangle \in \mathcal{R}$. Consequently, for any $\mathsf{mpk}, \mathsf{msk} \leftarrow \mathsf{Setup}(1^\lambda, 1^\ell)$, $\boldsymbol{k} \in \mathcal{K}$, and $\boldsymbol{m} \in \mathcal{M}$ it holds that $\mathsf{Dec}(\mathsf{mpk}, \mathsf{KeyDer}(\mathsf{msk}, \boldsymbol{k}), \mathsf{Enc}(\mathsf{mpk}, \boldsymbol{m}))$ outputs $\langle \boldsymbol{k}, \boldsymbol{m} \rangle \in \mathcal{R}$.

Remark 11. If instantiated from DDH or $\mathsf{HSM_{CL}}$, the IPFE of Fig. 1 yields schemes of [ALS16] and [CLT18]. Moreover, as detailed in Appx. E.3, though the construction is identical to the $\mathsf{ind\text{-}fe\text{-}cpa}$-secure construction of [BBL17], we significantly lower the bound on the adversary's advantage (when $\delta_{vs} \neq 0$).

Theorem 1. *Let $\mathcal{R}$ be a ring, either $\mathbf{Z}$ or $\mathbf{Z}/q\mathbf{Z}$ for some prime q; $\ell, a \in \mathbf{N}$; $\mathcal{SM} := (\widehat{\mathcal{X}}, \mathcal{X}, \widehat{\mathcal{L}}, \mathcal{W}, \mathsf{R})$ an instance of a $\delta_{\mathcal{L}}$-hard subgroup membership problem; and H the associated projective hash function. If H is $(\mathcal{R}, a, f, \aleph, \ell, \mathcal{M}, \mathcal{K}, \widehat{\Upsilon}, \Upsilon, \delta_{\mathcal{L}}, \delta_{vs})$-$\mathsf{pip\text{-}safe}$ then the IPFE scheme IPFE depicted in Fig. 1 is $\mathsf{ind\text{-}fe\text{-}cpa}$-secure, and $\mathsf{Adv}^{\mathsf{ind\text{-}fe\text{-}cpa}}_{\mathsf{IPFE},\mathcal{A}}(\lambda) \leqslant \delta_{\mathcal{L}} + \delta_{vs}$.*

Proof. We proceed via a sequence of games. *Game 0* is the original $\mathsf{ind\text{-}fe\text{-}cpa}$ experiment. *Game 2* is the final game, in which $\mathcal{A}$'s advantage is negligible. Let S_i denote the event "The output of *Game i* is 1".

```
1. Sample $(\mathsf{mpk}, \mathsf{msk}) := (\mathbf{hp}, \mathbf{hk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^\ell)$ and $\beta \hookleftarrow \{0,1\}$
2. Send $\mathsf{mpk}$ to $\mathcal{A}$ and answer pre-challenge phase key derivation queries
3. Receive $\boldsymbol{m}_0$, $\boldsymbol{m}_1$ from $\mathcal{A}$
4. Sample $(x_0, w) \leftarrow \mathsf{R}$ and let $c_0 := x_0$
5. $\boxed{\boxed{\text{Sample } \upsilon \hookleftarrow (\mathbf{Z}/\aleph\mathbf{Z})^* \text{ and overwrite } c_0 \leftarrow x_0 \cdot \varUpsilon^\upsilon \in \mathcal{X}\backslash\mathcal{L}}}$
6. For $1 \leqslant i \leqslant \ell$ :
7.     $\boxed{c_i := \mathsf{hash}(\mathsf{hk}_i, c_0) \cdot f^{m_{\beta,i}}}$
8. Let $\boldsymbol{ct} := (c_0, \boldsymbol{c})$
9. Send $\boldsymbol{ct}$ to $\mathcal{A}$ and answer post-challenge phase key derivation queries
10. Receive $\beta'$ from $\mathcal{A}$. If $(\beta = \beta')$ return 1, else return 0.
```

$\boxed{\text{Framed}}$ text highlights the evolution from *Game 0* to *Game 1.*

$\boxed{\boxed{\text{Double framed}}}$ text is only executed in *Game 2.*

Fig. 2: Security games for proof of Theorem 1.

Game 0. This is $\mathsf{Exp}^{\mathsf{ind\text{-}fe\text{-}cpa}}_{\mathsf{IPFE},\mathcal{A}}(\lambda)$, so $\mathsf{Adv}^{\mathsf{ind\text{-}fe\text{-}cpa}}_{\mathsf{IPFE},\mathcal{A}}(\lambda) = |\Pr[S_0\ ] - 1/2|$.
Game 1. $\mathcal{C}$ computes $\boldsymbol{ct}$ using the hash keys instead of the projection keys and the witness. Though computed differently, the values of the ciphertext components remain unchanged, as is $\mathcal{A}$'s success probability, hence

$$\Pr[S_0] = \Pr[S_1]. \tag{5}$$

Game 2. Here $\mathcal{C}$ samples c_0 at random from $\mathcal{X}\backslash\mathcal{L}$ instead of $\mathcal{L}$. Both games are indistinguishable under the $\delta_{\mathcal{L}}$-hardness of $\mathcal{SM}$, so

$$|\Pr[S_1] - \Pr[S_2\ ]| \leqslant \delta_{\mathcal{L}}. \tag{6}$$

Let us now bound the probability S_2 occurs. Observe that when $\mathcal{A}$ submits its' guess β' for β all the information it can use for its guess comes from (1) the public key mpk; (2) the challenge ciphertext $\boldsymbol{ct}$; and (3) key derivation queries.
Intuition. Following [ALS16]'s proof methodology, we first delimit the information leaked by $\boldsymbol{ct}$ by only considering the dimension in which both potential challenge ciphertexts differ. To this end we project this information onto the subspace generated by $\boldsymbol{m}_0 - \boldsymbol{m}_1$. We then consider the distribution of the projection of $\mathbf{hk}$ on the subspace generated by $\boldsymbol{m}_0 - \boldsymbol{m}_1$, conditionally on $\mathcal{A}$'s view (given the information fixed by key derivation queries and the public projection keys). Since $\mathcal{A}$ cannot query decryption keys for vectors $\boldsymbol{k}$ s.t. $\langle \boldsymbol{m}_0 - \boldsymbol{m}_1, \boldsymbol{k}\rangle \neq 0$, the δ_{vs}-vector-smoothness of H ensures that projecting $\mathbf{hk}$ onto the subspace generated by $\boldsymbol{m}_0 - \boldsymbol{m}_1$ induces a distribution $\{\mathsf{hash}(\langle \mathbf{hk}, \boldsymbol{m}_0 - \boldsymbol{m}_1\rangle, \upsilon) \mid \upsilon \in \langle \varUpsilon \rangle\}$ which is δ_{vs}-close to $\mathcal{U}(F)$, and thus $\boldsymbol{m}_\beta$ is statistically hidden in $\boldsymbol{c}$.
Details. First consider the challenge ciphertext; the decomposability and homomorphic properties of H allow to write its coordinates as: $c_0 = x_0 \cdot \varUpsilon^\upsilon \in \mathcal{X}\backslash\mathcal{L}$, where $\upsilon \in (\mathbf{Z}/\aleph\mathbf{Z})^*$, and $c_i = f^{m_{\beta,i}} \cdot \mathsf{hash}(\mathsf{hk}_i, x_0) \cdot \mathsf{hash}(\mathsf{hk}_i, \varUpsilon^\upsilon) \in \varPi$ for $i \in [\ell]$. Since this decomposition of c_0 is unique (by Def. 11), and since $x_0 \in \mathcal{L}$, information theoretically, for $i \in [\ell]$ the value $\mathsf{hash}(\mathsf{hk}_i, x_0)$ is fixed by the values hp_i and c_0. We denote $y_i := f^{m_{\beta,i}} \cdot \mathsf{hash}(\mathsf{hk}_i, \varUpsilon^\upsilon) \in F$ for $i \in [\ell]$. Any information fixed on β by c_i is thus contained in y_i, and it suffices to consider the information on the bit β which is leaked by $\boldsymbol{y} := (y_1, \ldots, y_\ell) \in F^\ell$.

Let $\mathbf{B}_{\boldsymbol{m}} \in \mathcal{R}^{\ell \times \ell}$ denote the matrix associated to $\boldsymbol{m} := \boldsymbol{m}_1 - \boldsymbol{m}_0$ built as per Lemma 1, whose rows we denote $(\boldsymbol{b}_1, \dots, \boldsymbol{b}_\ell)$. Since $\mathbf{B}_{\boldsymbol{m}}$ is invertible $\bmod \aleph$, $\boldsymbol{y} \in F^\ell$, and $\mathsf{ord}(F) = \aleph$, all the information contained in $\boldsymbol{y}$ is contained in the vector $(\prod_{i \in [\ell]} y_i^{b_{1,i}}, \dots, \prod_{i \in [\ell]} y_i^{b_{\ell,i}})$. It thus suffices to consider the information on β given by: $\prod_{i \in [\ell]} y_i^{b_{j,i}} = f^{\langle \boldsymbol{m}_\beta, \boldsymbol{b}_j \rangle} \mathsf{hash}(\lessdot \mathbf{hk}, \boldsymbol{b}_j \gtrdot_a, \varUpsilon^v)$ for $j \in [\ell]$. For $j \in [\ell - 1]$ we have $\boldsymbol{b}_j \in \boldsymbol{m}^\perp$, so the value of $f^{\langle \boldsymbol{m}_\beta, \boldsymbol{b}_j \rangle} \cdot \mathsf{hash}(\lessdot \mathbf{hk}, \boldsymbol{b}_j \gtrdot_a, \varUpsilon^v)$ provides no information on β. Let us now consider that contained in:

$$f^{\langle \boldsymbol{m}_\beta, \boldsymbol{b}_\ell \rangle} \cdot \mathsf{hash}(\lessdot \mathbf{hk}, \boldsymbol{b}_\ell \gtrdot_a, \varUpsilon^v) \tag{7}$$

To this end we evaluate the distribution of $\mathsf{hash}(\lessdot \mathbf{hk}, \boldsymbol{b}_\ell \gtrdot_a, \varUpsilon^v)$, and so we need to consider the information leaked on $\mathbf{hk}$ via key derivation queries. First observe that any key derivation query $\boldsymbol{k} \in \mathcal{K}$ must belong to $\boldsymbol{m}^\perp$, and so is a linear combination of vectors $(\boldsymbol{b}_1, \dots, \boldsymbol{b}_{\ell-1})$. We can thus apply the $\delta_{vs}(\ell)$-vector-smoothness over $\mathcal{X}$ on F of H, s.t. given $\boldsymbol{m}$, c_0, $\mathbf{hp}$ and $\lessdot \mathbf{hk}, \boldsymbol{b}_j \gtrdot_a$ for $j \in [\ell - 1]$, the distribution of $\mathsf{hash}(\lessdot \mathbf{hk}, \boldsymbol{b}_\ell \gtrdot_a, \varUpsilon^v)$ is δ_{vs}-close to $\mathcal{U}(F)$, and statistically hides $f^{\langle \boldsymbol{m}_\beta, \boldsymbol{b}_\ell \rangle}$ in Eq. (7), thereby hiding the bit β, and so: $|\Pr[S_2] - 1/2| \leqslant \delta_{vs}$ Combined with equations (5) and (6) we have: $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{ind\text{-}fe\text{-}cpa}} \leqslant \delta_{\mathcal{L}} + \delta_{vs}$, this concludes the proof that the IPFE scheme of Fig. 1 is ind-fe-cpa-secure. □

5 IPFE from PHFs secure against active adversaries

Let $\mathcal{R}$ be either the ring $\mathbf{Z}$ or $\mathbf{Z}/q\mathbf{Z}$ for some prime q; $\mathsf{Gen}_{\mathcal{SM}}$ be a subgroup membership problem generator outputting the description $\mathcal{SM} := (\widehat{\mathcal{X}}, \mathcal{X}, \widehat{\mathcal{L}}, \mathcal{W}, \mathsf{R})$ of an SMP; ℓ and a be positive integers; $\mathcal{M} \subseteq \mathcal{R}^\ell$ be the plaintext space; $\mathcal{K} \subseteq \mathcal{R}^\ell$ be the space from which keys are derived; $(\mathsf{H}, \mathsf{eH})$ a pair of $(\mathcal{R}, a, f, \aleph, \ell, \mathcal{M}, \mathcal{K}, \widehat{\varUpsilon}, \varUpsilon, \delta_{\mathcal{L}}, \delta_{vs}, \delta_{vu})$-aip-safe PHFs; $\mathcal{H}$ a family of CRHF such that $\hbar \hookleftarrow \mathcal{H}$ maps $\{0,1\}^*$ to the efficiently recognisable set E; and $\mathsf{OTS} := (\mathsf{Setup}, \mathsf{Sign}, \mathsf{Verif})$ a strongly unforgeable OTS scheme. The resulting ind-fe-cca-secure FE scheme, depicted in Fig. 3, recovers $\langle \boldsymbol{m}, \boldsymbol{k} \rangle \in \mathcal{R}$ for $\boldsymbol{m} \in \mathcal{M}$, $\boldsymbol{k} \in \mathcal{K}$. Instantiations from $\mathsf{HSM}_{\mathsf{CL}}$, DDH and DCR are detailed in Appx. D.

Note that the auxiliary input space E of eH is the output of a CRHF. This ensures that adversaries – given a ciphertext computed from some $e \in E$ and $c_0 \in \mathcal{X}$ – cannot compute a *different* ciphertext for the same e and c_0. Indeed if it could do so, it needn't compute the hash values c_i for $i \in [\ell]$, since once could reuse those from the given ciphertext, and thereby break ciphertext integrity.

Correctness. As $K_{\mathsf{hk}} = \mathcal{R}^a$ and $K_{\mathsf{ehk}} = \mathcal{R}^{2a}$, one has $\mathbf{hk} \in (\mathcal{R}^a)^\ell$, and $\mathbf{ehk} \in (\mathcal{R}^{2a})^\ell$, so $\lessdot \mathbf{hk}, \boldsymbol{k} \gtrdot_a \in \mathcal{R}^a$ and $\lessdot \mathbf{ehk}, \boldsymbol{k} \gtrdot_{2a} \in \mathcal{R}^{2a}$. Next, by correctness of OTS, ciphertexts output by Enc pass the check on line 2 of Dec. Moreover by the correctness and key homomorphism of eH:

$$\mathsf{ehash}(\overline{\mathsf{sk}}_{\boldsymbol{k}}, c_0, e) = \prod_{i \in [\ell]} \mathsf{eprojhash}(\mathsf{ehp}_i, c_0, w, e)^{k_i} = \prod_{i \in [\ell]} \overline{c}_i^{k_i}.$$

Now correctness follows from that of the ind-fe-cca construction of Section 4.

```
Setup($1^\lambda, 1^\ell$):
  Sample $\mathcal{SM} \leftarrow \mathsf{Gen}_{\mathcal{SM}}(1^\lambda)$
  Sample $\hbar \hookleftarrow \mathcal{H}$
  For $1 \leqslant i \leqslant \ell$ :
      Sample $\mathsf{hk}_i \leftarrow \mathsf{hashkg}(\mathcal{SM})$
      $\mathsf{hp}_i \leftarrow \mathsf{projkg}(\mathsf{hk}_i)$
      Sample $\mathsf{ehk}_i \leftarrow \mathsf{ehashkg}(\mathcal{SM})$
      $\mathsf{ehp}_i \leftarrow \mathsf{eprojkg}(\mathsf{ehk}_i)$
  Return $\mathsf{mpk} := (\mathbf{hp}, \mathbf{ehp}, \hbar)$; $\mathsf{msk} := (\mathbf{hk}, \mathbf{ehk})$
```

```
KeyDer($\mathsf{msk}, \boldsymbol{k}$):
  If $\boldsymbol{k} \notin \mathcal{K}$ return $\perp$
  $\mathsf{sk}_{\boldsymbol{k}} \leftarrow \lessdot \mathbf{hk}, \boldsymbol{k} \gtrdot_a$
  $\overline{\mathsf{sk}}_{\boldsymbol{k}} \leftarrow \lessdot \mathbf{ehk}, \boldsymbol{k} \gtrdot_{2a}$
  Return $(\mathsf{sk}_{\boldsymbol{k}}, \overline{\mathsf{sk}}_{\boldsymbol{k}}, \boldsymbol{k})$
```

```
Enc($\mathsf{mpk}, \boldsymbol{m}$):
  If $\boldsymbol{m} \notin \mathcal{M}$ return $\perp$
  $(\mathsf{sk}_{\mathsf{OTS}}, \mathsf{vk}) \leftarrow \mathsf{OTS.Setup}(1^\lambda)$
  $e \leftarrow \hbar(\mathsf{vk})$
  Sample $(c_0, w) \leftarrow \mathsf{R}$
  For $1 \leqslant i \leqslant \ell$ :
      $c_i \leftarrow \mathsf{projhash}(\mathsf{hp}_i, c_0, w) \cdot f^{m_i}$
      $\overline{c}_i \leftarrow \mathsf{eprojhash}(\mathsf{ehp}_i, c_0, w, e)$
  Set $\boldsymbol{ct} := (c_0, \boldsymbol{c}, \overline{\boldsymbol{c}})$
  $\sigma \leftarrow \mathsf{OTS.Sign}(\mathsf{sk}_{\mathsf{OTS}}, \boldsymbol{ct})$
  Return $(\boldsymbol{ct}, \mathsf{vk}, \sigma)$
```

```
Dec($\mathsf{mpk}, (\mathsf{sk}_{\boldsymbol{k}}, \overline{\mathsf{sk}}_{\boldsymbol{k}}, \boldsymbol{k}), (\boldsymbol{ct}, \mathsf{vk}, \sigma)$):
  If $\boldsymbol{ct} \notin \widehat{\mathcal{X}} \times \Pi^{2\ell}$ then return $\perp$
  If $\mathsf{OTS.Verif}(\mathsf{vk}, \boldsymbol{ct}, \sigma) = 0$
      Then return $\perp$
  $e \leftarrow \hbar(\mathsf{vk})$
  Parse $(c_0, \boldsymbol{c}, \overline{\boldsymbol{c}}) = \boldsymbol{ct}$
  If $\mathsf{ehash}(\overline{\mathsf{sk}}_{\boldsymbol{k}}, c_0, e) \neq \prod_{i \in [\ell]} \overline{c}_i^{k_i}$
      Then return $\perp$
  $M \leftarrow (\prod_{i \in [\ell]} c_i^{k_i}) \mathsf{hash}(\mathsf{sk}_{\boldsymbol{k}}, c_0)^{-1}$
  If $M \notin F$ then return $\perp$
  Return $\mathsf{sol} = \log_f(M)$
```

Fig. 3: IPFE that is ind-fe-cca-secure from projective hash functions

Theorem 2. *Let $\mathcal{R}$ be a ring, either $\mathbf{Z}$ or $\mathbf{Z}/q\mathbf{Z}$ for some prime q; $\ell, a \in \mathbf{N}$; $\mathcal{SM} := (\widehat{\mathcal{X}}, \mathcal{X}, \widehat{\mathcal{L}}, \mathcal{W}, \mathsf{R})$ be an instance of a $\delta_{\mathcal{L}}$-hard SMP; H be the associated PHF; eH be the resulting EPHF*[6]*; $\mathcal{H}$ be a family of $\epsilon_{\hbar}$-collision resistant hash functions; and $\mathsf{OTS} := (\mathsf{Setup}, \mathsf{Sign}, \mathsf{Verif})$ be an ϵ_{OTS}-strongly unforgeable one time signature scheme. If the pair $(\mathsf{H}, \mathsf{eH})$ is $(\mathcal{R}, a, f, \aleph, \ell, \mathcal{M}, \mathcal{K}, \widehat{\Upsilon}, \Upsilon, \delta_{\mathcal{L}}, \delta_{vs}, \delta_{vu})$-aip-safe then, denoting q_{dec} an upper bound on the number of decryption queries made by the adversary $\mathcal{A}$ for ind-fe-cca security, the IPFE scheme IPFE depicted in Fig. 3 is ind-fe-cca-secure. Precisely: $\mathsf{Adv}^{\mathsf{ind\text{-}fe\text{-}cca}}_{\mathsf{IPFE},\mathcal{A}}(\lambda) \leqslant \delta_{\mathcal{L}} + q_{\mathsf{dec}}(\frac{\aleph}{\aleph - q_{\mathsf{dec}} + 1} \cdot \delta_{vu} + \epsilon_{\hbar} + \epsilon_{\mathsf{OTS}}) + \delta_{vs}$.*

Remark 12. Before proving Thm. 2, we briefly compare to the results of [BBL17]. For details see Appx. E; for comparisons of concrete instantiations see Section 6. They bound the adversary's advantage[7] by $\delta_{\mathsf{bbl}} = \delta_{\mathcal{L}} + |\Delta\mathcal{M}|(\delta_{vs} + \delta_{vu}) + q_{\mathsf{dec}}|\Delta\mathcal{M}|(\epsilon_{\hbar} + \epsilon_{\mathsf{OTS}})$, where $|\Delta\mathcal{M}| \leqslant (4(\frac{\aleph}{2\ell})^{1/2})^{\ell}$. In our work, as $q_{\mathsf{dec}} \leqslant \mathsf{poly}(\lambda)$, whereas $\aleph$ is exponential in λ, this advantage is upper bounded by $\delta_{\mathsf{us}} = \delta_{\mathcal{L}} + (\delta_{vs} + q_{\mathsf{dec}}\delta_{vu}) + q_{\mathsf{dec}}(\epsilon_{\hbar} + \epsilon_{\mathsf{OTS}})$. Since $|\Delta\mathcal{M}|$ grows exponentially with ℓ, and polynomially with $\aleph$ (which itself is exponential in λ), they require stronger properties from the underlying PHFs to compensate for the factor $|\Delta\mathcal{M}|$.

To give an example, for $\aleph$ of 128 bits, $\ell = 100$, and allowing the adversary to make $q_{\mathsf{dec}} = 2^{20}$ queries, one gets $|\Delta\mathcal{M}| = 2^{6218}$, and $\delta_{\mathsf{bbl}} = \delta_{\mathcal{L}} + 2^{6218}(\delta_{vs} + \delta_{vu}) + 2^{6238}(\epsilon_{\hbar} + \epsilon_{\mathsf{OTS}})$, whereas in this work $\delta_{\mathsf{us}} = \delta_{\mathcal{L}} + (\delta_{vs} + 2^{20}\delta_{vu}) + 2^{20}(\epsilon_{\hbar} + \epsilon_{\mathsf{OTS}})$. We note that even if hashing keys are sampled uniformly, which implies $\delta_{vs} = 0$, our security proof significantly reduces $\mathcal{A}$'s advantage, which allows us to use smaller keys, and significantly gain in efficiency.

[6] As per the generic construction of [CS02], *cf.* Appx. B

[7] To simplify the comparison, we neglect a factor q_{dec} in their favour.

To prove Thm 2 we first define *valid* and *invalid* decryption queries: valid and rejected decryption queries reveal negligibly more information on the hash keys than what an adversary $\mathcal{A}$ could gain from projection keys and key derivation queries (*cf.* Lemmas 7, 8). On the other hand the probability an invalid query was not rejected, thereby potentially leaking relevant information, is negligible (*cf.* Lemma 9). The analysis of this probability is conditioned on $\mathcal{A}$'s view, and performed *a posteriori*, i.e. when $\mathcal{A}$ guesses the challenge bit. Hence the notions of *valid* and *invalid* decryption queries can depend on the challenge messages (even though $\mathcal{A}$ may request decryption queries *before* choosing $\boldsymbol{m}_0$ and $\boldsymbol{m}_1$).

Notation. Consider a decryption query $(\mathsf{decrypt}, (\boldsymbol{ct}, \mathsf{vk}, \sigma), \boldsymbol{k})$ performed by $\mathcal{A}$, where $\boldsymbol{ct} = (c_0, \boldsymbol{c}, \overline{\boldsymbol{c}})$, and $\boldsymbol{k} \in \mathcal{K}$. After the post-challenge phase of $\mathsf{Exp}^{\mathsf{ind\text{-}fe\text{-}cca}}_{\mathsf{IPFE},\mathcal{A}}$ (*cf.* Section 2.4), one can categorise the query. It is said to be *valid* if either (1) $c_0 \in \widehat{\mathcal{L}}$, or (2) $\langle \boldsymbol{k}, \boldsymbol{m}_0 \rangle = \langle \boldsymbol{k}, \boldsymbol{m}_1 \rangle$ where $\boldsymbol{m}_0$, $\boldsymbol{m}_1$ are the challenge messages. Any decryption query which is not valid is said to be *invalid.*

Proof of Thm 2. We proceed via a sequence of games. *Game 0* is experiment $\mathsf{Exp}^{\mathsf{ind\text{-}fe\text{-}cca}}_{\mathsf{IPFE},\mathcal{A}}$. In *Game 2*, with overwhelming probability all the information revealed by decryption queries is contained in that revealed by public keys, the challenge ciphertext and key derivation queries. We then conclude as in the ind-fe-cpa setting. The evolution of these games is highlighted in Fig. 4, *Game 0* is not depicted, as it follows immediately from the security definition. Let S_i denote the event "The output of *Game i* is 1".

```
1. Sample $\beta \hookleftarrow \{0,1\}$ and $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^\ell)$
2. Send $\mathsf{mpk}$ to $\mathcal{A}$ and answer pre-challenge phase queries
3. Receive $\boldsymbol{m}_0$, $\boldsymbol{m}_1$ from $\mathcal{A}$
4. Sample $(\mathsf{sk_{OTS}}, \mathsf{vk}) \hookleftarrow \mathsf{OTS.Setup}(1^\lambda)$ and let $e \leftarrow \hbar(\mathsf{vk})$
5. Sample $(x_0, w) \leftarrow \mathsf{R}$ and let $c_0 := x_0$
6. $\boxed{\boxed{\text{Sample } \upsilon \hookleftarrow (\mathbf{Z}/\aleph\mathbf{Z})^* \text{ and overwrite } c_0 \leftarrow x_0 \cdot \Upsilon^{\upsilon} \in \mathcal{X}\backslash\mathcal{L}}}$
7. For $1 \leqslant i \leqslant \ell$ :
8.     $\boxed{c_i := \mathsf{hash}(\mathsf{hk}_i, c_0) \cdot f^{m_{\beta,i}}}$ and $\boxed{\overline{c}_i := \mathsf{ehash}(\mathsf{ehk}_i, c_0, e)}$
9. Let $\boldsymbol{ct} := (c_0, \boldsymbol{c}, \overline{\boldsymbol{c}})$ and compute $\sigma \leftarrow \mathsf{OTS.Sign}(\mathsf{sk_{OTS}}, \boldsymbol{ct})$
10. Send $(\boldsymbol{ct}, \mathsf{vk}, \sigma)$ to $\mathcal{A}$ and answer post-challenge phase queries
11. Receive $\beta'$ from $\mathcal{A}$. If $(\beta = \beta')$ return 1, else return 0.
```

$\boxed{\text{Framed}}$ text highlights the evolution from *Game 0* to *Game 1.*

$\boxed{\boxed{\text{Double framed}}}$ text is only executed in *Game 2.*

Fig. 4: Evolution of security games for proof of Theorem 2.

Game 0. The original ind-fe-cca game, so $\mathsf{Adv}^{\mathsf{ind\text{-}fe\text{-}cca}}_{\mathsf{IPFE},\mathcal{A}}(\lambda) = |\Pr[S_0] - 1/2|$.
Game 1. $\mathcal{C}$ computes $\boldsymbol{ct}$ using the hash keys instead of the projection keys and the witness. $\mathcal{A}$'s view remains unchanged: $\Pr[S_0] = \Pr[S_1]$.
Game 2. Here $\mathcal{C}$ samples c_0 at random from $\mathcal{X}\backslash\mathcal{L}$ instead of $\mathcal{L}$. Both games are indistinguishable under the $\delta_{\mathcal{L}}$-hardness of $\mathcal{SM}$, and then $|\Pr[S_1] - \Pr[S_2]| \leqslant \delta_{\mathcal{L}}$.

Let us now bound the probability S_2 occurs. When $\mathcal{A}$ submits β', all $\mathcal{A}$'s queries are valid or invalid, as the post-challenge phase is over. Since $\mathcal{A}$ has finished collecting information, one can analyse $\mathcal{A}$'s probability of guessing β conditioned on this information; which comes from: (1) the public key mpk; (2) the challenge ciphertext $\boldsymbol{ct}$; (3) key derivation queries; (4) decryption queries.

Intuition. We first upper bound the probability any invalid decryption query was *not* rejected. Then, assuming all invalid queries were rejected, we demonstrate that all decryption queries (either valid or rejected) provide no further information to $\mathcal{A}$ than that revealed by projection keys and key derivation queries. We can then reduce the $\mathsf{ind\text{-}fe\text{-}cca}$-security of the protocol of Fig. 3 to the $\mathsf{ind\text{-}fe\text{-}cpa}$-security of the protocol of Fig. 1, only where mpk also contains $\widehat{\mathbf{hp}} := \widehat{\mathsf{projkg}}(\mathsf{hk})$.

Bounding the information revealed by decryption queries. Let $\textsc{Bad}$ denote the event an invalid decryption query was not rejected; $\overline{\textsc{Bad}}$ the complement event; and q_{dec} an upper bound on the number of decryption queries performed by $\mathcal{A}$. From Lemma 9, we have: $\Pr[\textsc{Bad}] \leqslant q_{\mathsf{dec}}(\frac{\aleph}{\aleph - q_{\mathsf{dec}}+1} \cdot \delta_{vu} + \epsilon_{\hbar} + \epsilon_{\mathsf{OTS}})$.

It holds that $\Pr[S_2] = \Pr[S_2 \wedge \textsc{Bad}] + \Pr[S_2 \wedge \overline{\textsc{Bad}}] \leqslant \Pr[\textsc{Bad}] + \Pr[S_2 \wedge \overline{\textsc{Bad}}]$. We hereafter bound $\Pr[S_2 \wedge \overline{\textsc{Bad}}]$. We thus assume that $\textsc{Bad}$ does not occur.

Claim. $|\Pr[S_2 \wedge \overline{\textsc{Bad}}] - 1/2| \leqslant \delta_{vs}$.

Proof of claim. From Lemmas 7 and 8, decryption queries which do not cause $\textsc{Bad}$ to occur provide no further information to $\mathcal{A}$ on $\mathbf{hk}$ than what it can obtain from $\widehat{\mathbf{hp}} := \widehat{\mathsf{projkg}}(\mathbf{hk})$ and key derivation requests. Moreover, the key $\mathbf{hk}$ used to mask the bit β is sampled independently from $\mathbf{ehk}$ which has no influence on $\mathcal{A}$'s view of $\mathbf{hk}$. So to analyse $\mathcal{A}$'s view of β, it suffices to consider the distribution of $\mathbf{hk}$ from $\mathcal{A}$'s view, and given this distribution the information revealed by $(c_0, \boldsymbol{c})$ on β (we can ignore $\overline{\boldsymbol{c}}$ for this analysis). It thus suffices to prove the $\mathsf{ind\text{-}fe\text{-}cpa}$ security of a reduced version of the scheme, identical to that of Fig. 1, only where mpk also contains $\widehat{\mathbf{hp}} := \widehat{\mathsf{projkg}}(\mathsf{hk})$ (at a high level, we have proven ciphertext integrity; we now prove the ciphertext ensures confidentiality).

The only difference between proving $\mathsf{ind\text{-}fe\text{-}cpa}$-security of the reduced scheme and that of Fig. 1 is that $\mathcal{A}$ is granted the projection keys in $\widehat{\mathbf{hp}}$. This information is only used when using the δ_{vs}-vector smoothness of H, which by definition holds even given $\widehat{\mathbf{hp}}$. Hence $|\Pr[S_2 \wedge \overline{\textsc{Bad}}] - 1/2| \leqslant \delta_{vs}$; and the claim holds. □

From the above claim, it holds that if $\textsc{Bad}$ does not occur, the bit β is statistically hidden from $\mathcal{A}$. Combining previous equations concludes proof of Thm 2: $\mathsf{Adv}^{\mathsf{ind\text{-}fe\text{-}cca}}_{\mathcal{A}} \leqslant \delta_{\mathcal{L}} + q_{\mathsf{dec}}(\frac{\aleph}{\aleph - q_{\mathsf{dec}}+1} \cdot \delta_{vu} + \epsilon_{\hbar} + \epsilon_{\mathsf{OTS}}) + \delta_{vs}$. □

Lemmas 7, 8, 9 for the scheme of Fig. 3 hold for all games in proof of Thm. 2.

Lemma 7. *It holds that – in all security games of proof of Thm. 2 – valid decryption queries performed by $\mathcal{A}$ provide no further information on $\mathbf{hk}$ or $\mathbf{ehk}$ than what can be deduced from the value of the projection keys $\widehat{\mathbf{hp}} := \widehat{\mathsf{projkg}}(\mathbf{hk})$, $\widehat{\mathbf{ehp}} := \widehat{\mathsf{eprojkg}}(\mathbf{ehk})$, and key derivation queries.*

Proof. The lemma holds for all security games of proof of Thm. 2 since the projection keys $\widehat{\mathsf{\mathbf{hp}}}$ and $\widehat{\mathsf{\mathbf{ehp}}}$ (which fix the value of the scheme's public keys $\mathsf{\mathbf{hp}} = \mathsf{projkg}(\mathsf{\mathbf{hk}})$ and $\mathsf{\mathbf{ehp}} = \mathsf{eprojkg}(\mathsf{\mathbf{ehk}})$), and the information $\mathscr{A}$ can learn from key derivation requests do not change throughout the games. Consider a query $(\mathsf{decrypt}, (\boldsymbol{ct}, \mathsf{vk}, \sigma), \boldsymbol{k})$, where $\boldsymbol{ct} = (c_0, \boldsymbol{c}, \overline{\boldsymbol{c}})$, and $\boldsymbol{k} \in \mathscr{K}$. Since the query is valid one of the following cases apply:

Case (1): $c_0 \in \widehat{\mathscr{L}}$. The Dec algorithm uses $\mathsf{\mathbf{ehk}}$ on line 6 to compute $\mathsf{ehash}(\leqslant\mathsf{\mathbf{ehk}}, \boldsymbol{k}\geqslant_{2a}, c_0, e)$, where $e = \hbar(\mathsf{vk})$, and $\mathsf{\mathbf{hk}}$ on line 8, to compute $\mathsf{hash}(\leqslant\mathsf{\mathbf{hk}}, \boldsymbol{k}\geqslant_a, c_0)$. By correctness of the PHFs, and by the key homomorphism the projective hash functions it holds that:

$$\mathsf{hash}(\leqslant\mathsf{\mathbf{hk}}, \boldsymbol{k}\geqslant_a, c_0) = \widehat{\mathsf{projhash}}(\leqslant\widehat{\mathsf{\mathbf{hp}}}, \boldsymbol{k}\geqslant_a, c_0)$$

and

$$\mathsf{ehash}(\leqslant\mathsf{\mathbf{ehk}}, \boldsymbol{k}\geqslant_{2a}, c_0, e) = \widehat{\mathsf{eprojhash}}(\leqslant\widehat{\mathsf{\mathbf{ehp}}}, \boldsymbol{k}\geqslant_{2a}, c_0, e).$$

The above values are fixed by $\boldsymbol{k}$, c_0, $\widehat{\mathsf{\mathbf{hp}}}$ and $\widehat{\mathsf{\mathbf{ehp}}}$. So information theoretically, no more information is fixed on $\mathsf{\mathbf{hk}}$ or $\mathsf{\mathbf{ehk}}$.

Case (2): $\langle \boldsymbol{k}, \boldsymbol{m}_0 \rangle = \langle \boldsymbol{k}, \boldsymbol{m}_1 \rangle$ in $\mathscr{R}$, s.t. $\mathscr{A}$ can query a key derived from $\boldsymbol{k}$ and run the decryption algorithm itself. □

Lemma 8. *In all security games of proof of Thm. 2, decryption queries* $(\mathsf{decrypt}, (\boldsymbol{ct}, \mathsf{vk}, \sigma), \boldsymbol{k})$ *which are rejected provide no information on* $\mathsf{\mathbf{hk}}$.

Proof. Let $\boldsymbol{ct} = (c_0, \boldsymbol{c}, \overline{\boldsymbol{c}})$ where $c_0 \in \widehat{\mathscr{X}}$, and $\boldsymbol{c}, \overline{\boldsymbol{c}} \in \Pi^\ell$. Since Lemma 7 ensures valid decryption queries do not leak information, it suffices to consider invalid decryption queries. If the rejection is due to the OTS verification algorithm, $\mathscr{A}$ learns nothing on $\mathsf{\mathbf{ehk}}$ or $\mathsf{\mathbf{hk}}$. Now suppose the rejection is due to $\mathsf{ehash}(\overline{\mathsf{sk}}_{\boldsymbol{k}}, c_0, e) \neq \prod_{i\in[\ell]} \overline{c}_i^{k_i}$. Each such rejected decryption request – information theoretically – provides $\mathscr{A}$ with an inequality on $\mathsf{\mathbf{ehk}}$, however $\mathsf{\mathbf{hk}}$ is sampled independently from $\mathsf{\mathbf{ehk}}$, consequently $\mathsf{\mathbf{ehk}}$ has no influence on $\mathscr{A}$'s view of $\mathsf{\mathbf{hk}}$. Thus this rejection does not leak any information on $\mathsf{\mathbf{hk}}$. □

Lemma 9 bounds the probability invalid decryption queries are not rejected.

Lemma 9. *Assume* OTS *is an* ϵ_{OTS}*-strongly unforgeable one time signature scheme and* $\mathscr{H}$ *is a family of* $\epsilon_{\hbar}$*-CRHF.* Bad *denotes the event that, by the end of the post-challenge phase,* $\mathscr{A}$ *has performed an invalid query* $(\mathsf{decrypt}, (\boldsymbol{ct'}, \mathsf{vk}', \sigma'), \boldsymbol{k'})$ *to which* $\mathcal{C}$ *does not answer* $\perp$*. Denoting* q_{dec} *an upper bound on the number of decryption queries performed by* $\mathscr{A}$ *it holds that:*

$$\Pr[\text{Bad}] \leqslant q_{\mathsf{dec}}\left(\frac{\aleph}{\aleph - q_{\mathsf{dec}} + 1} \cdot \delta_{vu} + \epsilon_{\hbar} + \epsilon_{\mathsf{OTS}}\right).$$

Proof. First observe the probability Bad occurs is the conditional probability $\mathscr{A}$ produces an invalid ciphertext which does not cause the decryption algorithm to return $\perp$, given A's view. Furthermore, only the challenge ciphertext $\boldsymbol{ct}$ differs between games 0, 1 and 2. In games 0 and 1, since it is computed from $c_0 \in \mathscr{L}$, it leaks no further information on the hashing keys than that revealed by the public projection keys. We thus bound the probability BAD occurs in game 2, since this is the scenario where $\mathscr{A}$ has the most information on the hashing keys.

The proof proceeds in two steps. (1) We first use the vector universality of eH to bound the probability the event BAD occurs for $q_{\mathsf{dec}} = 1$, i.e. the probability that $\mathscr{A}$'s first invalid decryption query is not rejected by the decryption oracle. (2) Next we take into account the information leaked by the rejection of such decryption queries. This allows us to bound the probability the event BAD occurs after q_{dec} decryption queries, for $q_{\mathsf{dec}} \geqslant 1$.

Since $\mathbf{hk}$ is sampled independently from $\mathbf{ehk}$, and has no influence on the latters' value, it suffices to consider the distribution of $\mathbf{ehk}$ from $\mathscr{A}$'s view, and in this proof, we ignore that of $\mathbf{hk}$. Moreover the value of the public key $\mathbf{ehp} := \mathsf{eprojkg}(\mathbf{ehk})$ is fixed by that of projection keys $\widehat{\mathbf{ehp}} := \widehat{\mathsf{eprojkg}}(\mathbf{ehk})$. We consider $\mathscr{A}$'s success probability given $\widehat{\mathbf{ehp}}$, which is at least that given $\mathbf{ehp}$.

Let $M \in \Delta\mathcal{M}$ be a random variable ($\mathscr{A}$'s possible choices for $\boldsymbol{m}_0 - \boldsymbol{m}_1$); and $\mathbf{B}_M \in \mathscr{R}^{\ell\times\ell}$ be the associated matrix. The current analysis is performed *à posteriori*, which implies $\boldsymbol{m}_0$ and $\boldsymbol{m}_1$ are fixed, so denoting $\boldsymbol{m} := (\boldsymbol{m}_1 - \boldsymbol{m}_0) \in \mathscr{R}^\ell$ we condition $\mathscr{A}$'s success probability on $M = \boldsymbol{m}$.

Step 1. Before its first invalid decryption query, $\mathscr{A}$ has access to:

1. the vector $\mathbf{ehp}$, whose value is fixed by that of $\widehat{\mathbf{ehp}}$;
2. the challenge ciphertext, which fixes $c_0 \in \mathscr{X}\backslash\mathscr{L}$, $e = \hbar(\mathsf{vk}) \in E$; and the evaluation $\overline{c}_i = \mathsf{ehash}(\mathsf{ehk}_i, c_0, e)$ for $i \in [\ell]$;
3. key derivation queries. Any query $\boldsymbol{k} \in \mathscr{K}$ is a linear combination of the top rows $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{\ell-1}$ of $\mathbf{B}_{\boldsymbol{m}}$. Thus, for fixed $\boldsymbol{v}_j \in \mathscr{R}^{2a}$, $j \in [\ell-1]$, the information leaked via key derivation queries on $\mathbf{ehk}$ is upper bounded by the information given by equalities: $\lessdot \mathbf{ehk}, \boldsymbol{b}_j \gtrdot_{2a} = \boldsymbol{v}_j$.
4. valid decryption queries, which reveal no more information than the value $\widehat{\mathbf{ehp}}$ and key derivation queries (*cf.* Lemma 7);

Now consider query $(\mathsf{decrypt}, (\boldsymbol{ct'}, \mathsf{vk}', \sigma'), \boldsymbol{k'})$, where $(c_0', \boldsymbol{c'}, \overline{\boldsymbol{c}}') := \boldsymbol{ct'}$ and $e' := \hbar(\mathsf{vk}')$. Assume this is $\mathscr{A}$'s first *invalid* decryption query. As such it satisfies (1) $c_0' \in \widehat{\mathscr{X}}\backslash\widehat{\mathscr{L}}$ and (2) $\boldsymbol{k'} \notin \boldsymbol{m}^\perp$. Now if we let $\pi := \prod_{i\in[\ell]}(\overline{c}_i')^{k_i'}$ we can apply the δ_{vu}-vector-universality of eH, which ensures that as long as $(c_0, e) \neq (c_0', e')$, the probability (conditioned on $\mathscr{A}$'s view) that this first invalid decryption query satisfies $\mathsf{Dec}(\mathsf{mpk}, \mathsf{KeyDer}\ (\mathsf{msk}, \boldsymbol{k'}), (\boldsymbol{ct'}, \mathsf{vk}', \sigma')) \neq \perp$ is upper bounded by δ_{vu}.

Let us now bound the probability that $(c_0, e) = (c_0', e')$. Since $\mathscr{A}$ cannot query the challenge ciphertext to the decryption oracle, it must hold that $(\boldsymbol{ct'}, \mathsf{vk}', \sigma') \neq (\boldsymbol{ct}, \mathsf{vk}, \sigma)$. We consider three cases: (1) $\mathsf{vk}' = \mathsf{vk}$ and $(e = e' \mod \aleph)$ but $(\boldsymbol{ct}, \sigma) \neq (\boldsymbol{ct'}, \sigma')$. In this case, either $\mathsf{OTS.Verif}(\mathsf{vk}, \boldsymbol{ct'}, \sigma') = 0$ and the oracle rejects, or $\mathsf{OTS.Verif}(\mathsf{vk}, \boldsymbol{ct'}, \sigma') = 1$, s.t. $\mathscr{A}$ has forged a signature for $\boldsymbol{ct'}$ with verification key vk, thus breaking the unforgeability of OTS. This occurs with probability $\leqslant \epsilon_{\mathsf{OTS}}$; (2) $\mathsf{vk}' \neq \mathsf{vk}$ but $(e = e' \mod \aleph)$. In this case we have found a collision for $\hbar$. This occurs with probability $\leqslant \epsilon_{\hbar}$; (3) $\mathsf{vk}' \neq \mathsf{vk}$ and $e \neq e' \mod \aleph$, in which case $(c_0, e) \neq (c_0', e')$, and as proved above, $\mathscr{A}$ has probability $\leqslant \delta_{vu}$ that the oracle does not reject.

Thus the probability this first invalid decryption query $(\mathsf{decrypt}, (\boldsymbol{ct'}, \mathsf{vk}', \sigma'), \boldsymbol{k'})$ is not rejected is upper bounded by $\delta_{vu} + \epsilon_{\hbar} + \epsilon_{\mathsf{OTS}}$. It now remains to consider the information revealed by each rejected invalid decryption query.

Step 2: Rejected invalid decryption queries. Consider an invalid query $(\mathsf{decrypt}, (\boldsymbol{ct'}, \mathsf{vk}', \sigma'), \boldsymbol{k'})$ which is rejected. Let $(c_0', \boldsymbol{c'}, \overline{\boldsymbol{c}}') := \boldsymbol{ct'}$ and $\pi' := \prod_{i \in [\ell]} \overline{c}_i'^{k_i'}$. If the rejection is due to the OTS verification algorithm, $\mathcal{A}$ learns nothing about $\mathbf{ehk}$. Now suppose the rejection is due to $\mathsf{ehash}(\lessdot \mathbf{ehk}, \boldsymbol{k'} \gtrdot_{2a}, c_0', e') \neq \pi'$. Since $c_0' \notin \widehat{\mathcal{L}}$, and eH is $(\widehat{\Upsilon}, \Upsilon, F)$-decomposable, there exist unique $x' \in \widehat{\mathcal{L}}$ and $y' \in \langle \widehat{\Upsilon} \rangle$ such that $c_0' = x' \cdot y'$. Let us parse $(\mathbf{hk}_0, \mathbf{hk}_1) := \mathbf{ehk}$, where $\mathbf{hk}_0, \mathbf{hk}_1 \in K_{\mathsf{hk}}$. By the homomorphic properties of H, and the generic construction for eH from H we can write:

$$\begin{aligned}\mathsf{ehash}(\lessdot \mathbf{ehk}, \boldsymbol{k'} \gtrdot_{2a}, c_0', e') = {} & \mathsf{hash}(\lessdot \mathbf{hk}_0, \boldsymbol{k'} \gtrdot_a, x')\mathsf{hash}(\lessdot \mathbf{hk}_0, \boldsymbol{k'} \gtrdot_a, y') \\ & \cdot \big(\mathsf{hash}(\lessdot \mathbf{hk}_1, \boldsymbol{k'} \gtrdot_a, x')\mathsf{hash}(\lessdot \mathbf{hk}_1, \boldsymbol{k'} \gtrdot_a, y')\big)^{\Gamma(c_0', e')}\end{aligned}$$

where information theoretically, the value of $\mathsf{hash}(\lessdot \mathbf{hk}_0, \boldsymbol{k'} \gtrdot_a, x') \cdot \mathsf{hash}(\lessdot \mathbf{hk}_1, \boldsymbol{k'} \gtrdot_a, x'))^{\Gamma(c_0', e')}$ is already fixed by the projection keys and c_0'. Consequently, the information revealed by the rejection of this query amounts to ruling out one possible value for $\mathsf{hash}(\lessdot \mathbf{hk}_0, \boldsymbol{k'} \gtrdot_a, y') \cdot (\mathsf{hash}(\lessdot \mathbf{hk}_1, \boldsymbol{k'} \gtrdot_a, y'))^{\Gamma(c_0', e')} \in F$, and thereby a proportion $1/\aleph$ of the possible values for $\lessdot \mathbf{ehk}, \boldsymbol{k'} \gtrdot_{2a} \bmod \aleph$.

Thus the probability that $\mathcal{A}$'s i^{th} invalid query is not answered by $\perp$ is upper bounded by $(\frac{\aleph}{\aleph - (i+1)} \cdot \delta_{vu} + \epsilon_{\hbar} + \epsilon_{\mathsf{OTS}})$. This allows us to conclude that, after q_{dec} decryption queries, the probability the event Bad occurs is upper bounded by:

$$\Pr[\text{Bad}] \leqslant q_{\mathsf{dec}}(\frac{\aleph}{\aleph - q_{\mathsf{dec}} + 1} \cdot \delta_{vu} + \epsilon_{\hbar} + \epsilon_{\mathsf{OTS}}).$$

□

6 Efficiency

Instantiation from DDH. We instantiate our generic construction with $\mathsf{H}_{\mathsf{DDH}}$ of running example 2. The detailed scheme can be found in Appx D, Figure 6. The drawback of all IPFE constructions based on DDH is the necessary limitation on the size of the message and key spaces, to compute efficiently the discrete logarithm of $g^{\langle \boldsymbol{m}, \boldsymbol{k} \rangle}$ during decryption. Concretely, if we allow ourselves to spend one second on that operation, this DL value must be less than say 33 bits[8]. For instance, in dimension $\ell = 100$, $\boldsymbol{m}$ and $\boldsymbol{k}$ must then have less than 13 bits per coordinates, which may be not sufficient for many applications. In this case, one should consider instantiations from $\mathsf{HSM}_{\mathsf{CL}}$ or DCR, where this limitation disappears and decryption takes less than a second, as we shall see. Nevertheless, our instantiation improves over the DDH scheme of [BBL17] as indicated by Table 1 in the introduction. Thanks to our proof strategy, independent of the choice of the plaintext by the adversary, we improve a term $q_{\mathsf{dec}} q^{-\nu} |\Delta \mathcal{M}|$ by

[8] This is the order of magnitude for a computation with BG/GS for an elliptic curve with 128 bits of security.

$q_{\mathsf{dec}}(q - q_{\mathsf{dec}} + 1)^{-1}$ in the security reduction. Restricting the message space as mentioned above, $\nu = 7$, this means that we gain a factor 5 on keys sizes and encryption cost, and 4 on the size of ciphertexts. We will see that for an instantiation based on DCR, where the message space is less restricted, our gains will be of higher orders of magnitude.

Instantiations from DCR and $\mathsf{HSM}_{\mathsf{CL}}$. We consider an instantiation of our generic construction with $\mathsf{eH}_{\mathsf{CL}}$ of running example 1 (cf. Appx. D, Figure 5 for the detailed scheme). We also consider an instantiation based on DCR: the details on the constructions of the PHFs based on DCR are not given in the main body of the paper as they are similar to that from $\mathsf{HSM}_{\mathsf{CL}}$, and again we refer the interested reader to Appx. D for details. In both constructions, we can take large message spaces compared to the DDH instantiation. As a result, our proof strategy now leads to huge gains on the instantiation of [BBL17] based on DCR. Again, in Theorem 35, they have a term of the form $q_{\mathsf{dec}}2^{-\nu}|\Delta\mathcal{M}|$ in the security reduction and set $\nu > \lambda + \log_2(2q_{\mathsf{dec}}|\Delta\mathcal{M}|)$ (in practice $\nu > 5000$). As shown in Table 1, in the introduction, this deeply impacts the performance of Setup, Enc and key sizes: the number of components of the mpk and msk are linear in ℓ in our work, but quadratic in [BBL17]. Besides, each component of their msk is sampled with a uniform distribution bounded by $\approx B^{\ell+1}N^2$ where B is s.t. $||\boldsymbol{m}||_\infty \leq B$ and N is the RSA modulus used for DCR. As a result, for large B, the size of each component increases rapidly with ℓ compared to our work, where each component is sampled from a Gausssian discrete distribution $\mathcal{D}_{\mathbf{Z},\sigma}$ where $\sigma \approx BN^2\sqrt{\ell}$ (cf. Remark 13).

C implementation and comparison. To fairly compare our DCR and $\mathsf{HSM}_{\mathsf{CL}}$ instantiations with the DCR instantiation of [BBL17], we have implemented in C the 3 schemes. Our program uses the Pari C library [PAR20], for arithmetic in class groups and $\mathbf{Z}/N\mathbf{Z}$, and the DGS library [AW18] for Discrete Gaussian sampling. We summarize our running tests in Tab. 2. For $\lambda = 112$ (resp. 128) security level, we use a 2048 (resp. 3072) bits RSA modulus for DCR and a 1348 (resp. 1827) bits fundamental discriminant for $\mathsf{HSM}_{\mathsf{CL}}$ as suggested in [CCL$^+$19]. For all schemes, we use the same message bound, $B = \sqrt{2^{\lambda-2}/\ell}$ (maximum for $\mathsf{HSM}_{\mathsf{CL}}$) for fair comparison. In practice, this is large enough to perform computations with double or quadruple precision. We use a dimension of $\ell = 100$. We omit the cost due to the OTS scheme and CRHF (they appear in all schemes with equal cost). Similarly, timings for Setup do not measure the time to generate the global parameters (N and quadratic discriminant, in practice the latter has cheaper generating cost). Our implementation does not use parallelism: timings were performed on a single core of an Intel(R) Core(TM) i7-7700 @ 3.60GHz.

Our DCR and $\mathsf{HSM}_{\mathsf{CL}}$ instantiations yield very efficient schemes and scale well, as they are linear in ℓ. As usual with class groups, the $\mathsf{HSM}_{\mathsf{CL}}$ instantiation has smaller ciphertexts and keys than the DCR one, while having larger timings but of the same order of magnitude. Note that at the 192 security level, the $\mathsf{HSM}_{\mathsf{CL}}$ instantiation becomes faster for Setup and Enc. To reduce further the size of class groups elements, we use the elegant compression technique recently

	$\lambda = 112, \ell = 100$			$\lambda = 128, \ell = 100$		
	This work		[BBL17]	This work		[BBL17]
	$\mathsf{HSM}_{\mathsf{CL}}$	DCR	DCR	$\mathsf{HSM}_{\mathsf{CL}}$	DCR	DCR
ν	–	–	5 533	–	–	6 349
group elt. (bits)	**1 179**	4 096	4 096	**1 563**	6 144	6 144
mpk	**44.2kB**	153kB	566MB	**56.6kB**	230kB	975MB
msk	**31.6kB**	155kB	1.3GB	**41.5kB**	233kB	2GB
$(\mathsf{sk}_{\boldsymbol{k}}, \overline{\mathsf{sk}}_{\boldsymbol{k}})$	**0.3kB**	1.5kB	13.4MB	**0.4kB**	2.3kB	19.9MB
ciphertext	**30kB**	103kB	283MB	**39kB**	154kB	488MB
Setup	9.7s	**4.8s**	≈ 11h	17.3s	**14.2s**	≈ 34h
Enc	5.2s	**1.6s**	1h20min	9.7s	**4.9s**	4h40min
Dec	0.5s	**0.08s**	5min43s	0.8s	**0.19s**	16min28s

Table 2: Our IPFE from $\mathsf{HSM}_{\mathsf{CL}}$ & DCR vs. the DCR scheme of [BBL17]

introduced in [DGS20]. As discussed above, our schemes dramatically improve the DCR instantiation of [BBL17], in all aspects, by several orders of magnitude.

7 Conclusion and open problems

We provide the first ind-fe-cca-security proof for IPFE built from PHF for which reduction tightness does not degrade with the size of the message space, and show that the resulting schemes are practical, even for large inner product values.

The problem of building ind-fe-cca-secure FE computing inner products *modulo a prime* remains open. In previous ind-fe-cpa work [ALS16, CLT18], such constructions require a stateful key generation to prevent adversaries from learning a combination of the master key components which is singular mod p but invertible over $\mathbf{Z}$ (thus revealing msk). For ind-fe-cca security, KeyGen is also executed for decryption queries. This results in various complications: should key derivation queries and decryption queries maintain independent states? which decryption queries result in a leakage of information (and should thereby be rejected)? Such an extension is non-trivial, and deserves an independent study.

The next step to strengthen security is simulation based security. Agrawal *et al.* showed in [ALMT20] that (variants of) the schemes in [ALS16] are adaptively secure in the simulation based security model. One can extract from their proof methodology new properties for PHFs which are sufficient for our constructions to attain simulation based security against *passive* adversaries. We note however that their proof technique does not go through for an instantiation from $\mathsf{HSM}_{\mathsf{CL}}$, since it requires computing a non zero multiple of the unknown order s of G^p.

Acknowledgements. This work was supported by the French ANR ALAMBIC project (ANR-16-CE39-0006).

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.

ABG19. M. Abdalla, F. Benhamouda, and R. Gay. From single-input to multi-client inner-product functional encryption. In *ASIACRYPT 2019, Part III*, *LNCS* 11923, pages 552–582. Springer, Heidelberg, December 2019.

ABP$^+$17. S. Agrawal, S. Bhattacherjee, D. H. Phan, D. Stehlé, and S. Yamada. Efficient public trace and revoke from standard assumptions: Extended abstract. In *ACM CCS 2017*, pages 2277–2293. ACM Press, October / November 2017.

ALMT20. S. Agrawal, B. Libert, M. Maitra, and R. Titiu. Adaptive simulation security for inner product functional encryption. In *PKC 2020, Part I*, *LNCS* 12110, pages 34–64. Springer, Heidelberg, May 2020.

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.

AW18. M. R. Albrecht and M. Walter. dgs, Discrete Gaussians over the Integers. Available at https://bitbucket.org/malb/dgs, 2018.

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.

BH01. J. Buchmann and S. Hamdy. A survey on IQ cryptography. In *Public Key Cryptography and Computational Number Theory*, pages 1–15. De Gruyter Proceedings in Mathematics, 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.

CCL$^+$19. G. Castagnos, D. Catalano, F. Laguillaumie, F. Savasta, and I. Tucker. Two-party ECDSA from hash proof systems and efficient instantiations. In *CRYPTO 2019, Part III*, *LNCS* 11694, pages 191–221. Springer, Heidelberg, August 2019.

CDG$^+$18. J. Chotard, E. Dufour Sans, R. Gay, D. H. Phan, and D. Pointcheval. Decentralized multi-client functional encryption for inner product. In *ASIACRYPT 2018, Part II*, *LNCS* 11273, pages 703–732. Springer, Heidelberg, December 2018.

CL15. G. Castagnos and F. Laguillaumie. Linearly homomorphic encryption from DDH. In *CT-RSA 2015*, *LNCS* 9048, pages 487–505. Springer, Heidelberg, April 2015.

CLT18. G. Castagnos, F. Laguillaumie, and I. Tucker. Practical fully secure unrestricted inner product functional encryption modulo p. In *ASIACRYPT 2018, Part II*, *LNCS* 11273, pages 733–764. Springer, Heidelberg, December 2018.

CS02. R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *EUROCRYPT 2002*, *LNCS* 2332, pages 45–64. Springer, Heidelberg, April / May 2002.

DGS20. S. Dobson, S. D. Galbraith, and B. Smith. Trustless groups of unknown order with hyperelliptic curves. Cryptology ePrint Archive, Report 2020/196, 2020. https://eprint.iacr.org/2020/196.

DOT18. P. Datta, T. Okamoto, and J. Tomida. Full-hiding (unbounded) multi-input inner product functional encryption from the k-Linear assumption. In

PKC 2018, Part II, *LNCS* 10770, pages 245–277. Springer, Heidelberg, March 2018.

DPP20. X. T. Do, D. H. Phan, and D. Pointcheval. Traceable inner product functional encryption. In *CT-RSA 2020*, *LNCS* 12006, pages 564–585. Springer, Heidelberg, February 2020.

GPV08. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *40th ACM STOC*, pages 197–206. ACM Press, May 2008.

HO09. B. Hemenway and R. Ostrovsky. Lossy trapdoor functions from smooth homomorphic hash proof systems. In *Electronic Colloquium on Computational Complexity, Report*, pages 09–127, 2009.

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.

Lag80. J. Lagarias. Worst-case complexity bounds for algorithms in the theory of integral quadratic forms. *Journal of Algorithms*, 1(2):142 – 186, 1980.

MR07. D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.*, 37(1):267–302, 2007.

NP15. M. Nandi and T. Pandit. Generic conversions from cpa to cca secure functional encryption. Cryptology ePrint Archive, Report 2015/457, 2015. https://eprint.iacr.org/2015/457.

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

PAR20. PARI Group, Univ. Bordeaux. *PARI/GP version* `2.11.4`, 2020. available from http://pari.math.u-bordeaux.fr/.

SW05. A. Sahai and B. R. Waters. Fuzzy identity-based encryption. In *EUROCRYPT 2005*, *LNCS* 3494, pages 457–473. Springer, Heidelberg, May 2005.

ZMY17. S. Zhang, Y. Mu, and G. Yang. Achieving ind-cca security for functional encryption for inner products. In *Information Security and Cryptology*, pages 119–139, Cham, 2017. Springer International Publishing.

A Backgound on lattices.

We here recall some definitions and basic results about Gaussian distributions. We use these in our security proofs to evaluate the distribution of an inner product when one of the two vectors follows a Gaussian distribution. We also recall an important result from [GPV08]. This explains the conditions for a Gaussian distribution over a lattice which is reduced modulo a sublattice to be close to a uniform distribution, another crucial point of our proofs.

Definition 17 (Gaussian Function). *For any $\sigma > 0$ define the Gaussian function on $\mathbf{R}^\ell$ centred at $\boldsymbol{c}$ with parameter σ:*

$$\forall \boldsymbol{x} \in \mathbf{R}^\ell, \rho_{\sigma,\boldsymbol{c}}(\boldsymbol{x}) = \exp(-\pi||\boldsymbol{x} - \boldsymbol{c}||^2/\sigma^2).$$

If $\sigma = 1$ (resp. $\boldsymbol{c} = \mathbf{0}$), then the subscript σ (resp. $\boldsymbol{c}$) is omitted.

Definition 18 (Discrete Gaussians). *For any $\boldsymbol{c} \in \mathbf{R}^\ell$, real $\sigma > 0$, and ℓ-dimensional lattice Λ, define the discrete Gaussian distribution over Λ as:*

$$\forall \boldsymbol{x} \in \Lambda, \quad \mathcal{D}_{\Lambda,\sigma,\boldsymbol{c}}(\boldsymbol{x}) = \rho_{\sigma,\boldsymbol{c}}(\boldsymbol{x})/\rho_{\sigma,\boldsymbol{c}}(\Lambda),$$

where $\rho_{\sigma,\boldsymbol{c}}(\Lambda) = \sum_{\boldsymbol{x}\in\Lambda} \rho_{\sigma,\boldsymbol{c}}(\boldsymbol{x})$.

The following lemma allows to evaluate the distribution of the inner product resulting from a constant vector $\boldsymbol{x}$, and a vector with coordinates sampled from a Gaussian distribution over the lattice $\boldsymbol{x} \cdot \mathbf{Z}$. Proof of Lemma 10 can be found in [CLT18, Aux. Material I].

Lemma 10. *Let $\boldsymbol{x} \in \mathbf{R}^\ell$ with $\boldsymbol{x} \neq \mathbf{0}$, $\boldsymbol{c} \in \mathbf{R}^\ell$, $\sigma \in \mathbf{R}$ with $\sigma > 0$. Let V be a random variable distributed according to $\mathcal{D}_{\boldsymbol{x}\cdot\mathbf{Z},\sigma,\boldsymbol{c}}$. Then the random variable S defined as $S = \langle \boldsymbol{x}, V\rangle$ is distributed according to $\mathcal{D}_{||\boldsymbol{x}||_2^2\cdot\mathbf{Z},\sigma\cdot||\boldsymbol{x}||_2,\langle \boldsymbol{c},\boldsymbol{x}\rangle}$.*

Lemma 11 ([GPV08]). *Let $\Lambda_0' \subset \Lambda_0 \subset \mathbf{R}^\ell$ be two lattices with the same dimension. Let $\epsilon \in (0, 1/2)$. Then for any $c \in \mathbf{R}^\ell$ and any $\sigma \geqslant \eta_\epsilon(\Lambda_0')$, the distribution $D_{\Lambda_0,\sigma,c} \mod \Lambda_0'$ is within statistical distance 2ϵ from the uniform distribution over Λ_0/Λ_0'. The value $\eta_\epsilon(\Lambda_0')$ is the* smoothing parameter *of the lattice Λ_0', as defined in [MR07].*

B Generic construction for building extended projective hash functions

Let $\mathcal{SM} := (\widehat{\mathcal{X}}, \mathcal{X}, \widehat{\mathcal{L}}, \mathcal{W}, \mathsf{R})$ be an SMP; let $\mathsf{H} := (\mathsf{hashkg}, \widehat{\mathsf{projkg}}, \mathsf{projkg}, \mathsf{hash}, \widehat{\mathsf{projhash}}, \mathsf{projhash})$ be the associated PHF; let $\tilde{p}$ be the smallest prime dividing $|\widehat{\mathcal{X}}/\widehat{\mathcal{L}}|$; and let $\Gamma : \widehat{\mathcal{X}} \times E \mapsto \{0, \ldots, \tilde{p} - 1\}$ be sampled from a family of δ_Γ-crhf. Cramer and Shoup, in [CS02, Sec. 7.2] provide a generic construction to build an extended projective hash function eH from H and Γ as follows:

- $\mathsf{ehashkg}(\mathcal{SM})$: Run $\mathsf{hk}_0 \hookleftarrow \mathsf{hashkg}(\mathcal{SM})$; $\mathsf{hk}_1 \hookleftarrow \mathsf{hashkg}(\mathcal{SM})$. Output $\mathsf{ehk} := (\mathsf{hk}_0, \mathsf{hk}_1)$, such that $K_{\mathsf{ehk}} = K_{\mathsf{hk}}^2$.

- $\mathsf{ehash}(\mathsf{ehk}, x, e)$: Parse $(\mathsf{hk}_0, \mathsf{hk}_1) = \mathsf{ehk}$; compute $\gamma := \Gamma(x, e)$. Output $\mathsf{hash}(\mathsf{hk}_0,\, x) \cdot \mathsf{hash}(\mathsf{hk}_1, x)^\gamma$, such that $\Sigma = \Pi$.
- $\widehat{\mathsf{eprojkg}}(\mathsf{ehk})$: Parse $(\mathsf{hk}_0, \mathsf{hk}_1) = \mathsf{ehk}$, compute $\widehat{\mathsf{hp}}_0 := \widehat{\mathsf{projkg}}(\mathsf{hk}_0)$; $\widehat{\mathsf{hp}}_1 := \widehat{\mathsf{projkg}}(\mathsf{hk}_1)$. Output $\widehat{\mathsf{ehp}} := (\widehat{\mathsf{hp}}_0, \widehat{\mathsf{hp}}_1)$, such that $K_{\widehat{\mathsf{ehp}}} = K^2_{\widehat{\mathsf{hp}}}$.
- $\mathsf{eprojkg}(\mathsf{ehk})$: Parse $(\mathsf{hk}_0, \mathsf{hk}_1) = \mathsf{ehk}$, compute $\mathsf{hp}_0 := \mathsf{projkg}(\mathsf{hk}_0)$; $\mathsf{hp}_1 := \mathsf{projkg}(\mathsf{hk}_1)$. Output $\mathsf{ehp} := (\mathsf{hp}_0, \mathsf{hp}_1)$, such that $K_{\mathsf{ehp}} = K^2_{\mathsf{hp}}$.
- $\widehat{\mathsf{eprojhash}}(\widehat{\mathsf{ehp}}, \widehat{x}, e)$: Parse $(\widehat{\mathsf{hp}}_0, \widehat{\mathsf{hp}}_1) = \widehat{\mathsf{ehp}}$. Compute $\gamma := \Gamma(\widehat{x}, e)$. Output $\widehat{\mathsf{projhash}}(\widehat{\mathsf{hp}}_0, \widehat{x}) \cdot \widehat{\mathsf{projhash}}(\widehat{\mathsf{hp}}_1, \widehat{x})^\gamma$.
- $\mathsf{eprojhash}(\mathsf{ehp}, x, w, e)$: Parse $(\mathsf{hp}_0, \mathsf{hp}_1) = \mathsf{ehp}$. Compute $\gamma := \Gamma(x, e)$. Output $\mathsf{projhash}(\mathsf{hp}_0, x, w) \cdot \mathsf{projhash}(\mathsf{hp}_1, x, w)^\gamma$.

C Proof of Lemma 1

Lemma 1. *Let $\mathcal{R}$ be either the ring $\mathbf{Z}$ or $\mathbf{Z}/q\mathbf{Z}$ for some prime q; ℓ and a be positive integers; and consider an $(\mathcal{R}, a, f, \aleph, \ell, \mathcal{M}, \mathcal{K})$-$\mathsf{ipfe}$-compatible projective hash function H, where $\aleph$ is either prime or hard to factor. From any $\boldsymbol{m} \in \{\boldsymbol{x}_0 - \boldsymbol{x}_1 \mid \boldsymbol{x}_0 \neq \boldsymbol{x}_1 \in \mathcal{M}\}$ one can efficiently and deterministically construct a matrix $\mathbf{B}_{\boldsymbol{m}} \in \mathcal{R}^{\ell\times\ell}$ associated to $\boldsymbol{m}$.*

Proof. If $\mathcal{R} = \mathbf{Z}/q\mathbf{Z}$ then by Def. 10, $q = \aleph$ is prime. Given $\boldsymbol{m}$ one deterministically generates a $\mathbf{Z}/\aleph\mathbf{Z}$-basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{\ell-1}) \in (\mathbf{Z}/\aleph\mathbf{Z})^{(\ell-1)\times\ell}$ of $\boldsymbol{m}^\perp$. Let $\boldsymbol{b}_\ell \in (\mathbf{Z}/\aleph\mathbf{Z})^\ell$ be a vector outside the subspace $\boldsymbol{m}^\perp$, also chosen in a deterministic manner. The resulting matrix $\mathbf{B} \in (\mathbf{Z}/\aleph\mathbf{Z})^{\ell\times\ell}$ whose rows are the vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_\ell$ is invertible modulo $\aleph$.

If $\mathcal{R} = \mathbf{Z}$ then first observe that by Def. 10, one has $\log_f(f^{\langle \boldsymbol{x}, \boldsymbol{y}\rangle}) = \langle \boldsymbol{x}, \boldsymbol{y}\rangle$ for any $\boldsymbol{x} \in \mathcal{M}$, $\boldsymbol{y} \in \mathcal{K}$. Since f is of order $\aleph$, this implies that vectors in $\mathcal{M}$ (resp $\mathcal{K}$) are of bounded norm, i.e. $\mathcal{M}$ and $\mathcal{K}$ are subsets of $\{\boldsymbol{x} \in \mathbf{Z}^\ell \ : \ ||\boldsymbol{x}||_\infty < \sqrt{\frac{\aleph}{2\ell}}\}$.

Now w.l.o.g., assume the n_0 first coordinates of $\boldsymbol{m} \in \mathbf{Z}^\ell$ are zero (for some n_0), and all remaining entries are non-zero. The rows $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{\ell-1} \in \mathbf{Z}^\ell$ of the following matrix form a basis of $\boldsymbol{m}^\perp$:

$$\mathbf{B}_{\mathsf{top}} = \left[\begin{array}{c|ccccc} \mathbf{I}_{n_0} & & & & & \\ \hline & -m_{n_0+2} & m_{n_0+1} & & & \\ & & -m_{n_0+3} & m_{n_0+2} & & \\ & & & \ddots & \ddots & \\ & & & & -m_\ell & m_{\ell-1} \end{array}\right] \in \mathcal{R}^{(\ell-1)\times\ell}.$$

Letting $\boldsymbol{b}_\ell := \boldsymbol{m}$, the matrix $\mathbf{B} \in \mathcal{R}^{\ell\times\ell}$ whose rows are the vectors $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_\ell)$ is invertible mod $\aleph$. If $\aleph$ is prime, from the norm bounds this is always true [CLT18]. If $\aleph$ is composite, either $\mathbf{B}$ is invertible in $\mathbf{Z}/\aleph\mathbf{Z}$ otherwise its determinant reveals a non trivial factor of $\aleph$ [ALS16, CLT18]. □

D Instantiations of our ind-fe-cca-secure IPFE

Example 1 – $\mathsf{HSM}_{\mathsf{CL}}$**.** Instantiating the IPFE of Fig. 3 with H_{CL} yields the IPFE scheme depicted in Fig. 5.

Setting the parameters. We use the output $(\tilde{s}, f, g_p, \widehat{G}, F)$ of the $\mathsf{GenGroup}$ generator of Def. 1 and require that p is a μ bit prime, with $\mu \geqslant \lambda$. The message and key spaces are $\mathcal{M} = \mathcal{K} = \{\boldsymbol{x} \in \mathbf{Z}^\ell : ||\boldsymbol{x}||_\infty < \sqrt{\frac{p}{2\ell}}\}$. The decryption algorithm uses a centred modulus to recover $\langle \boldsymbol{k}, \boldsymbol{m} \rangle$ over $\mathbf{Z}$.

To guarantee the scheme's security we sample the coordinates of the secret key from $\mathcal{D}_{\mathbf{Z},\sigma}$, i.e. discrete Gaussian entries of standard deviation $\sigma > \tilde{s}p^{3/2}\sqrt{\lambda}$, which yields $\delta_{vs} = 2^{-\lambda}$ and $\delta_{vu} = 1/p + \delta_{\Gamma} + 2^{-\lambda}$ (*cf.* Lemmas 3 and 5). To sample encryption randomness (i.e. witnesses for H_{CL}), it suffices to use $\mathcal{D}_{\mathbf{Z},\sigma'}$ for $\sigma' > \tilde{s}\sqrt{\lambda}$, since $\{g_p^r, r \hookleftarrow \mathcal{D}_{\mathbf{Z},\sigma'}\}$ is at distance less than $2^{-\lambda}$ from the uniform distribution in G^p (*cf.* Remark 1).

We also use two families of CRHF $\mathcal{H}_{\widehat{G}}$ and $\mathcal{H}$, such that $\hbar \hookleftarrow \mathcal{H}_{\widehat{G}}(1^\lambda)$ maps $\{0,1\}^*$ to $\widehat{G}$; and $\Gamma \hookleftarrow \mathcal{H}(1^\lambda)$ maps $\widehat{G}^2$ to $\{0, \ldots, p-1\}$. Finally we use a strongly unforgeable OTS scheme $\mathsf{OTS} := (\mathsf{OTS.Setup}, \mathsf{OTS.Sign}, \mathsf{OTS.Verif})$.

Corollary 1 (of Thm. 2). *If the* $\mathsf{HSM}_{\mathsf{CL}}$ *problem is hard, the IPFE scheme of Fig. 5 is* ind-fe-cca*-secure.*

```
$\mathsf{Setup}(1^\lambda, 1^\mu, 1^\ell)$:
Sample a $\mu$ bit prime $p$
$\mathsf{params}_{\mathsf{CL}} \leftarrow \mathsf{GenGroup}(1^\lambda, p)$
$\hbar \hookleftarrow \mathcal{H}_{\widehat{G}}(1^\lambda)$; $\Gamma \hookleftarrow \mathcal{H}(1^\lambda)$
For $1 \leqslant i \leqslant \ell$ :
    Sample $\mathsf{hk}_i, \mathsf{ehk}_{0,i}, \mathsf{ehk}_{1,i} \hookleftarrow \mathcal{D}_{\mathbf{Z},\sigma}$
    Let $\mathsf{hp}_i \leftarrow g_p^{\mathsf{hk}_i}$
    Let $(\mathsf{ehp}_{0,i}, \mathsf{ehp}_{1,i}) \leftarrow (g_p^{\mathsf{ehk}_{0,i}}, g_p^{\mathsf{ehk}_{1,i}})$
Return $\mathsf{msk} := (\mathbf{hk}, \mathbf{ehk}_0, \mathbf{ehk}_1)$ and
$\mathsf{mpk} := (\mathsf{params}_{\mathsf{CL}}, p, \mathbf{hp}, \mathbf{ehp}_0, \mathbf{ehp}_1, \hbar, \Gamma, \mathsf{params})$
```

```
$\mathsf{Enc}(\mathsf{mpk}, \boldsymbol{m})$
If $\boldsymbol{m} \notin \mathcal{M}$, return $\perp$
$(\mathsf{vk}, \mathsf{sk}_{\mathsf{OTS}}) \leftarrow \mathsf{OTS.Setup}(1^\lambda)$;
$e \leftarrow \hbar(\mathsf{vk})$
Sample $r \hookleftarrow \mathcal{D}_{\mathbf{Z},\sigma'}$; set $c_0 \leftarrow g_p^r$
$\gamma \leftarrow \Gamma(c_0, e)$
For $1 \leqslant i \leqslant \ell$ :
    $c_i \leftarrow f^{m_i}\mathsf{hp}_i^r$
    $\overline{c}_i \leftarrow (\mathsf{ehp}_{0,i}\mathsf{ehp}_{1,i}^{\gamma})^r$
Let $\boldsymbol{ct} := (c_0, \boldsymbol{c}, \overline{\boldsymbol{c}})$;
$\sigma \leftarrow \mathsf{OTS.Sign}(\mathsf{sk}_{\mathsf{OTS}}, \boldsymbol{ct})$
Return $(\boldsymbol{ct}, \mathsf{vk}, \sigma)$
```

```
$\mathsf{KeyDer}(\mathsf{msk}, \boldsymbol{k})$
If $\boldsymbol{k} \notin \mathcal{K}$, return $\perp$
$sk_{\boldsymbol{k}} \leftarrow \langle \boldsymbol{k}, \mathbf{hk} \rangle$
$\overline{\mathsf{sk}}_0 \leftarrow \langle \boldsymbol{k}, \mathbf{ehk}_0 \rangle$;
$\overline{\mathsf{sk}}_1 \leftarrow \langle \boldsymbol{k}, \mathbf{ehk}_1 \rangle$
Return $(sk_{\boldsymbol{k}}, \overline{\mathsf{sk}}_0, \overline{\mathsf{sk}}_1, \boldsymbol{k})$
```

```
$\mathsf{Dec}(\mathsf{mpk}, (sk_{\boldsymbol{k}}, \overline{\mathsf{sk}}_0, \overline{\mathsf{sk}}_1, \boldsymbol{k}), (\boldsymbol{ct}, \mathsf{vk}, \sigma))$
If $\boldsymbol{ct} \notin \widehat{G}^{2\ell+1}$, return $\perp$
If $\mathsf{OTS.Verif}(\mathsf{vk}, \boldsymbol{ct}, \sigma) = 0$, return $\perp$
$e \leftarrow \hbar(\mathsf{vk})$; $\gamma \leftarrow \Gamma(c_0, e)$
If $c_0^{\overline{\mathsf{sk}}_0 + \gamma\overline{\mathsf{sk}}_1} \neq \prod_{i\in[\ell]} \overline{c}_i^{k_i}$, return $\perp$
$M \leftarrow (\prod_{i\in[\ell]} c_i^{k_i}) \cdot (c_0^{-sk_{\boldsymbol{k}}})$
If $M \notin F$, return $\perp$
$\mathsf{sol} \leftarrow \mathsf{Solve}(M)$
If $\mathsf{sol} \geqslant p/2$, return $(\mathsf{sol} - p)$
Else return $\mathsf{sol}$
```

Fig. 5: ind-fe-cca-secure IPFE from the $\mathsf{HSM}_{\mathsf{CL}}$ assumption.

Example 2 – DDH. Instantiating the IPFE of Fig. 3 with $\mathsf{H}_{\mathsf{DDH}}$ yields the IPFE scheme depicted in Fig. 6.

Setting the parameters. We use the output (G, g, q) of the $\mathsf{Gen}_{\mathsf{DDH}}$ generator of Def. 2, and consider generators $g_0, g_1 \hookleftarrow G$. We also use two CRHF generators $\mathscr{H}_G$ and $\mathscr{H}$, such that $\hbar \hookleftarrow \mathscr{H}_G(1^\lambda)$ maps $\{0,1\}^*$ to G; and $\Gamma \hookleftarrow \mathscr{H}(1^\lambda)$ maps G^3 to $\{0,\dots,q-1\}$. Finally we use a strongly unforgeable one time signature scheme $\mathsf{OTS} := (\mathsf{OTS.Setup}, \mathsf{OTS.Sign}, \mathsf{OTS.Verif})$.

The message and key spaces are subsets of $(\mathbf{Z}/q\mathbf{Z})^\ell$. The decryption algorithm recovers $\langle \boldsymbol{k}, \boldsymbol{m}\rangle$ over $\mathbf{Z}/q\mathbf{Z}$ if it is sufficiently small for the discrete logarithm of $g^{\langle \boldsymbol{k}, \boldsymbol{m}\rangle}$ to be efficient. Hashing key coordinates are sampled from $\mathcal{U}(\mathbf{Z}/q\mathbf{Z})$, as is the encryption randomness.

Corollary 2 (of Thm. 2). *If the* DDH *problem is hard, the IPFE scheme of Fig. 6 is* ind-fe-cca*-secure.*

```
$\mathsf{Setup}(1^\lambda, 1^\ell)$:
  $(G, g, q) \leftarrow \mathsf{Gen}_{\mathsf{DDH}}(1^\lambda)$
  $g_0, g_1 \hookleftarrow G$
  $\hbar \hookleftarrow \mathscr{H}_G(1^\lambda)$; $\Gamma \hookleftarrow \mathscr{H}(1^\lambda)$
  For $1 \leqslant i \leqslant \ell$ :
      $(\kappa_{0,i}, \kappa_{1,i}, \kappa_{2,i}, \kappa_{3,i}, \kappa_{4,i}, \kappa_{5,i}) \hookleftarrow (\mathbf{Z}/q\mathbf{Z})^6$
      $\mathsf{hp}_i \leftarrow g_0^{\kappa_{0,i}} g_1^{\kappa_{1,i}}$
      $\mathsf{ehp}_{0,i} \leftarrow g_0^{\kappa_{2,i}} g_1^{\kappa_{3,i}}$, $\mathsf{ehp}_{1,i} \leftarrow g_0^{\kappa_{4,i}} g_1^{\kappa_{5,i}}$
  Return $\mathsf{msk} := (\boldsymbol{\kappa}_0, \boldsymbol{\kappa}_1, \boldsymbol{\kappa}_2, \boldsymbol{\kappa}_3, \boldsymbol{\kappa}_4, \boldsymbol{\kappa}_5)$
      $\mathsf{mpk} := (G, q, g, g_0, g_1,$
              $\mathbf{hp}, \mathbf{ehp}_0, \mathbf{ehp}_1, \hbar, \Gamma, \mathsf{params})$

$\mathsf{KeyDer}(\mathsf{msk}, \boldsymbol{k})$
  If $\boldsymbol{k} \notin \mathcal{K}$, return $\perp$
  $(\mathsf{sk}_0, \mathsf{sk}_1) \leftarrow (\langle \boldsymbol{k}, \boldsymbol{\kappa}_0\rangle, \langle \boldsymbol{k}, \boldsymbol{\kappa}_1\rangle) \in \mathbf{Z}/q\mathbf{Z}$
  For $2 \leqslant \mu \leqslant 5$, let $\overline{\mathsf{sk}}_\mu \leftarrow \langle \boldsymbol{k}, \boldsymbol{\kappa}_\mu\rangle \in \mathbf{Z}/q\mathbf{Z}$
  $sk_{\boldsymbol{k}} := (\mathsf{sk}_0, \mathsf{sk}_1)$;
  $\overline{\mathsf{sk}}_{\boldsymbol{k}} := (\overline{\mathsf{sk}}_2, \overline{\mathsf{sk}}_3, \overline{\mathsf{sk}}_4, \overline{\mathsf{sk}}_5)$
  Return $(sk_{\boldsymbol{k}}, \overline{\mathsf{sk}}_{\boldsymbol{k}}, \boldsymbol{k})$

$\mathsf{Enc}(\mathsf{mpk}, \boldsymbol{m})$
  If $\boldsymbol{m} \notin \mathcal{M}$, return $\perp$
  $(\mathsf{vk}, \mathsf{sk}_{\mathsf{OTS}}) \leftarrow \mathsf{OTS.Setup}(1^\lambda)$
  $e \leftarrow \hbar(\mathsf{vk})$
  $r \hookleftarrow \mathbf{Z}/q\mathbf{Z}$
  Let $(x_0, x_1) \leftarrow (g_0^r, g_1^r)$
  $\gamma \leftarrow \Gamma(x_0, x_1, e)$
  For $1 \leqslant i \leqslant \ell$ :
      $c_i \leftarrow g^{m_i} \mathsf{hp}_i^r$
      $\overline{c}_i \leftarrow (\mathsf{ehp}_{0,i} \mathsf{ehp}_{1,i}^\gamma)^r$
  Let $\boldsymbol{ct} := (c_0, \boldsymbol{c}, \overline{\boldsymbol{c}})$
  $\sigma \leftarrow \mathsf{OTS.Sign}(\mathsf{sk}_{\mathsf{OTS}}, \boldsymbol{ct})$
  Return $(\boldsymbol{ct}, \mathsf{vk}, \sigma)$

$\mathsf{Dec}(\mathsf{mpk}, (sk_{\boldsymbol{k}}, \overline{\mathsf{sk}}_{\boldsymbol{k}}, \boldsymbol{k}), (\boldsymbol{ct}, \mathsf{vk}, \sigma))$
  If $\boldsymbol{ct} \notin G^{2\ell+2}$,
      return $\perp$
  If $\mathsf{OTS.Verif}(\mathsf{vk}, \boldsymbol{ct}, \sigma) = 0$,
      return $\perp$
  $e \leftarrow \hbar(\mathsf{vk})$; $\gamma \leftarrow \Gamma(x_0, x_1, e)$
  If $x_0^{\overline{\mathsf{sk}}_2 + \gamma\overline{\mathsf{sk}}_4} x_1^{\overline{\mathsf{sk}}_3 + \gamma\overline{\mathsf{sk}}_5} \neq \prod_{i\in[\ell]} \overline{c}_i^{k_i}$
      Then return $\perp$
  $M \leftarrow (\prod_{i\in[\ell]} c_i^{k_i}) \cdot (x_0^{-\mathsf{sk}_0} x_1^{-\mathsf{sk}_1})$
  Return $\log_g(M)$
```

Fig. 6: ind-fe-cca-secure IPFE scheme from the DDH assumption.

Example 3 – DCR. Let $N = pq$ be a product of two safe primes $p = 2p' + 1$ and $q = 2q' + 1$, where p' and q' are sufficiently large primes $p', q' > 2^{l(\lambda)}$. Here λ is the security parameter and l is some polynomial (such that factoring is 2^λ-hard). a function of κ. Let $N' = p'q'$; the order of the group $(\mathbf{Z}/N\mathbf{Z})^*$ of invertible elements in $\mathbf{Z}/N\mathbf{Z}$ is $4N'$. One can write $(\mathbf{Z}/N^2\mathbf{Z})^* \simeq G_N \times G_{N'} \times G_2 \times T$ where

$\simeq$ denotes group isomorphism, $\times$ is the Cartesian product, G_i are cyclic groups of order i, and T is the order-2 cyclic group generated by $-1 \bmod N^2$.

We denote g a random generator of $G_{N'}$; g can be thought of as a random $2N$-th residue. It further holds that $(1+N)$ is a generator for G_N.

One can define a subset membership problem from the DCR assumption as $\mathcal{SM}_{\mathsf{DCR}} := (\widehat{\mathcal{X}} \simeq G_N \times G_{N'} \times T, \mathcal{X} \simeq G_N \times G_{N'}, \widehat{\mathcal{L}} \simeq G_{N'} \times T, \mathcal{W} = \mathbf{Z}, \mathsf{R}_{\mathsf{DCR}})$, where $\mathsf{R}_{\mathsf{DCR}} := \{(x, w) \in (G_{N'} \times \mathbf{Z}) \mid x = g^w\}$.

The associated PHF, denoted $\mathsf{H}_{\mathsf{DCR}}$ very much resembles that from $\mathsf{HSM}_{\mathsf{CL}}$, only hashing keys are sampled uniformly from $\mathcal{D}_{\mathbf{Z}^\ell,\sigma}$, with discrete Gaussian entries of standard deviation $\sigma > \sqrt{\lambda} N^{5/2}$.

Compatibility. $\mathsf{H}_{\mathsf{DCR}}$ is $(Z, 1, 1{+}N, N, \ell, \mathcal{M}, \mathcal{K})$-$\mathsf{ipfe}$-compatible, where $\mathcal{M} = \mathcal{K} \subseteq \{\boldsymbol{x} \in \mathbf{Z}^\ell : ||x||_\infty < \sqrt{\frac{N}{2\ell}}\}$. We denote M_m the upper bound on the infinite norm of message vectors (for simplicity we assume the same bound for vectors in $\mathcal{K}$). For correctness, $M_m < \sqrt{\frac{N}{2\ell}}$, however one may chose a smaller bound according to one's needs in order to improve the schemes efficiency (*cf.* Remark 13). The projective hash family $\mathsf{H}_{\mathsf{DCR}}$ is also $(1+N, G_N)$-decomposable.

Security. Lemma 12, whose proof follows from that of [ALS16, Thm. 5] states sufficient conditions for $\mathsf{H}_{\mathsf{DCR}}$ to be vector smooth.

Lemma 12. *If* hashkg *samples* $\mathbf{hk} \hookleftarrow \mathcal{D}_{\mathbf{Z}^\ell,\sigma}$ *with discrete Gaussian entries of standard deviation* $\sigma > \sqrt{\lambda} N^{5/2}$ *then* $\mathsf{H}_{\mathsf{DCR}}$ *is* δ_{vs}*-vector smooth over* $G_N \times G_{N'}$ *on* G_N*, with* $\delta_{vs} = 2^{-\lambda}$.

Lemma 13, whose proof follows from that of Lemma 5, [ALS16, Thm. 5], and [CS02, Thm. 3] states sufficient conditions for $\mathsf{H}_{\mathsf{DCR}}$ to be vector universal.

Lemma 13. *Let* $spf(N)$ *denote the smallest prime factor of* N*. If algorithm* $\mathsf{ehashkg}$ *of* $\mathsf{eH}_{\mathsf{DCR}}$ *samples the vectors of hashing keys from* $\mathcal{D}_{\mathbf{Z}^\ell,\sigma}$ *with discrete Gaussian entries of standard deviation* $\sigma > \sqrt{\lambda} N^{5/2}$*, and* $\Gamma : \widehat{G}^2 \mapsto \{0, \ldots, N-1\}$ *is sampled from a family of* δ_Γ*-collision resistant hash functions, then* $\mathsf{eH}_{\mathsf{DCR}}$ *is* δ_{vu}*-vector universal, where* $\delta_{vu} = 1/spf(N) + \delta_\Gamma + 2^{-\lambda}$.

We also use two families of CRHF $\mathcal{H}_{\mathsf{DCR}}$ and $\mathcal{H}$, and a strongly unforgeable OTS scheme $\mathsf{OTS} := (\mathsf{OTS.Setup}, \mathsf{OTS.Sign}, \mathsf{OTS.Verif})$.

Remark 13. As was the case for H_{CL}, the choice of σ depends on the message space. In particular, denoting B an upper bound for the infinite norm of message vectors in $\mathcal{M}$, it holds that for any $\boldsymbol{m} \in \Delta\mathcal{M}$, $||\boldsymbol{m}||_2 \leq 2B\sqrt{\ell}$. One should then set $\sigma > 2BN^2\sqrt{\ell\lambda}$.

```
$\mathsf{Setup}(1^\lambda, 1^\ell, M_m)$:
$\quad$ Sample $l(\lambda)$-bit safe primes $p$ and $q$;
$\quad$ set $N \leftarrow pq$;
$\quad$ Sample $u \hookleftarrow \mathcal{U}((\mathbf{Z}/N^2\mathbf{Z}))$;
$\quad$ Let $g \leftarrow u^{2N}$;
$\quad$ Let $\sigma \leftarrow \sqrt{\lambda} N^{5/2}$;
$\quad$ $\hbar \hookleftarrow \mathscr{H}_{\mathsf{DCR}}(1^\lambda)$; $\mathit{\Gamma} \hookleftarrow \mathscr{H}(1^\lambda)$
$\quad$ For $1 \leqslant i \leqslant \ell$ :
$\quad\quad$ Sample $\mathsf{hk}_i, \mathsf{ehk}_{0,i}, \mathsf{ehk}_{1,i} \hookleftarrow \mathscr{D}_{\mathbf{Z},\sigma}$
$\quad\quad$ Let $\mathsf{hp}_i \leftarrow g^{\mathsf{hk}_i}$
$\quad\quad$ Let $(\mathsf{ehp}_{0,i}, \mathsf{ehp}_{1,i}) \leftarrow (g^{\mathsf{ehk}_{0,i}}, g^{\mathsf{ehk}_{1,i}})$
$\quad$ Return $\mathsf{msk} := (\mathbf{hk}, \mathbf{ehk}_0, \mathbf{ehk}_1)$ and
$\quad$ $\mathsf{mpk} := (N, g, \mathbf{hp}, \mathbf{ehp}_0, \mathbf{ehp}_1, \hbar, \mathit{\Gamma}, M_m)$
```

```
$\mathsf{Enc}(\mathsf{mpk}, \boldsymbol{m})$
$\quad$ If $\boldsymbol{m} \notin \mathcal{M}$, return $\perp$
$\quad$ $(\mathsf{vk}, \mathsf{sk}_{\mathsf{OTS}}) \leftarrow \mathsf{OTS.Setup}(1^\lambda)$;
$\quad$ $e \leftarrow \hbar(\mathsf{vk})$
$\quad$ Sample $r \hookleftarrow \{0, \ldots, \lfloor N/4 \rfloor\}$;
$\quad$ Set $c_0 \leftarrow g^r$
$\quad$ $\gamma \leftarrow \mathit{\Gamma}(c_0, e)$
$\quad$ For $1 \leqslant i \leqslant \ell$ :
$\quad\quad$ $c_i \leftarrow (1+N)^{m_i} \mathsf{hp}_i^r$
$\quad\quad$ $\overline{c}_i \leftarrow (\mathsf{ehp}_{0,i} \mathsf{ehp}_{1,i}^{\gamma})^r$
$\quad$ Let $\boldsymbol{ct} := (c_0, \boldsymbol{c}, \overline{\boldsymbol{c}})$;
$\quad$ $\sigma \leftarrow \mathsf{OTS.Sign}(\mathsf{sk}_{\mathsf{OTS}}, \boldsymbol{ct})$
$\quad$ Return $(\boldsymbol{ct}, \mathsf{vk}, \sigma)$
```

```
$\mathsf{KeyDer}(\mathsf{msk}, \boldsymbol{k})$
$\quad$ If $\boldsymbol{k} \notin \mathcal{K}$, return $\perp$
$\quad$ $sk_{\boldsymbol{k}} \leftarrow \langle \boldsymbol{k}, \mathbf{hk} \rangle$
$\quad$ $\overline{\mathsf{sk}}_0 \leftarrow \langle \boldsymbol{k}, \mathbf{ehk}_0 \rangle$;
$\quad$ $\overline{\mathsf{sk}}_1 \leftarrow \langle \boldsymbol{k}, \mathbf{ehk}_1 \rangle$
$\quad$ Return $(sk_{\boldsymbol{k}}, \overline{\mathsf{sk}}_0, \overline{\mathsf{sk}}_1, \boldsymbol{k})$
```

```
$\mathsf{Dec}(\mathsf{mpk}, (sk_{\boldsymbol{k}}, \overline{\mathsf{sk}}_0, \overline{\mathsf{sk}}_1, \boldsymbol{k}), (\boldsymbol{ct}, \mathsf{vk}, \sigma))$
$\quad$ If $\boldsymbol{ct} \notin \widehat{\mathcal{X}}^{2\ell+1}$,
$\quad\quad$ return $\perp$
$\quad$ If $\mathsf{OTS.Verif}(\mathsf{vk}, \boldsymbol{ct}, \sigma) = 0$,
$\quad\quad$ return $\perp$
$\quad$ $e \leftarrow \hbar(\mathsf{vk})$; $\gamma \leftarrow \mathit{\Gamma}(c_0, e)$
$\quad$ If $c_0^{\overline{\mathsf{sk}}_0 + \gamma \overline{\mathsf{sk}}_1} \neq \prod_{i \in [\ell]} \overline{c}_i^{k_i}$, return $\perp$
$\quad$ $M \leftarrow (\prod_{i \in [\ell]} c_i^{k_i}) \cdot (c_0^{-sk_{\boldsymbol{k}}})$
$\quad$ If $M \notin G_N$, return $\perp$
$\quad$ $\mathsf{sol} \leftarrow \frac{M-1}{N} \bmod N$
$\quad$ If $\mathsf{sol} \geqslant N/2$, return $(\mathsf{sol} - N)$
$\quad$ Else return $\mathsf{sol}$
```

Fig. 7: ind-fe-cca-secure IPFE from the DCR assumption.

E Comparing our PHF properties to those of [BBL17]

E.1 Chosen plaintext security

We here demonstrate that any PHF which can be used to instantiate the generic construction of [BBL17] for IPFE secure against chosen plaintext attacks is key homomorphic and vector-smooth.

We refer the reader to [BBL17] for definitions of *strong diversity*, *translation indistinguishability*, and *universal translation indistinguishability*.

Lemma 14. *Let $\mathcal{R}$ be a ring, either $\mathbf{Z}$ or $\mathbf{Z}/q\mathbf{Z}$ for some prime q; $\ell > 0$ an integer; $\mathcal{SM} := (\mathcal{X}, \mathcal{L}, \mathcal{W}, \mathsf{R})$ an SMP and H the associated PHF. Consider a function $\mathsf{hk}_\perp : \mathcal{X} \backslash \mathcal{L} \mapsto \mathcal{K}_{\mathsf{hk}}$, an element $f \in \mathit{\Pi}$, and positive integers $\aleph$ and M. In [BBL17], to build* ind-fe-cpa *IPFE schemes, H must be key homomorphic, $(\mathsf{hk}_\perp, M, \epsilon_{ti})$-translation-indistinguishable, $(\mathsf{hk}_\perp, f, \aleph)$-strongly diverse, and one sets $M := (\aleph/\ell)^{1/2}$. If so, H is $(\ell \cdot \epsilon_{ti})$-vector-smooth over $\mathcal{X}$ on $F := \langle f \rangle$.*

Lemma 14. Let H be a PHF as described in the lemma. For $i \in [\ell]$, let hk_i be independent random variables following the distribution sampled by $\mathsf{hashkg}(\mathcal{SM})$, let $\mathbf{hk} := (\mathsf{hk}_1, \ldots, \mathsf{hk}_\ell)$ and $\mathbf{hp} \leftarrow \mathsf{projkg}(\mathbf{hk})$. Consider a random variable m^* taking values in $\Delta\mathcal{M}$. For some $\boldsymbol{m} \in \Delta\mathcal{M}$, we condition our analysis on $m^* = \boldsymbol{m}$ (we assume, w.l.o.g. that the probability this even occurs is non zero). We denote $\mathbf{B}_{\boldsymbol{m}}$ the matrix associated to $\boldsymbol{m}$ built as per Lemma 1, and denote its' rows $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_\ell)$. Let $X \hookleftarrow \mathcal{X}\backslash\mathcal{L}$, and $Y \hookleftarrow \mathcal{U}(F)$. Then H is $\delta_{vs}(\ell)$-vector-smooth over $\mathcal{X}$ on F if $\mathcal{U}$ and $\mathcal{V}$ are $\delta_{vs}(\ell)$-close, where:

$$\mathcal{U} := \{X, \{\mathsf{projkg}(\mathsf{hk}_i)\}_{i\in[\ell]}, \{\langle \mathbf{hk}, \boldsymbol{b}_i^M\rangle\}_{i\in[\ell-1]}, \mathsf{hash}(\langle \mathbf{hk}, \boldsymbol{b}_\ell^M\rangle, X)\cdot Y\} \text{ and}$$
$$\mathcal{V} := \left\{X, \{\mathsf{projkg}(\mathsf{hk}_i)\}_{i\in[\ell]}, \{\langle \mathbf{hk}, \boldsymbol{b}_i^M\rangle\}_{i\in[\ell-1]}, \mathsf{hash}(\langle \mathbf{hk}, \boldsymbol{b}_\ell^M\rangle, X)\right\}.$$

We first use the $(\mathsf{hk}_\perp, M, \epsilon_{ti})$-translation-indistinguishability of H, and replace each hk_i by $\mathsf{hk}_i + a_i \cdot \mathsf{hk}_\perp(X)$ for $a_i \hookleftarrow \{-M, \ldots, M\}$ satisfying $\boldsymbol{a} = \mu\boldsymbol{m}$ for some $\mu \in \mathbf{R}$. By repeated sampling, it holds that $\mathcal{V}'$ is $\ell\epsilon_{ti}$-close to $\mathcal{V}$, where:

$$\begin{aligned}\mathcal{V}' := \{X, &\{\mathsf{projkg}(\mathsf{hk}_i + a_i \cdot \mathsf{hk}_\perp(X))\}_{i\in[\ell]},\\ &\{\langle \mathbf{hk} + \boldsymbol{a}\cdot \mathsf{hk}_\perp(X), \boldsymbol{b}_i^M\rangle\}_{i\in[\ell-1]}, \mathsf{hash}(\langle \mathbf{hk} + \boldsymbol{a}\cdot\mathsf{hk}_\perp(X), \boldsymbol{b}_\ell^M\rangle, X)\}.\end{aligned}$$

By construction, $\boldsymbol{b}_i \in \boldsymbol{m}^\perp$ for $i \in [\ell - 1]$, furthermore, by the homomorphic properties of H it holds that:

$$\begin{aligned}\mathcal{V}' := \{X, \{\mathsf{projkg}(\mathsf{hk}_i + a_i \cdot \mathsf{hk}_\perp(X))\}_{i\in[\ell]}, \{\langle \mathbf{hk}, \boldsymbol{b}_i\rangle\}_{i\in[\ell-1]},&\\ \mathsf{hash}(\langle \mathbf{hk}, \boldsymbol{b}_\ell\rangle, X)\cdot \mathsf{hash}(\mathsf{hk}_\perp(X), X)^{\langle \boldsymbol{a}, \boldsymbol{b}_\ell\rangle}\}.&\end{aligned}$$

From the $(\mathsf{hk}_\perp, f, \aleph)$-strong diversity of H, we can now write:

$$\mathcal{V}' = \{X, \{\mathsf{projkg}(\mathsf{hk}_i)\}_{i\in[\ell]}, \{\langle \mathbf{hk}, \boldsymbol{b}_i\rangle\}_{i\in[\ell-1]}, \mathsf{hash}(\langle \mathbf{hk}, \boldsymbol{b}_\ell\rangle, X)\cdot f^{\langle \boldsymbol{a}, \boldsymbol{b}_\ell\rangle}\}.$$

As f is of order $\aleph$, $M := (\aleph/\ell)^{1/2}$ s.t. $\boldsymbol{a}$ is sampled uniformly in $\{-(\aleph/\ell)^{1/2}, \ldots, (\aleph/\ell)^{1/2}\}$ subject on the condition $\boldsymbol{a} \in \langle \boldsymbol{m}\rangle$, and $\langle \boldsymbol{m}, \boldsymbol{b}_\ell\rangle \neq 0$, the distribution induced by $f^{\langle \boldsymbol{a}, \boldsymbol{b}_\ell\rangle}$ is the uniform distribution in the subgroup $F = \langle f\rangle$. Thus:

$$\mathcal{V}' = \mathcal{U} = \{X, \mathbf{hp}, \{\langle \mathbf{hk}, \boldsymbol{b}_i\rangle\}_{i\in[\ell-1]}, \mathsf{hash}(\langle \mathbf{hk}, \boldsymbol{b}_\ell\rangle, X)\cdot Y | Y \hookleftarrow \mathcal{U}(F)\}.$$

□

E.2 Chosen ciphertext security

We here demonstrate that any PHF which can be used to instantiate the generic construction of [BBL17] for ind-fe-cca-secure IPFE is key homomorphic and vector-universal. We refer the reader to [BBL17] for the definitions of *2-universality* and *universal translation indistinguishability.*

Lemma 15. *Let $\mathcal{R}$ be a ring, either $\mathbf{Z}$ or $\mathbf{Z}/q\mathbf{Z}$ for some prime q; ℓ, λ positive integers; $\mathcal{SM} := (\mathcal{X}, \mathcal{L}, \mathcal{W}, \mathsf{R})$ an SMP and* eH *the associated EPHF. Further consider a function* $\mathsf{ehashkg}'$*, which on input 1^λ outputs a hashing key* ehk *in some set $K'_{\mathsf{ehk}} \subseteq K_{\mathsf{ehk}}$; an element $f \in \Pi$; $\epsilon_{uti}, \epsilon_{2u} > 0$; positive integers $\aleph$, M; and a subset Σ of $\mathbf{Z}$. In [BBL17], to build an* ind-fe-cca *IPFE scheme one sets $M := (\aleph/\ell)^{1/2}$; $\Sigma := \{1, \ldots, \aleph - 1\}$; and* eH *must be key-homomorphic, projection-key-homomorphic,* $(\mathsf{ehashkg}', M, \epsilon_{uti})$*-universally-translation-indistinguishable and it must hold that for any $t \in \Sigma$, the PHF* $(t \cdot \mathsf{ehashkg}', \mathsf{eprojkg}, \mathsf{ehash}, \mathsf{eprojhash})$ *is ϵ_{2u}-universal$_2$, where the algorithm $t \cdot \mathsf{ehashkg}'$ runs* $\mathsf{ehashkg}'$

and multiplies the output by t. If all these properties hold then H *is* $(2\ell\cdot\epsilon_{uti}+\epsilon_{2u})$*-vector-universal.*

Proof. Consider an EPHF $\mathsf{eH} := (\mathsf{ehashkg}, \mathsf{eprojkg}, \mathsf{ehash}, \mathsf{eprojhash})$ as in the lemma statement. For $i \in [\ell]$, let ehk_i be independent random variables following the distribution sampled by $\mathsf{ehashkg}(\mathcal{SM})$, and denote $\boldsymbol{\mathsf{ehk}} := (\mathsf{ehk}_1, \ldots, \mathsf{ehk}_\ell)$. Consider a random variable M taking values in $\Delta\mathcal{M}$, and the associated matrix $\mathbf{B}_M$. The hash function eH is $\delta_{vu}(\ell)$-vector-universal if for any $\boldsymbol{\mathsf{ehp}} \in (K_{\widehat{\mathsf{ehp}}})^\ell$; any $\boldsymbol{m} \in \Delta\mathcal{M}$; any $\boldsymbol{k} \in \mathcal{K}$ s.t. $\boldsymbol{k} \notin \boldsymbol{m}^\perp$; any $(x^*, e^*) \in \widehat{\mathcal{X}} \times E$, $(x, e) \in \widehat{\mathcal{X}} \backslash \mathcal{L} \times E$, s.t. $(x, e) \neq (x^*, e^*)$, and for any $(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{\ell-1}) \in (K_{\mathsf{ehk}})^{\ell-1}$; $(\pi_1^*, \ldots, \pi_\ell^*) \in \mathit{\Pi}^\ell$ and $\pi \in \mathit{\Pi}$ it holds that:

$$\Pr\Big[\,\mathsf{ehash}(\langle \boldsymbol{\mathsf{ehk}}, \boldsymbol{k}\rangle, x, e) = \pi \;\Big|\; (\mathsf{ehash}(\mathsf{ehk}_i, x^*, e^*) = \pi_i^* \text{ for } i \in [\ell])$$
$$\wedge \mathsf{eprojkg}(\boldsymbol{\mathsf{ehk}}) = \boldsymbol{\mathsf{ehp}} \wedge (\lessdot \boldsymbol{\mathsf{ehk}}, \boldsymbol{b}_j^M \gtrdot_{2a} = \boldsymbol{v}_j \text{ for } j \in [\ell-1]) \wedge M = \boldsymbol{m}\Big] \leqslant \delta_{vu}(\ell).$$

Let E_1 denote the event "$\mathsf{ehash}(\mathsf{ehk}_i, x^*, e^*) = \pi_i^*$ for $i \in [\ell]$", E_2 denote the event "$\mathsf{eprojkg}(\boldsymbol{\mathsf{ehk}}) = \boldsymbol{\mathsf{ehp}}$", E_3 denote the event "$\lessdot \boldsymbol{\mathsf{ehk}}, \boldsymbol{b}_j^M \gtrdot_{2a} = \boldsymbol{v}_j$ for $j \in [\ell-1]$", and E_4 denote the event "$M = \boldsymbol{m}$". We hereafter condition on E_4 and denote $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_\ell) \in \mathcal{R}^{\ell\times\ell}$ the rows of $\mathbf{B}_{\boldsymbol{m}}$. Let X be a random variable following the same distribution as $\mathsf{ehash}(\langle \boldsymbol{\mathsf{ehk}}, \boldsymbol{k}\rangle, x, e)$ and denote E_0 the event "$X = \pi$".

We first use the $(\mathsf{ehashkg}', M, \epsilon_{uti})$-universal-translation-indistinguishability of eH: sample $\mathsf{ehk}' \hookleftarrow \mathsf{ehashkg}'(\mathcal{SM})$; $\mathsf{ehk}_i'' \hookleftarrow \mathsf{ehashkg}(\mathcal{SM})$; and $\alpha_i \hookleftarrow \{-M, \ldots, M\}$ for $i \in [\ell]$, such that $\boldsymbol{\alpha} \in \langle \boldsymbol{m} \rangle$. For $\mathcal{K}_{\mathsf{ehk}'} \subseteq \mathcal{R}^{2a}$, and $\boldsymbol{\alpha} \in \mathcal{R}^\ell$ we denote $\mathsf{ehk}' \cdot \boldsymbol{\alpha} := (\mathsf{ehk}_0' \cdot \boldsymbol{\alpha}, \ldots, \mathsf{ehk}_{2a-1}' \cdot \boldsymbol{\alpha}) \in (\mathcal{R}^\ell)^{2a}$.

1. Consider the random variable
$$X' := \mathsf{ehash}(\lessdot \boldsymbol{\mathsf{ehk}}'' + \mathsf{ehk}' \cdot \boldsymbol{\alpha}, \boldsymbol{k} \gtrdot_{2a}, x, e).$$
By the key homomorphism of eH, and denoting $t := \langle \boldsymbol{\alpha}, \boldsymbol{k} \rangle \in \mathcal{R}$, one gets:
$$X' = \mathsf{ehash}(\lessdot \boldsymbol{\mathsf{ehk}}'', \boldsymbol{k} \gtrdot_{2a}, x, e) \cdot \mathsf{ehash}(t \cdot \mathsf{ehk}', x, e).$$
Let E_0' denote the event "$X' = \pi$".
2. Let E_1' denote the event "$\mathsf{ehash}(\mathsf{ehk}_i'' + \alpha_i \mathsf{ehk}', x^*, e^*) = \pi_i^*$ for $i \in [\ell]$". Or equivalently $\mathsf{ehash}(\mathsf{ehk}_i'', x^*, e^*) \cdot \mathsf{ehash}(\mathsf{ehk}', x^*, e^*)^{\alpha_i} = \pi_i^*$.
3. Let E_2' denote the event: "$\boldsymbol{\mathsf{ehp}} = \mathsf{eprojkg}(\boldsymbol{\mathsf{ehk}}'' + \mathsf{ehk}' \cdot \boldsymbol{\alpha})$," which by key homomorphism of eH is exactly the event:
$$\text{“}\boldsymbol{\mathsf{ehp}} = \mathsf{eprojkg}(\boldsymbol{\mathsf{ehk}}'') \cdot \mathsf{eprojkg}(\mathsf{ehk}')^{\boldsymbol{\alpha}}\text{”}.$$
4. Let E_3' denote the event:"$\lessdot \boldsymbol{\mathsf{ehk}}'' + \boldsymbol{\alpha} \cdot \mathsf{ehk}', \boldsymbol{b}_j \gtrdot_{2a} = \boldsymbol{v}_j$ for $j \in [\ell-1]$", which, since $\boldsymbol{\alpha} \in \langle \boldsymbol{m} \rangle$ and $\boldsymbol{b}_i \in \boldsymbol{m}^\perp$ for $i \in [\ell-1]$, is exactly the event:
$$\text{“}\lessdot \boldsymbol{\mathsf{ehk}}'', \boldsymbol{b}_j \gtrdot_{2a} = \boldsymbol{v}_j \text{ for } j \in [\ell-1]''.$$

From the $(\mathsf{ehashkg}', M, \epsilon_{uti})$-universal-translation-indistinguishability of eH:

$$|\Pr[E_0 \wedge E_1 \wedge E_2 \wedge E_3 \wedge E_4] - \Pr[E_0' \wedge E_1' \wedge E_2' \wedge E_3' \wedge E_4]| \leqslant \ell \cdot \epsilon_{uti}.$$

Let us now consider the probability $\Pr[E_0' \wedge E_1' \wedge E_2' \wedge E_3' \wedge E_4] \leqslant \Pr[E_0' | E_1' \wedge E_2' \wedge E_3' \wedge E_4]$. We denote $\mathfrak{p} = \Pr[E_0' | E_1' \wedge E_2' \wedge E_3' \wedge E_4]$. We first observe that event E_3' is independent of ehk'. So the only fixed information on ehk' comes from event E_2' which, at most, fixes the value of $\mathsf{ehp}' := \mathsf{eprojkg}(\mathsf{ehk}')$; and E_1'

which fixes the value of $\mu^* := \mathsf{ehash}(\mathsf{ehk}', x^*, e^*)$. Now from the norm bounds on $\boldsymbol{k} \in \mathcal{K}$ and $\boldsymbol{\alpha}$, it holds that $t = \langle \boldsymbol{\alpha}, \boldsymbol{k} \rangle \in \{1, \ldots, \aleph - 1\}$, and so the PHF $(t \cdot \mathsf{ehashkg}', \mathsf{eprojkg}, \mathsf{ehash}, \mathsf{eprojhash})$ is ϵ_{2u}-universal$_2$. Thus, for any $\pi' \in \mathit{\Pi}$, it holds that:

$$\begin{aligned}\Pr[\pi' =& \mathsf{ehash}(t\mathsf{ehk}', x, e) \wedge \mu^* = \mathsf{ehash}(\mathsf{ehk}', x^*, e^*) \wedge \mathsf{ehp}' = \mathsf{eprojkg}(\mathsf{ehk}')] \\ &\leq \epsilon_{2u} \cdot \Pr[\mu^* = \mathsf{ehash}(\mathsf{ehk}', x^*, e^*) \wedge \mathsf{ehp}' = \mathsf{eprojkg}(\mathsf{ehk}')].\end{aligned}$$

Since $\mathbf{ehk}''$ and ehk' are sampled independently, we have: $\Pr[E_0' \wedge E_1' \wedge E_2' \wedge E_3' \wedge E_4] \leqslant \epsilon_{2u} \Pr[E_1' \wedge E_2' \wedge E_3' \wedge E_4]$; we can thus conclude: $\Pr[E_0 | E_1 \wedge E_2 \wedge E_3 \wedge E_4] \leqslant \epsilon_{2u} + 2\ell \cdot \epsilon_{uti}$, and so eH is $(\epsilon_{2u} + 2\ell \cdot \epsilon_{uti})$-vector universal. □

E.3 Comparing tightness of security reductions

We here compare the quality of security reductions obtained in [BBL17] to ours. Let ℓ and a be positive integers; $\mathcal{R}$ be a ring, either $\mathbf{Z}$ or $\mathbf{Z}/q\mathbf{Z}$ for some prime q; $\mathcal{SM} := (\mathcal{X}, \mathcal{L}, \mathcal{W}, \mathsf{R})$ an SMP; H the associated $(\mathcal{R}, a, f, \aleph, \ell, \mathcal{M}, \mathcal{K})$-$\mathsf{ipfe}$-compatible PHF; and eH the resulting extended PHF eH, obtained via the generic construction of [CS02] (detailed in Appendix B).

As explained Lemmas 14 and 15, in [BBL17], to build $\mathsf{ind\text{-}fe\text{-}cpa}$-secure IPFE, H must be translation indistinguishable (parametrised by ϵ_{ti}), moreover to build $\mathsf{ind\text{-}fe\text{-}cca}$-secure IPFE, eH must be universal translation indistinguishable (parametrised by ϵ_{uti}) and a slight variant of H must be universal$_2$ (parametrised by ϵ_{2u}). These properties imply δ_{vs}-vector smoothness for H and δ_{vu}-vector universality for eH where $\delta_{vs} = \ell \cdot \epsilon_{ti}$, and $\delta_{vu} = 2\ell \cdot \epsilon_{uti} + \epsilon_{2u}$.

Chosen plaintext attacks. In [BBL17] the adversarial advantage is bound by:

$$\mathsf{Adv}^{BBL17,\mathsf{fe\text{-}cpa}}_{\mathsf{FE},\mathcal{A}} \leq \delta_{\mathcal{L}} + \ell \cdot |\mathit{\Delta}\mathcal{M}| \cdot \epsilon_{ti} \text{ where } |\mathit{\Delta}\mathcal{M}| \leqslant (4 \cdot (\frac{\aleph}{2\ell})^{1/2})^{\ell}.$$

From our proof this advantage is upper bounded by:

$$\mathsf{Adv}^{\mathsf{fe\text{-}cpa}}_{\mathsf{FE},\mathcal{A}} \leq \delta_{\mathcal{L}} + \ell \cdot \epsilon_{ti}.$$

We thus gain a factor $|\mathit{\Delta}\mathcal{M}|$. Note that for PHFs where hash keys are sampled uniformly from K_{hk} this term disappears since such PHF's are 0-vector smooth. In this case the quality of our security reduction and that of [BBL17] coincide.

Chosen ciphertext attacks. In [BBL17] the adversarial advantage is bounded above by:

$$\mathsf{Adv}^{BBL17,\mathsf{fe\text{-}cca}}_{\mathsf{FE},\mathcal{A}} \leqslant \delta_{\mathcal{L}} + \ell|\mathit{\Delta}\mathcal{M}|(\epsilon_{ti} + 2\epsilon_{uti}) + 2q_{\mathsf{dec}}|\mathit{\Delta}\mathcal{M}|(\epsilon_{2u} + \epsilon_{\hbar} + \epsilon_{\mathsf{OTS}}).$$

From our security proof this advantage is upper bounded by:

$$\mathsf{Adv}^{\mathrm{us,fe\text{-}cca}}_{\mathsf{FE},\mathcal{A}} \leqslant \delta_{\mathcal{L}} + \ell(\frac{q_{\mathsf{dec}}\aleph}{\aleph - q_{\mathsf{dec}} + 1} 2\epsilon_{uti} + \epsilon_{ti}) + q_{\mathsf{dec}}(\frac{\aleph}{\aleph - q_{\mathsf{dec}} + 1}\epsilon_{2u} + \epsilon_{\hbar} + \epsilon_{\mathsf{OTS}}).$$

For a message space of order $\aleph$ of 128 bits, and allowing the adversary to make $q_{\mathsf{dec}} = 2^{20}$ decryption queries this yields:

$$\mathsf{Adv}^{\mathsf{fe\text{-}cca}}_{BBL17} \leqslant \delta_{\mathcal{L}} + 2^{66\ell}\ell^{1-\ell/2}(\epsilon_{ti} + 2 \cdot \epsilon_{uti}) + 2^{66\ell+21}\ell^{-\ell/2}(\epsilon_{2u} + \epsilon_{\hbar} + \epsilon_{\mathsf{OTS}}),$$

whereas in this work:

$$\mathsf{Adv}^{\mathrm{us},\mathsf{fe\text{-}cpa}}_{\mathsf{FE},\mathscr{A}} < \delta_{\mathscr{L}} + \ell \cdot (2^{21}\epsilon_{uti} + \epsilon_{ti}) + 2^{20}(\epsilon_{2u} + \epsilon_{\hbar} + \epsilon_{\mathsf{OTS}}).$$

Finally for vectors of length $\ell = 100$:

$$\mathsf{Adv}^{\mathsf{fe\text{-}cca}}_{BBL17} < \delta_{\mathscr{L}} + 2^{6224}(\epsilon_{ti} + 2\epsilon_{uti}) + 2^{6238}(\epsilon_{2u} + \epsilon_{\hbar} + \epsilon_{\mathsf{OTS}}),$$

whereas in this work:

$$\mathsf{Adv}_{\mathrm{us}} < \delta_{\mathscr{L}} + 2^{27}(\epsilon_{ti} + 2\epsilon_{uti}) + 2^{20}(\epsilon_{2u} + \epsilon_{\hbar} + \epsilon_{\mathsf{OTS}}).$$

We note that even if hashing keys are sampled uniformly, which sets $\epsilon_{ti} = \epsilon_{uti} = 0$, our security proof significantly reduces $\mathscr{A}$'s advantage (we do not have the $|\mathit{\Delta}\mathcal{M}|$ term), which allows us to use smaller keys, and significantly gain in efficiency (*cf.* Section 6).

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: