Fully Secure Functional Encryption for Inner Products, from Standard Assumptions

Shweta Agrawal[1], Benoît Libert[2(✉)], and Damien Stehlé[2]

[1] IIT Delhi, New Delhi, India
[2] Laboratoire LIP (U. Lyon, CNRS, ENSL, INRIA, UCBL),
ENS de Lyon, Lyon, France
benoit.libert@ens-lyon.fr

Abstract. Functional encryption is a modern public-key paradigm where a master secret key can be used to derive sub-keys SK_F associated with certain functions F in such a way that the decryption operation reveals $F(M)$, if M is the encrypted message, and nothing else. Recently, Abdalla *et al.* gave simple and efficient realizations of the primitive for the computation of linear functions on encrypted data: given an encryption of a vector $\boldsymbol{y}$ over some specified base ring, a secret key $SK_{\boldsymbol{x}}$ for the vector $\boldsymbol{x}$ allows computing $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$. Their technique surprisingly allows for instantiations under standard assumptions, like the hardness of the Decision Diffie-Hellman (DDH) and Learning-with-Errors (LWE) problems. Their constructions, however, are only proved secure against *selective* adversaries, which have to declare the challenge messages M_0 and M_1 at the outset of the game.

In this paper, we provide constructions that provably achieve security against more realistic *adaptive* attacks (where the messages M_0 and M_1 may be chosen in the challenge phase, based on the previously collected information) for the same inner product functionality. Our constructions are obtained from hash proof systems endowed with homomorphic properties over the key space. They are (almost) as efficient as those of Abdalla *et al.* and rely on the same hardness assumptions.

In addition, we obtain a solution based on Paillier's composite residuosity assumption, which was an open problem even in the case of selective adversaries. We also propose LWE-based schemes that allow evaluation of inner products modulo a prime p, as opposed to the schemes of Abdalla *et al.* that are restricted to evaluations of integer inner products of short integer vectors. We finally propose a solution based on Paillier's composite residuosity assumption that enables evaluation of inner products modulo an RSA integer $N = p \cdot q$.

We demonstrate that the functionality of inner products over a prime field is powerful and can be used to construct bounded collusion FE for all circuits.

Keywords: Functional encryption · Adaptive security · Standard assumptions · DDH · LWE · Extended LWE · Composite residuosity

1   Introduction

Functional encryption (FE) [19,56] is a generalization of public-key encryption, which overcomes the all-or-nothing, user-based access to data that is inherent to public key encryption and enables fine grained, role-based access that makes it very desirable for modern applications. A bit more formally, given an encryption $\mathsf{enc}(X)$ and a key corresponding to a function F, the key holder only learns $F(X)$ and nothing else. Apart from its theoretical appeal, the concept of FE also finds numerous applications. In cloud computing platforms, users can store encrypted data on a remote server and subsequently provide the server with a key SK_F which allows it to compute the function F of the underlying data without learning anything else.

In some cases, the message $X = (\mathsf{IND}, M)$ consists of an index IND (which can be thought of as a set of descriptive attributes) and a message M, which is sometimes called "payload". One distinguishes FE systems with public index, where IND is publicly revealed by the ciphertext but M is hidden, from those with private index, where IND and M are both hidden. Public index FE is popularly referred to as attribute based encryption.

A Brief History of FE. The birth of Functional Encryption can be traced back to Identity Based Encryption [17,57] which can be seen as the first non-trivial generalization of Public Key Encryption. However, it was the work of Sahai and Waters [56] that coined the term Attribute Based Encryption, and the subsequent, natural unification of all these primitives under the umbrella of Functional Encryption took place only relatively recently [19,49]. Constructions of public index FE have matured from specialized – equality testing [13,17,35], keyword search [1,16,44], Boolean formulae [42], inner product predicates [44], regular languages [58] – to general polynomial-size circuits [18,34,40] and even Turing machines [37]. The journey of private index FE has been significantly more difficult, with inner product predicate constructions [3,44] being the state of the art for a long time until the recent elegant generalization to polynomial-size circuits [41].

However, although private index FE comes closer than ever before to the goal of general FE, it falls frustratingly short. This is because all known constructions of private index FE only achieve *weak attribute hiding*, which severely restricts the function keys that the adversary can request in the security game – the adversary may request keys for functions f_i that do not decrypt the challenge ciphertext (IND^*, M^*), i.e., $f_i(\mathsf{IND}^*) \neq 0$ holds for all i. The most general notion of FE – private index, strongly attribute hiding – has been built for the restricted case of bounded collusions [38,39] or using the brilliant, but ill-understood[1] machinery of multi-linear maps [33] and indistinguishability obfuscation [33]. These constructions provide FE for general polynomial-size circuits and Turing machines [37], but, perhaps surprisingly, there has been little effort to build the general notion of FE ground-up, starting from smaller functionalities.

---

[1] Indeed, the two candidate multi-linear maps [24,32] put forth in 2013 were recently found to be insecure [23,43].

This appears as a gaping hole that begs to be filled. Often, from the practical standpoint, efficient constructions for a smaller range of functionalities, such as linear functions or polynomials, are extremely relevant, and such an endeavour will also help us understand the fundamental barriers that thwart our attempts for general FE. This motivates the question:

Can we build FE for restricted classes of functions, satisfying standard security definitions, under well-understood assumptions?

In 2015, Abdalla et al. [2] considered the question of building FE for linear functions. Here, a ciphertext C encrypts a vector $\boldsymbol{y} \in \mathcal{D}^\ell$ over some ring $\mathcal{D}$, a secret key for the vector $\boldsymbol{x} \in \mathcal{D}^\ell$ allows computing $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ and nothing else about $\boldsymbol{y}$. Note that this is quite different from the inner product predicate functionality of [3,44]: the former computes the actual value of the inner product while the latter tests whether the inner product is zero or not, and reveals a hidden bit M if so. Abdalla *et al.* [2] showed, surprisingly, that this functionality allows for very simple and efficient realizations under standard assumptions like the Decision Diffie-Hellman (DDH) and Learning-with-Errors (LWE) assumptions [53]. The instantiation from DDH was especially unexpected since DDH is not known to easily lend itself to the design of such primitives.[2] What enables this surprising result is that the functionality itself is rather limited – note that with ℓ queries, the adversary can reconstruct the entire message vector. Due to this, the scheme need not provide *collusion resistance*, which posits that no collection of secret keys for functions $F_1, \ldots, F_q$ should make it possible to decrypt a ciphertext that no individual such key can decrypt. Collusion resistance is usually the chief obstacle in proving security of FE schemes. On the contrary, for linear FE constructions, if two adversaries combine their keys, they do get a valid new key, but this key gives them a plaintext which could anyway be computed by their individual plaintexts. Hence, collusion is permitted by the functionality itself, and constructions can be much simpler. As we shall see below, linear FE is already very useful and yields many interesting applications, as we discuss in the full version of the paper [4].

More recently, Bishop et al. [12] considered the same functionality as Abdalla *et al.* in the secret-key setting with the motivation of achieving function privacy.

While [12] considers adaptive adversaries, their construction requires bilinear maps and does not operate over standard DDH-hard groups. In the public-key setting, Abdalla *et al.* [2] only proved their schemes to be secure against *selective* adversaries, that have to declare the challenge messages M_0, M_1 of the semantic security game upfront, before seeing the master public key mpk. Selective security is usually too weak a notion for practical applications and is often seen as a stepping stone to proving full adaptive security. Historically, most flavors of functional encryption have been first realized for selective adversaries [13,33,42,44,56] before being upgraded to attain full security. Boneh and Boyen [14] observed that a standard complexity leveraging argument can be used

---

[2] And indeed, this unsuitability partially manifests itself in the limitation of message/function space of the aforementioned construction: message/function vectors must be short integer vectors, and the inner product is evaluated over the integers.

to argue that a selectively-secure system is also adaptively secure. However, this argument is not satisfactory in general as the reduction incurs an exponential security loss in the message length. Quite recently, Ananth et al. [8] described a generic method of building adaptively secure functional encryption systems from selectively secure ones. However their transformation is based on the existence of a *sufficiently expressive* selectively secure FE scheme, where sufficiently secure roughly means capable of evaluating a weak PRF. Since no such scheme from standard assumptions is known, their transformation does not apply to our case, and in any case would significantly increase the complexity of the construction, even if it did.

Our Results. In this paper, we describe fully secure functional encryption systems for the evaluation of inner products on encrypted data. We propose schemes that evaluate inner products of integer vectors, based on DDH, LWE and the Composite Residuosity hardness assumptions. Our DDH-based and LWE-based constructions for integer inner products are of efficiency comparable to those of Abdalla *et al.* [2] and rely on the same standard assumptions. Note that a system based on Paillier's composite residuosity assumption was an open problem even for the case of selective adversaries, which we resolve in this work.

Additionally, we propose schemes that evaluate inner products modulo a prime p or a composite $N = pq$, based on the LWE and Composite Residuosity hardness assumptions. In contrast, the constructions of [2] must restrict the ring $\mathcal{D}$ to the ring of integers, which is a significant drawback. Indeed, although their DDH-based realization allows evaluating $\langle \boldsymbol{x}, \boldsymbol{y} \rangle \mod p$ when the latter value is sufficiently small, their security proof restricts the functionality to the computation of $\langle \boldsymbol{x}, \boldsymbol{y} \rangle \in \mathbb{Z}$.

The functionality of inner products over a prime field is powerful: we show that it can be bootstrapped all the way to yield a conceptually simple construction for bounded collusion FE for all circuits. The only known construction for general FE handling bounded collusions is by Gorbunov et al. [39]. Our construction is conceptually simpler, albeit a bit more inefficient. Also, since it requires the inner product functionality over a prime field, it can only be instantiated with our LWE-based scheme for now.

1.1   Overview of Techniques

We briefly summarize our techniques below.

Fully Secure Linear FE: Hash Proof Systems. Our DDH-based construction and its security proof implicitly build on hash proof systems [26]. It involves public parameters comprised of group elements $\left(g, h, \{ h_i = g^{s_i} \cdot h^{t_i} \}_{i=1}^{\ell} \right)$, where g, h generate a cyclic group $\mathbb{G}$ of prime order q, and the master secret key is $\mathsf{msk} = (\boldsymbol{s}, \boldsymbol{t}) \in \mathbb{Z}_q^{\ell} \times \mathbb{Z}_q^{\ell}$. On input of a vector $\boldsymbol{y} = (y_1, \ldots, y_\ell) \in \mathbb{Z}_q^{\ell}$, the encryption algorithm computes $(g^r, h^r, \{ g^{y_i} \cdot h_i^r \}_{i=1}^{\ell})$ in such a way that a secret key of the form $SK_{\boldsymbol{x}} = (\langle \boldsymbol{s}, \boldsymbol{x} \rangle, \langle \boldsymbol{t}, \boldsymbol{x} \rangle)$ allows computing $g^{\langle \boldsymbol{y}, \boldsymbol{x} \rangle}$ in the same way as in [2]. Despite its simplicity and its efficiency (only one more group element than

in [2] is needed in the ciphertext), we show that the above system can be proved fully secure using arguments – akin to those of Cramer and Shoup [25] – which consider what the adversary knows about the master secret key $(\boldsymbol{s}, \boldsymbol{t}) \in \mathbb{Z}_q^\ell \times \mathbb{Z}_q^\ell$ in the information theoretic sense. The security proof is arguably simpler than its counterpart in the selective case [2]. As in all security proofs based on hash proof systems, it uses the fact that the secret key is known to the reduction at any time, which makes it simpler to handle secret key queries without knowing the adversary's target messages $\boldsymbol{y}_0, \boldsymbol{y}_1 \in \mathbb{Z}_q^\ell$ in advance.

While our DDH-based realization only enables efficient decryption when the inner product $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ is contained in a sufficiently small interval, we show how to eliminate this restriction using Paillier's cryptosystem in the same way as in [21,22]. We thus obtain the first solution based on the Composite Residuosity assumption, which was previously an open problem (even in the case of selective security).

LWE-Based Fully Secure Linear FE. Our LWE-based construction builds on the dual Regev encryption scheme from Gentry et al. [35]. Its security analysis requires more work. The master public key contains a random matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$. For simplicity, we restrict ourselves to plaintext vectors and secret key vectors with binary coordinates. Each vector coordinate $i \in \{1, \ldots, \ell\}$ requires a master public key component $\boldsymbol{u}_i^T = \boldsymbol{z}_i^T \cdot \mathbf{A} \in \mathbb{Z}_q^n$, for a small norm vector $\boldsymbol{z}_i \in \mathbb{Z}^m$ made of Gaussian entries which will be part of the master secret key $\mathsf{msk} = \{\boldsymbol{z}_i\}_{i=1}^\ell$. Each $\{\mathbf{u}_i\}_{i=1}^\ell$ can be seen as a syndrome in the GPV trapdoor function for which vector $\mathbf{z}_i$ is a pre-image. Our security analysis will rely on the fact that each GPV syndrome has a large number of pre-images and, conditionally on $\mathbf{u}_i \in \mathbb{Z}_q^n$, each $\mathbf{z}_i$ retains a large amount of entropy. In the security proof, this will allow us to apply arguments similar to those of hash proof systems [26] when we will generate the challenge ciphertext using $\{\mathbf{z}_i\}_{i=1}^\ell$. More precisely, when the first part $\mathbf{c}_0 \in \mathbb{Z}_q^m$ of the ciphertext is a random vector instead of an actual LWE sample $\mathbf{c}_0 = \mathbf{A} \cdot \mathbf{s} + \mathbf{e}_0$, the action of $\{\mathbf{z}_i\}_{i=1}^\ell$ on $\mathbf{c}_0 \in \mathbb{Z}_q^m$ produces vectors that appear statistically uniform to any legitimate adversary. In order to properly simulate the challenge ciphertext using the master secret key $\{\mathbf{z}_i\}_{i=1}^\ell$, we use a variant of the extended LWE assumption [50] (eLWE) so as to have the (hint) values $\{\langle \mathbf{z}_i, \mathbf{e}_0 \rangle\}_{i=1}^\ell$ at disposal. One difficulty is that the reductions from LWE to eLWE proved in [7,20] handle a single hint vector $\boldsymbol{z}$. Fortunately, we extend the techniques of Brakerski et al. [20] using the gadget matrix from [45] to obtain a reduction from LWE to the multi-hint variant of eLWE that we use in the security proof. More specifically, we prove that the multi-hint variant of eLWE remains at least as hard as LWE when the adversary obtains as many as $n/2$ hints, where n is the dimension of the LWE secret.

Evaluation Inner Products Modulo p. Our construction from the DDH assumption natively supports the computation of inner products modulo a prime p as long as the remainder $\langle \boldsymbol{x}, \boldsymbol{y} \rangle \bmod p$ falls in a polynomial-size interval. Under the Paillier and LWE assumptions, we first show how to compute integer

inner products $\langle \boldsymbol{x}, \boldsymbol{y} \rangle \in \mathbb{Z}$. In a second step, we upgrade our Paillier and LWE-based systems so as to compute inner products modulo a composite $N = pq$ and a prime p, respectively, without leaking the actual value $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ over $\mathbb{Z}$.

Hiding anything but the remainder modulo N or p requires additional techniques. In the context of LWE-based FE, this is achieved by using an LWE modulus of the form $q = p \cdot p'$ and multiplying plaintexts by p', so that an inner product modulo q over the ciphertext space natively translates into an inner product modulo p for the underlying plaintexts.

The latter plaintext/ciphertext manipulations do not solve another difficulty which arises from the discrepancy between the base rings of the master key and the secret key vectors: indeed, the master key consists of integer vectors, whereas the secret keys are defined modulo an integer. When the adversary queries a secret key vector $\boldsymbol{x} \in \mathbb{Z}_p^\ell$ (or $\mathbb{Z}_N^\ell$), it gets the corresponding combination modulo p of the master key components. By making appropriate vector queries that are linearly dependent modulo p (and hence valid), an attacker could learn a combination of the master key components which is singular modulo p but invertible over the field of rational numbers: it would then obtain the whole master key! However, note that as long as the adversary only queries secret keys for $\ell - 1$ independent vectors over $\mathbb{Z}_p^\ell$ (or $\mathbb{Z}_N^\ell$), there is no reason not to reveal more than $\ell-1$ secret keys overall. In order to make sure that the adversary only obtains redundant information by making more than $\ell - 1$ queries, we assume that a trusted authority keeps track of all vectors $\boldsymbol{x}$ for which secret keys were previously given out (more formally, the key generation algorithm is stateful).

Compiling Linear FE to Bounded Collusion General FE. We provide a conceptually simpler way to build q-query Functional Encryption for all circuits. The only known construction for this functionality was suggested by Gorbunov *et al.* in [39]. At a high level, the q-query construction by Gorbunov *et al.* is built in several layers, as follows:

1. They start with a single key FE scheme for all circuits, which was provided by [55].
2. The single FE scheme is compiled into a q-query scheme for NC_1 circuits. This is the most non-trivial part of the construction. They run N copies of the single key scheme, where $N = O(q^4)$. To encrypt, they encrypt the views of some N-party MPC protocol computing some functionality related to C, *à la* "MPC in the head". For the MPC protocol, they use the BGW [10] semi-honest MPC protocol without degree reduction and exploit the fact that this protocol is completely non-interactive when used to compute bounded degree functions. The key generator provides the decryptor with a subset of the single query FE keys, where the subsets are guaranteed to have small pairwise intersections. This subset of keys enables the decryptor to recover sufficiently many shares of $C(x)$ which enables her to compute $C(x)$ via polynomial interpolation. However, an attacker with q keys only learns a share x_i in the clear if two subsets of keys intersect, and due to small pairwise intersections, this does not occur often enough for him learn sufficiently many shares of x, hence, by the guarantees of secret sharing, input x remains hidden.

3. Finally, they bootstrap the q-query FE for NC_1 to a q-query FE for all circuits using computational randomized encodings [9]. They must additionally use cover-free sets to ensure that fresh randomness is used for each randomized encoding.

Our construction replaces steps 1 and 2 with a inner product modulo p FE scheme, and then uses step 3 as in [39]. Thus, the construction of single key FE in step 1 by Sahai and Seyalioglu, and the nontrivial "MPC in the head" of step 2 can both be replaced by the simple abstraction of an inner product FE scheme. For step 3, observe that the bootstrapping theorem of [39] provides a method to bootstrap an FE for NC_1 that handles q queries to an FE for all polynomial-size circuits that is also secure against q queries. The bootstrapping relies on the result of Applebaum *et al.* [9, Theorem 4.11] which states that every polynomial time computable function f admits a perfectly correct computational randomized encoding of degree 3. In more detail, let $\mathcal{C}$ be a family of polynomial-size circuits. Let $C \in \mathcal{C}$ and let x be some input. Let $\widetilde{C}(x, R)$ be a randomized encoding of C that is computable by a constant depth circuit with respect to inputs x and R. Then consider a new family of circuits $\mathcal{G}$ defined by:

$$G_{C,\Delta}(x, R_1, \ldots, R_S) = \left\{ \widetilde{C}\Big(x; \underset{a \in \Delta}{\oplus} R_a\Big) : \ C \in \mathcal{C}, \ \Delta \subseteq [S] \right\},$$

for some sufficiently large S (quadratic in the number of queries q). As observed in [39], circuit $G_{C,\Delta}(\cdot, \cdot)$ is computable by a constant degree polynomial (one for each output bit). Given an FE scheme for $\mathcal{G}$, one may construct a scheme for $\mathcal{C}$ by having the decryptor first recover the output of $G_{C,\Delta}(x, R_1, \ldots, R_S)$ and then applying the decoder for the randomized encoding to recover $C(x)$.

However, to support q queries the decryptor must compute q randomized encodings, each of which needs fresh randomness. This is handled by hardcoding S random elements in the ciphertext and using random subsets $\Delta \subseteq [S]$ (which are cover-free with overwhelming probability) to compute fresh randomness $\underset{a \in \Delta}{\oplus} R_a$ for every query. The authors then conclude that bounded query FE for NC_1 suffices to construct a bounded query FE scheme for all circuits.

We observe that the ingredient required to bootstrap is *not FE for the entire circuit class* NC_1 but rather only the particular circuit class $\mathcal{G}$ as described above. This circuit class, being computable by degree 3 polynomials, may be supported by a linear FE scheme, via linearization of the degree 3 polynomials! To illustrate, let us consider FE secure only for a single key. Then, the functionality that the initial FE must support is exactly the randomized encoding of [9], which, indeed, is in NC_0. Now, to support q queries, we must ensure that each key uses a fresh piece of randomness, and this is provided using a cover-free set family S as in [39] – the key generator picks a random subset $\Delta \subseteq [S]$ and sums up its elements to obtain i.i.d. randomness for the key being requested. To obtain a random element in this manner, addition over the integers does not suffice, we must take addition modulo p. Here, our inner product modulo p construction comes to our rescue!

Putting it together, the encryptor encrypts all degree 3 monomials in the inputs $R_1, \ldots, R_S$ and $x_1, \ldots, x_\ell$. Note that this ciphertext is polynomial in size. Now, for a given circuit C, the keygen algorithm samples some $\Delta \subseteq [S]$ and computes the symbolic degree 3 polynomials which must be released to the decryptor. It then provides the linear FE keys to compute the same. By correctness and security of Linear FE as well as the randomizing polynomial construction, the decryptor learns $C(x)$ and nothing else. The final notion of security that we obtain is non-adaptive simulation based security NA-SIM [39,49], i.e. (poly, poly, 0) SIM security, where the adversary can request a polynomial number of pre-challenge keys, ask for polynomially sized challenge ciphertexts but may not request post-challenge keys. For more details, we refer the reader to Sect. 6. We note that the construction of [39] also achieves the stronger AD-SIM security, but for a scheme that supports only a *single* ciphertext and bounded number of keys. The bound on the number of ciphertexts is necessary due to a lower bound by [19]. The notion of single ciphertext, bounded key FE appears to be quite restrictive, hence we do not study AD-SIM security here.

We note that subsequent to our work, Agrawal and Rosen [6] used our adaptively secure mod p inner products FE scheme in a more sophisticated manner than we do here, to achieve ciphertext size that improves upon the construction of [39].

2   Background

In this section, we recall the hardness assumptions underlying the security of the schemes we will describe. The functionality and security definitions of functional and non-interactive controlled functional encryption schemes are given in the full version of the paper [4].

Our first scheme relies on the standard DDH assumption in ordinary (i.e., non-pairing-friendly) cyclic groups.

Definition 1. *In a cyclic group $\mathbb{G}$ of prime order q, the **Decision Diffie-Hellman** (DDH) problem is to distinguish the distributions $D_0 = \{(g, g^a, g^b, g^{ab}) \mid g \hookleftarrow \mathbb{G}, a, b \hookleftarrow \mathbb{Z}_q\}, D_1 = \{(g, g^a, g^b, g^c) \mid g \hookleftarrow \mathbb{G}, a, b, c \hookleftarrow \mathbb{Z}_q\}.$*

A variant of our first scheme relies on Paillier's composite residuosity assumption.

Definition 2 [51]**.** *Let $N = pq$, for prime numbers p, q. The **Decision Composite Residuosity** (DCR) problem is to distinguish the distributions $D_0 := \{z = z_0^N \bmod N^2 \mid z_0 \hookleftarrow \mathbb{Z}_N^*\}$ and $D_1 := \{z \hookleftarrow \mathbb{Z}_{N^2}^*\}.$*

Our third construction builds on the Learning-With-Errors (LWE) problem, which is known to be at least as hard as certain standard lattice problems in the worst case [20,54].

Definition 3. *Let q, α, m be functions of a parameter n. For a secret $\mathbf{s} \in \mathbb{Z}_q^n$, the distribution $A_{q,\alpha,\mathbf{s}}$ over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ is obtained by sampling $\mathbf{a} \hookleftarrow \mathbb{Z}_q^n$ and an $e \hookleftarrow D_{\mathbb{Z},\alpha q}$, and returning $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathbb{Z}_q^{n+1}$. The Learning With Errors*

problem $\mathsf{LWE}_{q,\alpha,m}$ *is as follows: For* $\mathbf{s} \hookleftarrow \mathbb{Z}_q^n$, *the goal is to distinguish between the distributions:*

$$D_0(\mathbf{s}) := U(\mathbb{Z}_q^{m \times (n+1)}) \quad and \quad D_1(\mathbf{s}) := (A_{q,\alpha,\mathbf{s}})^m.$$

We say that a PPT algorithm $\mathcal{A}$ *solves* $\mathsf{LWE}_{q,\alpha}$ *if it distinguishes* $D_0(\mathbf{s})$ *and* $D_1(\mathbf{s})$ *with non-negligible advantage (over the random coins of* $\mathcal{A}$ *and the randomness of the samples), with non-negligible probability over the randomness of* $\mathbf{s}$.

3 Fully Secure Functional Encryption for Inner Products from DDH

In this section, we show that an adaptation of the DDH-based construction of Abdalla *et al.* [2] provides full security under the standard DDH assumption. Like [2], the scheme computes inner products over $\mathbb{Z}$ as long as they land in a sufficiently small interval.

In comparison with the solution of Abdalla *et al.*, we only introduce one more group element in the ciphertext and all operations are just as efficient as in [2]. Our scheme is obtained by modifying [2] in the same way as Damgård's encryption scheme [27] was obtained from the Elgamal cryptosystem. The original DDH-based system of [2] encrypts a vector $\boldsymbol{y} = (y_1, \ldots, y_\ell) \in \mathbb{Z}_q^\ell$ by computing $(g^r, \{g^{y_i} \cdot h_i^r\}_{i=1}^\ell)$, where $\{h_i = g^{s_i}\}_{i=1}^\ell$ are part of the master public key and $\mathsf{sk}_{\boldsymbol{x}} = \sum_{i=1}^\ell s_i \cdot x_i \bmod q$ is the secret key associated with the vector $\boldsymbol{x} = (x_1, \ldots, x_\ell) \in \mathbb{Z}_q^\ell$. Here, we encrypt $\boldsymbol{y}$ in the fashion of Damgård's Elgamal, by computing $(g^r, h^r, \{g^{y_i} \cdot h_i^r\}_{i=1}^\ell)$. The decryption algorithm uses secret keys of the form $\mathsf{sk}_{\boldsymbol{x}} = (\sum_{i=1}^\ell s_i \cdot x_i, \sum_{i=1}^\ell t_i \cdot x_i)$, where $h_i = g^{s_i} \cdot h^{t_i}$ for each i and $\boldsymbol{s} = (s_1, \ldots, s_\ell) \in \mathbb{Z}_q^\ell$ and $\boldsymbol{t} = (t_1, \ldots, t_\ell) \in \mathbb{Z}_q^\ell$ are part of the master key msk.

The scheme and its security proof also build on ideas from the Cramer-Shoup cryptosystem [25,26]. Analogously to the bounded-collusion-resistant IBE schemes of Goldwasser *et al.* [36], the construction can be seen as an applying a hash proof system [26] with homomorphic properties over the key space. It also bears similarities with the broadcast encryption system of Dodis and Fazio [29] in the way to use hash proof systems to achieve adaptive security.

Setup$(1^\lambda, 1^\ell)$: Choose a cyclic group $\mathbb{G}$ of prime order $q > 2^\lambda$ with generators $g, h \hookleftarrow \mathbb{G}$. Then, for each $i \in \{1, \ldots, \ell\}$, sample $s_i, t_i \hookleftarrow \mathbb{Z}_q$ and compute $h_i = g^{s_i} \cdot h^{t_i}$. Define $\mathsf{msk} := \{(s_i, t_i)\}_{i=1}^\ell$ and

$$\mathsf{mpk} := \left(\mathbb{G}, g, h, \{h_i\}_{i=1}^\ell \right).$$

Keygen$(\mathsf{msk}, \boldsymbol{x})$: To generate a key for the vector $\boldsymbol{x} = (x_1, \ldots, x_\ell) \in \mathbb{Z}_q^\ell$, compute $\mathsf{sk}_{\boldsymbol{x}} = (s_{\boldsymbol{x}}, t_{\boldsymbol{x}}) = (\sum_{i=1}^\ell s_i \cdot x_i, \sum_{i=1}^\ell t_i \cdot x_i) = (\langle \boldsymbol{s}, \boldsymbol{x} \rangle, \langle \boldsymbol{t}, \boldsymbol{x} \rangle)$.

Encrypt$(\mathsf{mpk}, \boldsymbol{y})$: To encrypt a vector $\boldsymbol{y} = (y_1, \ldots, y_\ell) \in \mathbb{Z}_q^\ell$, sample $r \hookleftarrow \mathbb{Z}_q$ and compute

$$C = g^r, \qquad D = h^r, \qquad \{E_i = g^{y_i} \cdot h_i^r\}_{i=1}^\ell.$$

Return $C_{\boldsymbol{y}} = (C, D, E_1, \ldots, E_\ell)$.

Decrypt$(\mathsf{mpk}, \mathsf{sk}_{\boldsymbol{x}}, C_{\boldsymbol{y}})$**:** Given $\mathsf{sk}_{\boldsymbol{x}} = (s_{\boldsymbol{x}}, t_{\boldsymbol{x}})$, compute

$$E_{\boldsymbol{x}} = (\prod_{i=1}^{\ell} E_i^{x_i})/(C^{s_{\boldsymbol{x}}} \cdot D^{t_{\boldsymbol{x}}}).$$

Then, compute and output $\log_g(E_{\boldsymbol{x}})$.

The decryption algorithm requires to compute a discrete logarithm. This is in general too expensive. Like in [2], this can be circumvented by imposing that the computed inner product lies in an interval $\{0, \dots, L\}$, for some polynomially bounded integer L. Then, computing the required discrete logarithm may be performed in time $\widetilde{O}(L^{1/2})$ using Pollard's kangaroo method [52]. As reported in [11], this can be reduced to $\widetilde{O}(L^{1/3})$ operations by precomputing a table of size $\widetilde{O}(L^{1/3})$. Note that even though the functionality is limited (decryption may not be performed efficiently for all key vectors and for all message vectors), while proving security we will let the adversary query any key vector in $\mathbb{Z}_q^{\ell}$.

Before proceeding with the security proof, we would like to clarify that, although the scheme of [2] only decrypts values in a polynomial-size space, the usual complexity leveraging argument does not prove it fully secure via a polynomial reduction. Indeed, when ℓ is polynomial in λ, having the inner product $\langle \boldsymbol{y}, \boldsymbol{x} \rangle$ in a small interval does not mean that original vector $\boldsymbol{y} \in \mathbb{Z}_q^{\ell}$ lives in a polynomial-size universe. In Sect. 5, we show how to eliminate the small-interval restriction using Paillier's cryptosystem [51].

The security analysis uses similar arguments to those of Cramer and Shoup [25,26] in that it exploits the fact that mpk does not reveal too much information about the master secret key. At some step, the challenge ciphertext is generated using msk instead of the public key and, as long as msk retains a sufficient amount of entropy from the adversary's view, it will perfectly hide which vector among $\boldsymbol{y}_0, \boldsymbol{y}_1$ is actually encrypted. The reason why we can prove adaptive security is the fact that, as usual in security proofs relying on hash proof systems [25,26], the reduction knows the master secret key at any time. It can thus correctly answer all secret key queries without knowing the challenge messages $\boldsymbol{y}_0, \boldsymbol{y}_1$ beforehand.

The DDH-based scheme can easily be generalized so as to rely on weaker variants of DDH, like the Decision Linear assumption [15] or the Matrix DDH assumption [31].

Theorem 1. *The scheme provides full security under the* DDH *assumption.* (The proof is given in the full version of the paper [4]).

4   Full Security Under the **LWE** Assumption

We describe two LWE-based schemes: the first one for integer inner products of short integer vectors, the second one for inner products over a prime field $\mathbb{Z}_p$.

In both cases, the security relies on the hardness of a variant of the extended-LWE problem. The extended-LWE problem introduced by O'Neill et al. [50] and

further investigated in [7,20]. At a high level, the extended-LWE problem can be seen as $\mathsf{LWE}_{\alpha,q}$ with a fixed number m of samples, for which some extra information on the LWE noises is provided: the adversary is provided a given linear combination of the noise terms. More concretely, the problem is to distinguish between the distributions

$$\big(\mathbf{A},\ \mathbf{A}\cdot\boldsymbol{s}+\boldsymbol{e},\boldsymbol{z},\langle\boldsymbol{e},\boldsymbol{z}\rangle\big)\ \text{ and }\ \big(\mathbf{A},\ \boldsymbol{u},\ \boldsymbol{z},\ \langle\boldsymbol{e},\boldsymbol{z}\rangle\big),$$

where $\mathbf{A}\hookleftarrow\mathbb{Z}_q^{m\times n}, \boldsymbol{s}\hookleftarrow\mathbb{Z}_q^n, \boldsymbol{u}\hookleftarrow\mathbb{Z}_q^m, \boldsymbol{e}\hookleftarrow D_{\mathbb{Z},\alpha q}^m$, and $\boldsymbol{z}$ is sampled from a specified distribution. Note that in [50], a noise was added to the term $\langle\boldsymbol{e},\boldsymbol{z}\rangle$. The LWE to extended-LWE reductions from [7,20] do not require such an extra noise term.

We will use a variant of extended-LWE for which multiple hints $(\boldsymbol{z}_i,\langle\boldsymbol{e},\boldsymbol{z}_i\rangle)$ are given, for the same noise vector $\boldsymbol{e}$.

Definition 4 (Multi-hint Extended-LWE). *Let q,m,t be integers, α be a real and τ be a distribution over $\mathbb{Z}^{t\times m}$, all of them functions of a parameter n. The multi-hint extended-LWE problem $\mathsf{mheLWE}_{q,\alpha,m,t,\tau}$ is to distinguish between the distributions of the tuples*

$$\big(\mathbf{A},\ \mathbf{A}\cdot\boldsymbol{s}+\boldsymbol{e},\mathbf{Z},\ \mathbf{Z}\cdot\boldsymbol{e}\big)\ \ \text{and}\ \ \big(\mathbf{A},\ \boldsymbol{u},\ \mathbf{Z},\ \mathbf{Z}\cdot\boldsymbol{e}\big),$$

where $\mathbf{A}\hookleftarrow\mathbb{Z}_q^{m\times n}, \boldsymbol{s}\hookleftarrow\mathbb{Z}_q^n, \boldsymbol{u}\hookleftarrow\mathbb{Z}_q^m, \boldsymbol{e}\hookleftarrow D_{\mathbb{Z},\alpha q}^m$, and $\mathbf{Z}\hookleftarrow\tau$.

A reduction from LWE to mheLWE is presented in Subsect. 4.3.

4.1   Integer Inner Products of Short Integer Vectors

In the description hereunder, we consider the message space $\mathcal{P}=\{0,\dots,P-1\}^\ell$, for some integer P and where $\ell\in\mathsf{poly}(n)$ denotes the dimension of vectors to encrypt. Secret keys are associated with vectors in $\mathcal{V}=\{0,\dots,V-1\}^\ell$ for some integer V. As in the DDH case, inner products are evaluated over $\mathbb{Z}$. However, unlike our DDH-based construction, we can efficiently decrypt without confining inner product values within a small interval: here the inner product between the plaintext and key vectors belongs to $\{0,\dots,K-1\}$ with $K=\ell PV$, and it is possible to set parameters so that the scheme is secure under standard hardness assumptions while K is more than polynomial in the security parameter. We compute ciphertexts using a prime modulus q, with q significantly larger than K.

Setup$(1^n,1^\ell,P,V)$: Set integers $m,q\geq 2$, real $\alpha\in(0,1)$ and distribution τ over $\mathbb{Z}^{\ell\times m}$ as explained below. Set $K=\ell PV$. Sample $\mathbf{A}\hookleftarrow\mathbb{Z}_q^{m\times n}$ and $\mathbf{Z}\hookleftarrow\tau$. Compute $\mathbf{U}=\mathbf{Z}\cdot\mathbf{A}\in\mathbb{Z}_q^{\ell\times n}$. Define $\mathsf{mpk}:=(\mathbf{A},\mathbf{U},K,P,V)$ and $\mathsf{msk}:=\mathbf{Z}$.

Keygen$(\mathsf{msk},\boldsymbol{x})$: Given a vector $\boldsymbol{x}\in\mathcal{V}$, compute and return the secret key $\boldsymbol{z_x}:=\boldsymbol{x}^T\cdot\mathbf{Z}\in\mathbb{Z}^m$.

Encrypt$(\mathsf{mpk},\boldsymbol{y})$: To encrypt a vector $\boldsymbol{y}\in\mathcal{P}$, sample $\boldsymbol{s}\hookleftarrow\mathbb{Z}_q^n$, $\boldsymbol{e}_0\hookleftarrow D_{\mathbb{Z},\alpha q}^m$ and $\boldsymbol{e}_1\hookleftarrow D_{\mathbb{Z},\alpha q}^\ell$ and compute

$$\boldsymbol{c}_0=\mathbf{A}\cdot\boldsymbol{s}+\boldsymbol{e}_0\ \in\mathbb{Z}_q^m,$$
$$\boldsymbol{c}_1=\mathbf{U}\cdot\boldsymbol{s}+\boldsymbol{e}_1+\left\lfloor\frac{q}{K}\right\rfloor\cdot\boldsymbol{y}\ \in\mathbb{Z}_q^\ell.$$

Then, return $C := (\boldsymbol{c}_0, \boldsymbol{c}_1)$.

Decrypt$(\mathsf{mpk}, \boldsymbol{x}, \boldsymbol{z}_{\boldsymbol{x}}, C)$**:** Given a ciphertext $C := (\boldsymbol{c}_0, \boldsymbol{c}_1)$ and a secret key $\boldsymbol{z}_{\boldsymbol{x}}$ for $\boldsymbol{x} \in \mathcal{V}$, compute $\mu' = \langle \boldsymbol{x}, \boldsymbol{c}_1 \rangle - \langle \boldsymbol{z}_{\boldsymbol{x}}, \boldsymbol{c}_0 \rangle \bmod q$ and output the value $\mu \in \{-K+1, \dots, K-1\}$ that minimizes $|\lfloor \frac{q}{K} \rfloor \cdot \mu - \mu'|$.

Setting the Parameters. Let B_τ be such that with probability $\geq 1 - n^{-\omega(1)}$, each row of sample from τ has norm $\leq B_\tau$. As explained just below, correctness may be ensured by setting

$$\alpha^{-1} \geq K^2 B_\tau \omega(\sqrt{\log n}) \quad \text{and} \quad q \geq \alpha^{-1} \omega(\sqrt{\log n}).$$

The choice of τ is driven by the reduction from LWE to mheLWE (as summarized in Theorem 4), and more precisely from Lemma 4 (another constraint arises from the use of [35, Corollary 2.8] at the end of the security proof). We may choose $\tau = D_{\mathbb{Z}, \sigma_1}^{\ell \times m/2} \times (D_{\mathbb{Z}^{m/2}, \sigma_2, \boldsymbol{\delta}_1} \times \dots \times D_{\mathbb{Z}^{m/2}, \sigma_2, \boldsymbol{\delta}_\ell})$, where $\boldsymbol{\delta}_i \in \mathbb{Z}^\ell$ denotes the ith canonical vector, and the standard deviation parameters satisfy $\sigma_1 = \Theta(\sqrt{n \log m} \max(\sqrt{m}, K))$ and $\sigma_2 = \Theta(n^{7/2} m^{1/2} \max(m, K^2) \log^{5/2} m)$.

To ensure security based on $\mathsf{LWE}_{q, \alpha', m}$ in dimension $\geq c \cdot n$ for some $c \in (0, 1)$ via Theorems 2 and 4 below, one may further impose that $\ell \leq (1 - c) \cdot n$ and $m = \Theta(n \log q)$, to obtain $\alpha' = \Omega(\alpha/(n^6 K \log^2 q \log^{5/2} n))$. Note that $\mathsf{LWE}_{q, \alpha', m}$ enjoys reductions from lattice problems when $q \geq \Omega(\sqrt{n}/\alpha')$.

Combining the security and correctness requirements, we may choose $\alpha' = 1/((n \log q)^{O(1)} \cdot K^2)$ and $q = \Omega(\sqrt{n}/\alpha')$, resulting in LWE parameters that make LWE resist all known attacks running in time 2^λ, as long as $n \geq \widetilde{\Omega}(\lambda \log K)$.

Decryption Correctness. To show the correctness of the scheme, we first observe that, modulo q:

$$\mu' = \langle \boldsymbol{x}, \boldsymbol{c}_1 \rangle - \langle \boldsymbol{z}_{\boldsymbol{x}}, \boldsymbol{c}_0 \rangle = \lfloor q/K \rfloor \cdot \langle \boldsymbol{x}, \boldsymbol{y} \rangle + \langle \boldsymbol{x}, \boldsymbol{e}_1 \rangle - \langle \boldsymbol{z}_{\boldsymbol{x}}, \boldsymbol{e}_0 \rangle.$$

Below, we show that the magnitude of the term $\langle \boldsymbol{x}, \boldsymbol{e}_1 \rangle - \langle \boldsymbol{z}_{\boldsymbol{x}}, \boldsymbol{e}_0 \rangle$ is $\leq \ell V B_\tau \alpha q \omega(\sqrt{\log n})$ with probability $\geq 1 - n^{-\omega(1)}$. Thanks to the choices of α and q, the latter upper bound is $\leq \lfloor q/K \rfloor / 4$, which suffices to guarantee decryption correctness.

Note that $\boldsymbol{e}_1$ is an integer Gaussian vector of dimension ℓ and standard deviation $\alpha q \geq \omega(\sqrt{\log n})$, and that $\|\boldsymbol{x}\| \leq \sqrt{\ell} V$. As a result, we have that $|\langle \boldsymbol{x}, \boldsymbol{e}_1 \rangle| \leq \sqrt{\ell} V \alpha q \omega(\sqrt{\log n})$ holds with probability $1 - n^{-\omega(1)}$. Similarly, as $\|\boldsymbol{z}_{\boldsymbol{x}}\| \leq \ell V B_\tau$, we obtain that $|\langle \boldsymbol{z}_{\boldsymbol{x}}, \boldsymbol{e}_0 \rangle| \leq \ell V B_\tau \alpha q \omega(\sqrt{\log n})$ holds with probability $1 - n^{-\omega(1)}$.

Full Security. In order to prove adaptive security of the scheme, we use the multi-hint extended-LWE from Definition 4. Before we provide the formal proof, we provide some intuition.

Intuition. Here we describe some challenges in proving adaptive security for our LWE construction. To begin we describe the approach used by Abdalla et al. [2] in showing selective security for a similar construction. In the selective game, the adversary must announce the challenge vectors $\mathbf{y}_0, \mathbf{y}_1$ at the outset of the game.

By definition of an admissible adversary, every query $\mathbf{x}^i$ made must satisfy the property that $\langle \boldsymbol{x}^i, (\boldsymbol{y}_0 - \mathbf{y}_1) \rangle = 0$ (over $\mathbb{Z}$) for all i. For ease of exposition, consider challenge messages $\mathbf{y}_0, \mathbf{y}_1$ that only differ in the last co-ordinate. Then, the simulator knows at the very beginning of the game, the subspace within which all queries must lie. Since the secret key is structured as $(\mathbf{x}^i)^T \mathbf{Z}$, it suffices for the simulator to pick all but the final column of $\mathbf{Z}$ in order to answer all legitimate key requests. It can set the public parameters by constructing all except one row of $\mathbf{U}$ using its choice of $\mathbf{Z}$, and receiving the final $\mathbf{u}_\ell$ from the LWE oracle. Now the challenge ciphertext can be embedded along this dimension to argue security.

In the adaptive game however, the simulator cannot know in advance which subspace the adversary's queries will lie in, hence it must pick the entire master secret key $\mathbf{Z}$ to answer key requests. Given that the simulator has no secrets, it is unclear how it may leverage the adversary. To handle this, our approach is to carefully analyze the entropy loss that occurs in the master secret $\mathbf{Z}$ via that keys seen by the adversary. We show that despite seeing linear relations involving $\mathbf{Z}$, there is enough residual entropy left in the master secret so that the challenge ciphertext created using this appears uniform to the adversary.

To the best of our knowledge, this proof technique has not been used in prior constructions of LWE based FE systems, which mostly rely on a "punctured trap-door" approach. This approach roughly provides the simulator with a trapdoor that can be used to answer key requests but vanishes w.h.p for the challenge. Our simulator does not use trapdoors, but relies on an argument about entropy leakage as described above. We now proceed with the formal proof.

Theorem 2. *Assume that $\ell \leq n^{O(1)}$, $m \geq 4n \log_2 q$, $q > \ell K^2$ and τ is as described above. Then the functional encryption scheme above is fully secure, under the* mheLWE$_{q,\alpha,m,\ell,\tau}$ *hardness assumption.*

Proof. The proof proceeds with a sequence of games that starts with the real game and ends with a game in which the adversary's advantage is negligible. For each i, we call S_i the event that the adversary wins in Game i.

Game 0: This is the genuine full security game. Namely: the adversary $\mathcal{A}$ is given the master public key mpk; in the challenge phase, adversary $\mathcal{A}$ comes up with two distinct vectors $\boldsymbol{y}_0, \boldsymbol{y}_1 \in \mathcal{P}$ and receives an encryption C of $\boldsymbol{y}_\beta$ for $\beta \hookleftarrow \{0, 1\}$ sampled by the challenger; when $\mathcal{A}$ halts, it outputs $\beta' \in \{0, 1\}$ and S_0 is the event that $\beta' = \beta$. Note that any vector $\boldsymbol{x} \in \mathcal{V}$ queried by $\mathcal{A}$ to the secret key extraction oracle must satisfy $\langle \boldsymbol{x}, \boldsymbol{y}_0 \rangle = \langle \boldsymbol{x}, \boldsymbol{y}_1 \rangle$ over $\mathbb{Z}$ if $\mathcal{A}$ is a legitimate adversary.

Game 1: We modify the generation of $C = (\boldsymbol{c}_0, \boldsymbol{c}_1)$ in the challenge phase. Namely, at the outset of the game, the challenger picks $\boldsymbol{s} \hookleftarrow \mathbb{Z}_q^n$, $\boldsymbol{e}_0 \hookleftarrow D_{\mathbb{Z},\alpha q}^m$ (which may be chosen ahead of time) as well as $\mathbf{Z} \hookleftarrow \tau$. The master public key mpk is computed by setting $\mathbf{U} = \mathbf{Z} \cdot \mathbf{A} \bmod q$. In the challenge phase, the challenger picks a random bit $\beta \hookleftarrow \{0, 1\}$ and encrypts $\boldsymbol{y}_\beta$ by computing (modulo q)

$$c_0 = \mathbf{A} \cdot \boldsymbol{s} + \boldsymbol{e}_0,$$
$$c_1 = \mathbf{Z} \cdot c_0 - \mathbf{Z} \cdot \boldsymbol{e}_0 + \boldsymbol{e}_1 + \lfloor q/K \rfloor \cdot \boldsymbol{y}_\beta,$$

with $\boldsymbol{e}_1 \hookleftarrow D^\ell_{\mathbb{Z}, \alpha q}$. As the distribution of C is the same as in Game 0, we have $\Pr[S_1] = \Pr[S_0]$.

Game 2: We modify again the generation of $C = (c_0, c_1)$ in the challenge phase. Namely, the challenger picks $\boldsymbol{u} \hookleftarrow \mathbb{Z}_q^m$, sets $c_0 = \boldsymbol{u}$ and computes c_1 using $c_0, \mathbf{Z}$ and $\boldsymbol{e}_0$ as in Game 1.

Under the mheLWE hardness assumption with $t = \ell$, this modification has no noticeable effect on the behavior of $\mathcal{A}$. Below, we prove that $\Pr[S_2] \approx 1/2$, which completes the proof of the theorem.

Let $\boldsymbol{x}^i \in \mathcal{V}$ be the vectors corresponding to the secret key queries made by $\mathcal{A}$. As $\mathcal{A}$ is a legitimate adversary, we have $\langle \boldsymbol{x}^i, \boldsymbol{y}_0 \rangle = \langle \boldsymbol{x}^i, \boldsymbol{y}_1 \rangle$ over $\mathbb{Z}$ for each secret key query $\boldsymbol{x}^i$. Let $g \neq 0$ be the gcd of the coefficients of $\boldsymbol{y}_1 - \boldsymbol{y}_0$ and define $\boldsymbol{y} = (y_1, \ldots, y_\ell) = \frac{1}{g}(\boldsymbol{y}_1 - \boldsymbol{y}_0)$. We have that $\langle \boldsymbol{x}^i, \boldsymbol{y} \rangle = 0$ (over $\mathbb{Z}$) for all i. Consider the lattice $\{\boldsymbol{x} \in \mathbb{Z}^\ell : \langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0\}$: all the queries $\boldsymbol{x}^i$ must belong to that lattice. Without loss of generality, we assume the n_0 first entries of $\boldsymbol{y}$ are zero (for some n_0), and all remaining entries are non-zero. Further, the rows of the following matrix form a basis of a full-dimensional sublattice:

$$\mathbf{X}_{top} = \begin{pmatrix} \mathbf{I}_{n_0} & & & & & \\ \hline & -y_{n_0+2} & y_{n_0+1} & & & \\ & & -y_{n_0+3} & y_{n_0+2} & & \\ & & & \ddots & & \ddots & \\ & & & & -y_\ell & y_{\ell-1} \end{pmatrix} \in \mathbb{Z}^{(\ell-1)\times\ell}.$$

We may assume that through the secret key queries, the adversary learns exactly $\mathbf{X}_{top}\mathbf{Z}$, as all the queried vectors $\boldsymbol{x}^i$ can be obtained as rational combinations of the rows of $\mathbf{X}_{top}$.

Let $\mathbf{X}_{bot} = \boldsymbol{y}^T \in \mathbb{Z}^{1\times\ell}$. Consider the matrix $\mathbf{X} \in \mathbb{Z}_q^{\ell\times\ell}$ obtained by putting $\mathbf{X}_{top}$ on top of $\mathbf{X}_{bot}$. We claim that $\mathbf{X}$ is invertible modulo q. To see this, observe that

$$\mathbf{X}\mathbf{X}^T = \begin{pmatrix} \mathbf{I}_{n_0} & & & & & \\ \hline & y_{n_0+1}^2 + y_{n_0+2}^2 & -y_{n_0+1} \cdot y_{n_0+3} & & & \\ & -y_{n_0+1} \cdot y_{n_0+3} & y_{n_0+2}^2 + y_{n_0+3}^2 & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & -y_{\ell-2} \cdot y_\ell & y_{\ell-1}^2 + y_\ell^2 & \\ \hline & & & & & \|\boldsymbol{y}\|^2 \end{pmatrix}$$

It can be proved by induction that its determinant is

$$\det(\mathbf{X}\mathbf{X}^T) = \Big(\prod_{k=n_0+2}^{\ell-1} y_k^2 \Big) \cdot \|\boldsymbol{y}\|^4.$$

As each of the y_k's is small and non-zero, they are all non-zero modulo prime q. Similarly, the integer $(\sum_{k=n_0+1}^{\ell} y_k^2)$ is non-zero and $< \ell P^2 < q$. This shows that $(\det \mathbf{X})^2 \neq 0 \bmod q$, which implies that $\mathbf{X}$ is invertible modulo q.

In Game 2, we have $\boldsymbol{c}_1 = \mathbf{Z}\boldsymbol{u} - \boldsymbol{f} + \lfloor q/K \rfloor \cdot \boldsymbol{y}_\beta$, with $\boldsymbol{f} := -\mathbf{Z}\boldsymbol{e}_0 + \boldsymbol{e}_1$. We write:

$$\boldsymbol{c}_1 = \mathbf{X}^{-1} \cdot \mathbf{X} \cdot \left(\mathbf{Z}\boldsymbol{u} - \boldsymbol{f} + \lfloor q/K \rfloor \cdot \boldsymbol{y}_\beta \right) \bmod q.$$

We will show that the distribution of $\mathbf{X} \cdot \boldsymbol{c}_1 \bmod q$ is (almost) independent of β. As $\mathbf{X}$ is (almost) independent of β and invertible over $\mathbb{Z}_q$, this implies that the distribution of $\boldsymbol{c}_1$ is (almost) independent of β and $\Pr[S_2] \approx 1/2$.

The first $\ell - 1$ entries of $\mathbf{X} \cdot \boldsymbol{c}_1$ do not depend on β because $\mathbf{X}_{top} \cdot \boldsymbol{y}_0 = \mathbf{X}_{top} \cdot \boldsymbol{y}_1 \bmod q$.

It remains to prove that the last entry of $\mathbf{X} \cdot \boldsymbol{c}_1 \bmod q$ is (almost) independent of β. For this, we show that the residual distribution of $\mathbf{X}_{bot}\mathbf{Z}$ given the tuple $(\mathbf{A}, \mathbf{Z}\mathbf{A}, \mathbf{X}_{top}\mathbf{Z})$ has high entropy. Using (a variant of) the leftover hash lemma with randomness $\mathbf{X}_{bot}\mathbf{Z}$ and seed $\boldsymbol{u}$, we will then conclude that given $(\mathbf{A}, \mathbf{Z}\mathbf{A}, \mathbf{X}_{top}\mathbf{Z})$, the pair $(\boldsymbol{u}, \mathbf{X}_{bot}\mathbf{Z}\boldsymbol{u})$ is close to uniform and hence statistically hides $\lfloor q/K \rfloor \cdot \boldsymbol{y}_\beta$ in $\boldsymbol{c}_1$.

Write $\mathbf{A} = (\mathbf{A}_1^T | \mathbf{A}_2^T)^T$ with $\mathbf{A}_1, \mathbf{A}_2 \in \mathbb{Z}_q^{(m/2) \times n}$. Similarly, write $\mathbf{Z} = (\mathbf{Z}_1 | \mathbf{Z}_2)$ with $\mathbf{Z}_1, \mathbf{Z}_2 \in \mathbb{Z}_q^{\ell \times (m/2)}$. Recall that by construction, every entry of $\mathbf{Z}_1$ is independently sampled from a zero-centered integer Gaussian of standard deviation parameter $\sigma_1 = \Theta(\sqrt{n \log m} \max(\sqrt{m}, K))$. Further, every entry of $\mathbf{Z}_2$ is independently sampled from a (not zero-centered) integer Gaussian of standard deviation parameter σ_2 that is larger than σ_1.

Lemma 1. *Conditioned on $(\mathbf{A}, \mathbf{Z}\mathbf{A}, \mathbf{X}_{top}\mathbf{Z}_1)$, the row vector $\mathbf{X}_{bot}\mathbf{Z}_1$ is distributed as $\boldsymbol{c} + D_{\|\boldsymbol{y}\|^2 \mathbb{Z}^{m/2}, \|\boldsymbol{y}\|\sigma_1, -\boldsymbol{c}}$ for some vector $\boldsymbol{c}$ that depends only on $\mathbf{X}_{top}\mathbf{Z}_1$.*

Proof. Thanks to [35, Corollary 2.8], we have that $\mathbf{Z}_2\mathbf{A}_2$ is within $2^{-\Omega(n)}$ statistical distance to uniform. It hence statistically hides the term $\mathbf{Z}_1\mathbf{A}_1$ in $\mathbf{Z}\mathbf{A} = \mathbf{Z}_1\mathbf{A}_1 + \mathbf{Z}_2\mathbf{A}_2$, and we obtain that given $(\mathbf{A}, \mathbf{Z}\mathbf{A})$, the distribution of each entry of $\mathbf{Z}_1$ is still $D_{\mathbb{Z}, \sigma_1}$.

Note that in $\mathbf{X}_{top}\mathbf{Z}_1$ and $\mathbf{X}_{bot}\mathbf{Z}_1$, matrices $\mathbf{X}_{top}$ and $\mathbf{X}_{bot}$ act in parallel on the columns of $\mathbf{Z}_1$. To prove the claim, it suffices to consider the distribution of $\mathbf{X}_{bot}\boldsymbol{z}$ conditioned on $\mathbf{X}_{top}\boldsymbol{z}$, with $\boldsymbol{z}$ sampled from $D_{\mathbb{Z}^\ell, \sigma_1}$. Let $\boldsymbol{b} = \mathbf{X}_{top}\boldsymbol{z} \in \mathbb{Z}^{\ell-1}$ and fix $\boldsymbol{z}_0 \in \mathbb{Z}^\ell$ arbitrary such that $\boldsymbol{b} = \mathbf{X}_{top}\boldsymbol{z}_0$. The distribution of $\boldsymbol{z}$ given that $\mathbf{X}_{top}\boldsymbol{z} = \boldsymbol{b}$ is $\boldsymbol{z}_0 + D_{\Lambda, \sigma_1, -\boldsymbol{z}_0}$, with $\Lambda = \{\boldsymbol{x} \in \mathbb{Z}^\ell : \mathbf{X}_{top}\boldsymbol{x} = \boldsymbol{0}\}$. By construction of $\mathbf{X}$, we have that $\Lambda = \mathbb{Z}\boldsymbol{y}$. As a result, the conditional distribution of $\mathbf{X}_{bot}\boldsymbol{z}$ is $c + D_{\|\boldsymbol{y}\|^2\mathbb{Z}, \|\boldsymbol{y}\|\sigma_1, -c}$ with $c = \langle \boldsymbol{y}, \boldsymbol{z}_0 \rangle \in \mathbb{Z}$. $\qquad \square$

Now, let us write $\boldsymbol{u} = (\boldsymbol{u}_1^T | \boldsymbol{u}_2^T)^T$ for vectors $\boldsymbol{u}_1, \boldsymbol{u}_2 \in \mathbb{Z}_q^{m/2}$. We have $\mathbf{X}_{bot}\mathbf{Z}\boldsymbol{u} = \mathbf{X}_{bot}\mathbf{Z}_1\boldsymbol{u}_1 + \mathbf{X}_{bot}\mathbf{Z}_2\boldsymbol{u}_2$. Thanks to the claim above and the result of [35, Corollay 2.8], we obtain that the distribution of $(\boldsymbol{u}_1, \langle D_{\|\boldsymbol{y}\|^2\mathbb{Z}^{m/2}, \|\boldsymbol{y}\|\sigma_1, -\boldsymbol{c}}, \boldsymbol{u}_1 \rangle)$ is within $2^{-\Omega(n)}$ statistical distance to uniform (note that $\|\boldsymbol{y}\|^2$ is invertible modulo q, that $D_{\|\boldsymbol{y}\|^2\mathbb{Z}^{m/2}, \|\boldsymbol{y}\|\sigma_1, -\boldsymbol{c}} = \|\boldsymbol{y}\|^2 \cdot$

$D_{\mathbb{Z}^{m/2},\sigma_1/\|\boldsymbol{y}\|,-\boldsymbol{c}/\|\boldsymbol{y}\|^2}$, and that $\sigma_1/\|\boldsymbol{y}\|$ satisfies the assumption of [35, Corollary 2.8]). This implies that given $(\mathbf{A}, \mathbf{Z}\mathbf{A}, \mathbf{X}_{top}\mathbf{Z})$, the pair $(\boldsymbol{u}, \mathbf{X}_{bot}\mathbf{Z}\boldsymbol{u})$ is close to uniform, which completes the security proof.   □

4.2 Inner Products Modulo a Prime p

We now modify the LWE-based scheme above so that it enables secure functional encryption for inner products modulo prime p. The plaintext and key vectors now belong to $\mathbb{Z}_p^\ell$.

Note that the prior scheme evaluates inner products over the integers and is insecure if ported as is to the modulo p setting. To see this, consider the following simple attack in which the adversary requests a single key $\mathbf{x}$ so that integer inner product with the challenge messages $\mathbf{y}_0$ and $\mathbf{y}_1$ are different by a multiple of p. Since the functionality posits that the inner product evaluations only agree modulo p, this is an admissible query. However, since decryption is performed over $\mathbb{Z}_q$ with q much larger than p, the adversary can easily distinguish. To prevent this attack, we scale the encrypted message by a factor of q/p (instead of $\lfloor q/K \rfloor$ as in the previous scheme): decryption modulo q forces arithmetic modulo p on the underlying plaintext.

A related difficulty in adapting the previous LWE-based scheme to modular inner products is the distribution of the noise component after inner product evaluation. Ciphertexts are manipulated modulo q, which internally manipulates plaintexts modulo p. If implemented naively, the carries of the plaintext computations may spill outside of the plaintext slots and bias the noise components of the ciphertexts. This may result in distinguishing attacks. To handle this, we take q a multiple of p. This adds some technical complications, as $\mathbb{Z}_q$ is hence not a field anymore.

A different attack is that the adversary may request keys for vectors that are linearly dependent modulo p but linearly independent over the integers. Note that with ℓ such queries, the attacker can recover the master secret key. To prevent this attack, we modify the scheme in that the authority is now stateful and keeps a record of all key queries made so far, so that it can make sure that key queries that are linearly dependent modulo p remain so modulo q. We also take q a power of p to simplify the implementation of this idea.

We note that for our application to bounded query FE for all circuits, all queries will be linearly independent modulo p, hence we will not require a stateful keygen. For details, see Sect. 6.

We now describe our scheme for inner products modulo p.

Setup$(1^n, 1^\ell, p)$: Set integers $m, q = p^k$ for some integer k, real $\alpha \in (0,1)$ and distribution τ over $\mathbb{Z}_q^{\ell \times m}$ as explained below. Sample $\mathbf{A} \hookleftarrow \mathbb{Z}_q^{m \times n}$ and $\mathbf{Z} \hookleftarrow \tau$. Compute $\mathbf{U} = \mathbf{Z} \cdot \mathbf{A} \in \mathbb{Z}_q^{\ell \times n}$. Define $\mathsf{mpk} := (\mathbf{A}, \mathbf{U})$ and $\mathsf{msk} := \mathbf{Z}$.

Keygen$(\mathsf{msk}, \boldsymbol{x}, \mathsf{st})$: Given a vector $\boldsymbol{x} \in \mathbb{Z}_p^\ell$, and an internal state st, compute the secret key $\boldsymbol{z_x}$ as follows. Recall that Keygen is a stateful algorithm with empty initial State st. At any point in the scheme execution, State st contains at most ℓ tuples $(\boldsymbol{x}_i, \overline{\boldsymbol{x}}_i, \boldsymbol{z}_i)$ where the $\boldsymbol{x}_i$'s are (a subset of the) key queries

that have been made so far, and the $(\overline{\boldsymbol{x}}_i, \boldsymbol{z}_i)$'s are the corresponding secret keys. If $\boldsymbol{x}$ is linearly independent from the $\boldsymbol{x}_i$'s modulo p, set $\overline{\boldsymbol{x}} = \boldsymbol{x} \in \mathbb{Z}^\ell$ (with coefficients in $[0, p)$), $\boldsymbol{z}_{\boldsymbol{x}} = \overline{\boldsymbol{x}}^T \cdot \mathbf{Z} \in \mathbb{Z}^m$ and add $(\boldsymbol{x}, \overline{\boldsymbol{x}}, \boldsymbol{z}_{\boldsymbol{x}})$ to st. If $\boldsymbol{x} = \sum_i k_i \boldsymbol{x}_i \bmod p$ for some k_i's in $[0, p)$, then set $\overline{\boldsymbol{x}} = \sum_i k_i \overline{\boldsymbol{x}}_i \in \mathbb{Z}^\ell$ and $\boldsymbol{z}_{\boldsymbol{x}} = \sum_i k_i \boldsymbol{z}_i \in \mathbb{Z}^m$. In both cases, return $(\overline{\boldsymbol{x}}, \boldsymbol{z}_{\boldsymbol{x}})$.

Encrypt(mpk, $\boldsymbol{y}$): To encrypt a vector $\boldsymbol{y} \in \mathbb{Z}_p^\ell$, sample $\boldsymbol{s} \hookleftarrow \mathbb{Z}_q^n$, $\boldsymbol{e}_0 \hookleftarrow D_{\mathbb{Z}, \alpha q}^m$ and $\boldsymbol{e}_1 \hookleftarrow D_{\mathbb{Z}, \alpha q}^\ell$ and compute

$$\boldsymbol{c}_0 = \mathbf{A} \cdot \boldsymbol{s} + \boldsymbol{e}_0 \in \mathbb{Z}_q^m,$$
$$\boldsymbol{c}_1 = \mathbf{U} \cdot \boldsymbol{s} + \boldsymbol{e}_1 + p^{k-1} \cdot \boldsymbol{y} \in \mathbb{Z}_q^\ell.$$

Then, return $C := (\boldsymbol{c}_0, \boldsymbol{c}_1)$.

Decrypt(mpk, $(\overline{\boldsymbol{x}}, \boldsymbol{z}_{\boldsymbol{x}}), C$): Given a ciphertext $C := (\boldsymbol{c}_0, \boldsymbol{c}_1)$ and a secret key $(\overline{\boldsymbol{x}}, \boldsymbol{z}_{\boldsymbol{x}})$ for $\boldsymbol{x} \in \mathbb{Z}_p^\ell$, compute $\mu' = \langle \overline{\boldsymbol{x}}, \boldsymbol{c}_1 \rangle - \langle \boldsymbol{z}_{\boldsymbol{x}}, \boldsymbol{c}_0 \rangle \bmod q$ and output the value $\mu \in \mathbb{Z}_p$ that minimizes $|p^{k-1} \cdot \mu - \mu'|$.

Decryption Correctness. Correctness derives from the following observation:

$$\mu' = \langle \overline{\boldsymbol{x}}, \boldsymbol{c}_1 \rangle - \langle \boldsymbol{z}_{\boldsymbol{x}}, \boldsymbol{c}_0 \rangle = p^{k-1} \cdot (\langle \boldsymbol{x}, \boldsymbol{y} \rangle \bmod p) + \langle \overline{\boldsymbol{x}}, \boldsymbol{e}_1 \rangle - \langle \boldsymbol{z}_{\boldsymbol{x}}, \boldsymbol{e}_0 \rangle \bmod q.$$

By adapting the proof of the first LWE-based scheme, we can show that the magnitude of the term $\langle \overline{\boldsymbol{x}}, \boldsymbol{e}_1 \rangle - \langle \boldsymbol{z}_{\boldsymbol{x}}, \boldsymbol{e}_0 \rangle$ is $\leq \ell^2 p^2 B_\tau \alpha q \omega(\sqrt{\log n})$ with probability $\geq 1 - n^{-\omega(1)}$. This follows from the bound $\|\boldsymbol{z}_{\boldsymbol{x}}\| \leq \ell \|\overline{\boldsymbol{x}}\| \leq \ell^2 p^2 B_\tau$.

Setting the Parameters. The main difference with the previous LWE-based scheme with respect to parameter conditions is the choice of q of the form $q = p^k$ instead of q prime. As explained just above, correctness may be ensured by setting

$$\alpha^{-1} \geq \ell^2 p^3 B_\tau \omega(\sqrt{\log n}) \quad \text{and} \quad q \geq \alpha^{-1} \omega(\sqrt{\log n}).$$

The choice of τ is driven by Lemma 2 below (the proof requires that σ_1 is large) and the reduction from LWE to mheLWE (as summarized in Theorem 4), and more precisely from Lemma 4. We may choose $\tau = D_{\mathbb{Z}, \sigma_1}^{\ell \times m/2} \times (D_{\mathbb{Z}^{m/2}, \sigma_2, \boldsymbol{\delta}_1} \times \ldots \times D_{\mathbb{Z}^{m/2}, \sigma_2, \boldsymbol{\delta}_\ell})$, where $\boldsymbol{\delta}_i \in \mathbb{Z}^\ell$ denotes the ith canonical vector, and the standard deviation parameters satisfy $\sigma_1 = \Theta(\sqrt{n \log m} \max(\sqrt{m}, K'))$ and $\sigma_2 = \Theta(n^{7/2} m^{1/2} \max(m, K'^2) \log^{5/2} m)$, with $K' = (\sqrt{\ell} p)^\ell$.

To ensure security based on $\mathsf{LWE}_{q, \alpha', m}$ in dimension $\geq c \cdot n$ for some $c \in (0, 1)$ via Theorems 2 and 4 below, one may further impose that $\ell \leq (1 - c) \cdot n$ and $m = \Theta(n \log q)$, to obtain $\alpha' = \Omega(\alpha/(n^6 K' \log^2 q \log^{5/2} n))$. Remember that $\mathsf{LWE}_{q, \alpha', m}$ enjoys reductions from lattice problems when $q \geq \Omega(\sqrt{n}/\alpha')$.

Note that the parameter conditions make the scheme efficiency degrade quickly when ℓ increases, as K' is exponential in ℓ. Assume that $p \leq n^{O(1)}$ and $\ell = \Omega(\log n)$. Then σ_1, σ_2, $1/\alpha$, $1/\alpha'$ and q can all be set as $2^{\widetilde{O}(\ell)}$. To maintain security against all $2^{o(\lambda)}$ attacks, one may set $n = \widetilde{\Theta}(\ell \lambda)$.

Theorem 3. *Assume that $\ell \leq n^{O(1)}$, $m \geq 4n \log_2 q$ and τ is as described above. Then the stateful functional encryption scheme above is fully secure, under the* $\mathsf{mheLWE}_{q,\alpha,m,\ell,\tau}$ *hardness assumption.*

Proof. The sequence of games in the proof of Theorem 2 can be adapted to the modified scheme. The main difficulty is to show that in the adapted version of the last game, the winning probability is close to $1/2$. Let us recall that game.

Game $2'$: At the outset of the game, the challenger picks $\boldsymbol{s} \hookleftarrow \mathbb{Z}_q^n$, $\boldsymbol{e}_0 \hookleftarrow D_{\mathbb{Z},\alpha q}^m$ as well as $\mathbf{Z} \hookleftarrow \tau$. The master public key mpk is computed by setting $\mathbf{U} = \mathbf{Z} \cdot \mathbf{A} \bmod q$ and is provided to the adversary. In the challenge phase, adversary $\mathcal{A}$ comes up with two distinct vectors $\boldsymbol{y}_0, \boldsymbol{y}_1 \in \mathbb{Z}_p^\ell$. The challenger picks a random bit $\beta \hookleftarrow \{0,1\}$, $\boldsymbol{u} \hookleftarrow \mathbb{Z}_q^m$ and encrypts $\boldsymbol{y}_\beta$ by computing (modulo q)

$$\boldsymbol{c}_0 = \boldsymbol{u},$$
$$\boldsymbol{c}_1 = \mathbf{Z} \cdot \boldsymbol{c}_0 - \mathbf{Z} \cdot \boldsymbol{e}_0 + \boldsymbol{e}_1 + p^{k-1} \cdot \boldsymbol{y}_\beta,$$

with $\boldsymbol{e}_1 \hookleftarrow D_{\mathbb{Z},\alpha q}^\ell$. Note that any vector $\boldsymbol{x} \in \mathbb{Z}_p^\ell$ queried by $\mathcal{A}$ to the secret key extraction oracle must satisfy $\langle \boldsymbol{x}, \boldsymbol{y}_0 \rangle = \langle \boldsymbol{x}, \boldsymbol{y}_1 \rangle \bmod p$ if $\mathcal{A}$ is a legitimate adversary. Adversary $\mathcal{A}$ is then given a secret key $(\overline{\boldsymbol{x}}, \boldsymbol{z_x})$ as in the real scheme. When $\mathcal{A}$ halts, it outputs $\beta' \in \{0,1\}$ and wins in the event that $\beta' = \beta$.

Define $\boldsymbol{y} = \boldsymbol{y}_1 - \boldsymbol{y}_0 \in \mathbb{Z}_p^\ell$. Let $\boldsymbol{x}_i \in \mathbb{Z}_p^\ell$ be the vectors corresponding to the secret key queries made by $\mathcal{A}$. As $\mathcal{A}$ is a legitimate adversary, we have $\langle \boldsymbol{x}_i, \boldsymbol{y} \rangle = 0 \bmod p$ for each secret key query $\boldsymbol{x}_i$.

We consider the view of the adversary after it has made exactly j key queries that are linearly independent modulo p, for each j from 0 up to $\ell - 1$. In fact, counter j may stop increasing before reaching $\ell - 1$, but without loss of generality, we may assume that it eventually reaches $\ell - 1$. We are to show by induction that for any j, the view of the adversary is almost independent of β. In particular, for all $j < \ell - 1$, this implies that the $(j+1)$th linearly independent key query is almost (statistically) independent of β. It also implies, for $j = \ell - 1$, that the adversary's view through Game $2'$ is almost independent of β, which is exactly what we are aiming for. In what follows, we take $j \in \{0, \ldots, \ell - 1\}$, and assume that state st is independent from β. We also assume that the jth private key query occurs after the challenge phase since the adversary's view is trivially independent of β before the generation of the challenge ciphertext.

At this stage, the state st contains exactly j tuples $(\boldsymbol{x}_i, \overline{\boldsymbol{x}}_i, \boldsymbol{z}_i)$, where the vectors $\{\boldsymbol{x}_i\}_{i=1}^j$ form a $\mathbb{Z}_p$-basis of a subspace of the $(\ell - 1)$-dimensional vector space $\boldsymbol{y}^\perp := \{\boldsymbol{x} \in \mathbb{Z}_p^\ell : \langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0 \bmod p\}$. From $\boldsymbol{y}$, we deterministically extend $\{\boldsymbol{x}_i\}_{i=1}^j$ into a basis of $\boldsymbol{y}^\perp$ that is statistically independent of β. A way to interpret this is to imagine that the challenger makes dummy private key queries $\{\boldsymbol{x}_i\}_{i=j+1}^{\ell-1}$ for itself so as to get a full basis of $\boldsymbol{y}^\perp$ and creates the corresponding $\{\overline{\boldsymbol{x}}_i\}_{i=j+1}^{\ell-1}$ in $\mathbb{Z}^\ell$. We define $\mathbf{X}_{top} \in \mathbb{Z}^{(\ell-1)\times\ell}$ as the matrix whose ith row is $\overline{\boldsymbol{x}}_i$ for all i, including the genuine and dummy keys. Through the secret key queries, the adversary learns at most $\mathbf{X}_{top}\mathbf{Z} \in \mathbb{Z}^{(\ell-1)\times m}$.

Let $\boldsymbol{x}' \in \mathbb{Z}_p^\ell$ be a vector that does not belong to $\boldsymbol{y}^\perp$, and $\mathbf{X}_{bot} \in \mathbb{Z}^{1\times\ell}$ be the canonical lift of $(\boldsymbol{x}')^T$ over the integers. Consider the matrix $\mathbf{X} \in \mathbb{Z}^{\ell\times\ell}$ obtained

by putting $\mathbf{X}_{top}$ on top of $\mathbf{X}_{bot}$. By construction, the matrix $\mathbf{X}$ is invertible modulo p, and hence modulo $q = p^k$. Also, by induction and construction, $\mathbf{X} \in \mathbb{Z}^{\ell \times \ell}$ is statistically independent of $\beta \in \{0, 1\}$.

In Game $2'$, we have $\boldsymbol{c}_1 = \mathbf{Z}\boldsymbol{u} - \boldsymbol{f} + p^{k-1} \cdot \boldsymbol{y}_\beta$, with $\boldsymbol{f} := -\mathbf{Z}\boldsymbol{e}_0 + \boldsymbol{e}_1$. We write:

$$\boldsymbol{c}_1 = \mathbf{X}^{-1} \cdot \mathbf{X} \cdot \left(\mathbf{Z}\boldsymbol{u} - \boldsymbol{f} + p^{k-1} \cdot \boldsymbol{y}_\beta \right) \mod q.$$

We will show that the distribution of $\mathbf{X} \cdot \boldsymbol{c}_1 \mod q$ is (almost) independent of β. As the matrix $\mathbf{X}$ is independent of $\beta \in \{0, 1\}$ and invertible over $\mathbb{Z}_q$, this implies that the distribution of $\boldsymbol{c}_1$ is statistically independent of β (recall that $\mathbf{X}$ is information-theoretically known to $\mathcal{A}$, which means that, if $\boldsymbol{c}_1$ carries any noticeable information on β, so does $\mathbf{X} \cdot \boldsymbol{c}_1 \mod q$). This ensures that the winning probability in Game $2'$ is negligibly far from $1/2$.

First, the first $\ell - 1$ entries of $\mathbf{X} \cdot \boldsymbol{c}_1$ do not depend on β because we have the equality $p^{k-1} \cdot \mathbf{X}_{top} \cdot \boldsymbol{y}_0 = p^{k-1} \cdot \mathbf{X}_{top} \cdot \boldsymbol{y}_1 \mod q$ by construction of $\mathbf{X}_{top}$.

It remains to prove that the last entry of $\mathbf{X} \cdot \boldsymbol{c}_1 \mod q$ is (almost) independent of β. Let us write $\mathbf{A} = (\mathbf{A}_1^T | \mathbf{A}_2^T)^T$ with $\mathbf{A}_1, \mathbf{A}_2 \in \mathbb{Z}_q^{(m/2) \times n}$. Similarly, we also write $\mathbf{Z} = (\mathbf{Z}_1 | \mathbf{Z}_2)$ with $\mathbf{Z}_1, \mathbf{Z}_2 \in \mathbb{Z}^{\ell \times (m/2)}$. Recall that by construction, every entry of $\mathbf{Z}_1$ is independently sampled from a zero-centered integer Gaussian of standard deviation parameter $\sigma_1 = \Theta(\sqrt{n \log m} \max(\sqrt{m}, K'))$ with $K' = (\sqrt{\ell}p)^\ell$. Further, every entry of $\mathbf{Z}_2$ is independently sampled from a (not zero-centered) integer Gaussian of standard deviation parameter σ_2 that is larger than σ_1.

Lemma 2. *Conditioned on $(\mathbf{A}, \mathbf{ZA}, \mathbf{X}_{top}\mathbf{Z}_1)$, the row vector $\mathbf{X}_{bot}\mathbf{Z}_1 \mod p$ is within negligible statistical distance from the uniform distribution over $\mathbb{Z}_p^{m/2}$.*

Proof. Thanks to [35, Corollary 2.8], we have that $\mathbf{Z}_2\mathbf{A}_2$ is within $2^{-\Omega(n)}$ statistical distance to uniform over $\mathbb{Z}_q^{(\ell-1) \times m}$. It hence statistically hides the term $\mathbf{Z}_1\mathbf{A}_1$ in $\mathbf{ZA} = \mathbf{Z}_1\mathbf{A}_1 + \mathbf{Z}_2\mathbf{A}_2 \mod q$, and we obtain that given $(\mathbf{A}, \mathbf{ZA})$, the distribution of each entry of $\mathbf{Z}_1$ is still $D_{\mathbb{Z}, \sigma_1}$.

Note that in $\mathbf{X}_{top}\mathbf{Z}_1$ and $\mathbf{X}_{bot}\mathbf{Z}_1$, matrices $\mathbf{X}_{top}$ and $\mathbf{X}_{bot}$ act in parallel on the columns of $\mathbf{Z}_1$. To prove the claim, it suffices to consider the distribution of $\mathbf{X}_{bot}\boldsymbol{z}$ conditioned on $\mathbf{X}_{top}\boldsymbol{z}$, with $\boldsymbol{z}$ sampled from $D_{\mathbb{Z}^\ell, \sigma_1}$. Let $\boldsymbol{b} = \mathbf{X}_{top}\boldsymbol{z} \in \mathbb{Z}^{\ell-1}$ and fix $\boldsymbol{z}_0 \in \mathbb{Z}^\ell$ arbitrary such that $\boldsymbol{b} = \mathbf{X}_{top}\boldsymbol{z}_0$. The distribution of $\boldsymbol{z}$ given that $\mathbf{X}_{top}\boldsymbol{z} = \boldsymbol{b}$ is $\boldsymbol{z}_0 + D_{\Lambda, \sigma_1, -\boldsymbol{z}_0}$, with $\Lambda = \{\boldsymbol{x} \in \mathbb{Z}^\ell : \mathbf{X}_{top}\boldsymbol{x} = \mathbf{0}\}$ (where the equality holds over the integers). Note that Λ is a 1-dimensional lattice in $\mathbb{Z}^\ell$.

We can write $\Lambda = \boldsymbol{y}' \cdot \mathbb{Z}$, for some $\boldsymbol{y}' \in \mathbb{Z}^\ell$. Note that there exists $\alpha \in \mathbb{Z}_p \setminus \{0\}$ such that $\boldsymbol{y}' = \alpha \cdot \boldsymbol{y} \mod p$: otherwise, the vector $\boldsymbol{y}'/p$ would belong to $\Lambda \setminus \boldsymbol{y}' \cdot \mathbb{Z}$, contradicting the definition of $\boldsymbol{y}'$. Further, we have $\|\boldsymbol{y}'\| = \det \Lambda \leq \det \Lambda'$, where Λ' is the lattice spanned by the rows of $\mathbf{X}_{top}$ (see, e.g., [48], for properties on orthogonal lattices). Hadamard's bound implies that $\|\boldsymbol{y}'\| \leq (\sqrt{\ell}p)^{\ell-1}$.

By [35, Corollary 2.8], the fact that $\sigma_1 \geq \sqrt{n}(\sqrt{\ell}p)^\ell$ implies that the distribution $(D_{\Lambda, \sigma_1, -\boldsymbol{z}_0} \mod p\Lambda)$ is within $2^{-\Omega(n)}$ statistical distance from the uniform distribution over $\Lambda/p\Lambda \simeq \boldsymbol{y}\mathbb{Z}_p$. We conclude that the conditional distribution of $(\mathbf{X}_{bot}\boldsymbol{z} \mod p)$ is within exponentially small statistical distance from

the uniform distribution over $\mathbb{Z}_p$ (here we use the facts that p is prime and that $\mathbf{X}_{bot}\boldsymbol{y} \neq 0 \bmod p$, by construction of $\mathbf{X}_{bot}$). $\qquad\square$

Now, write $\boldsymbol{u} = (\boldsymbol{u}_1^T|\boldsymbol{u}_2^T)^T$ with $\boldsymbol{u}_1, \boldsymbol{u}_2 \in \mathbb{Z}_q^{m/2}$. We have $\mathbf{X}_{bot}\mathbf{Z}\boldsymbol{u} = \mathbf{X}_{bot}\mathbf{Z}_1\boldsymbol{u}_1 + \mathbf{X}_{bot}\mathbf{Z}_2\boldsymbol{u}_2$. Thanks to Lemma 2 and a variant of the leftover hash lemma modulo $q = p^k$ (given in the full version of the paper [4]), we obtain that conditioned on $(\mathbf{A}, \mathbf{Z}\mathbf{A}, \mathbf{X}_{top}\mathbf{Z})$, the distribution of $(\boldsymbol{u}_1, \mathbf{X}_{bot}\mathbf{Z}_1\boldsymbol{u}_1)$ is within $2^{-\Omega(n)}$ statistical distance to uniform modulo q (here we used the assumption that $m \geq k + n/(\log p)$). This implies that given $(\mathbf{A}, \mathbf{Z}\mathbf{A}, \mathbf{X}_{top}\mathbf{Z})$, the pair $(\boldsymbol{u}, \mathbf{X}_{bot}\mathbf{Z}\boldsymbol{u})$ is close to uniform, which completes the security proof. $\quad\square$

4.3 Hardness of Multi-hint Extended-LWE

In this section, we prove the following theorem, which shows that for some parameters, the mheLWE problem is no easier than the LWE problem.

Theorem 4. *Let $n \geq 100$, $q \geq 2$, $t < n$ and m with $m = \Omega(n \log n)$ and $m \leq n^{O(1)}$. There exists $\xi \leq O(n^4 m^2 \log^{5/2} n)$ and a distribution τ over $\mathbb{Z}^{t \times m}$ such that the following statements hold:*

- *There is a reduction from $\mathsf{LWE}_{q,\alpha,m}$ in dimension $n - t$ to $\mathsf{mheLWE}_{q,\alpha\xi,m,t,\tau}$ that reduces the advantage by at most $2^{\Omega(t-n)}$,*
- *It is possible to sample from τ in time polynomial in n,*
- *Each entry of matrix τ is an independent discrete Gaussian $\tau_{i,j} = D_{\mathbb{Z},\sigma_{i,j},\boldsymbol{c}_{i,j}}$ for some $\boldsymbol{c}_{i,j}$ and $\sigma_{i,j} \geq \Omega(\sqrt{mn \log m})$,*
- *With probability $\geq 1 - n^{-\omega(1)}$, all rows from a sample from τ have norms $\leq \xi$.*

Our reduction from LWE to mheLWE proceeds as the reduction from LWE to extended-LWE from [20], using the matrix gadget from [45] to handle the multiple hints. We first reduce LWE to the following variant of LWE in which the first samples are noise-free. This problem generalizes the first-is-errorless LWE problem from [20].

Definition 5 (First-are-errorless LWE). *Let q, α, m, t be functions of a parameter n. The first-are-errorless LWE problem $\mathsf{faeLWE}_{q,\alpha,m,t}$ is defined as follows: For $\boldsymbol{s} \hookleftarrow \mathbb{Z}_q^n$, the goal is to distinguish between the following two scenarios. In the first, all m samples are uniform over $\mathbb{Z}_q^n \times \mathbb{Z}_q$. In the second, the first t samples are from $A_{q,\{0\},\boldsymbol{s}}$ (where $\{0\}$ denotes the distribution that is deterministically zero) and the rest are from $A_{q,\alpha,\boldsymbol{s}}$.*

Lemma 3. *For any $n > t$, $m, q \geq 2$, and $\alpha \in (0, 1)$, there is an efficient reduction from $\mathsf{LWE}_{q,\alpha,m}$ in dimension $n - t$ to $\mathsf{faeLWE}_{q,\alpha,m,t}$ in dimension n that reduces the advantage by at most 2^{-n+t+1}.*

The proof, postponed to the appendices, is a direct adaptation of the one of [20, Lemma 4.3].

In our reduction from faeLWE to mheLWE, we use the following gadget matrix from [45, Corollary 10]. It generalizes the matrix construction from [20, Claim 4.6].

Lemma 4. *Let n, m_1, m_2 with $100 \leq n \leq m_1 \leq m_2 \leq n^{O(1)}$. Let $\sigma_1, \sigma_2 > 0$ be standard deviation parameters such that $\sigma_1 \geq \Omega(\sqrt{m_1 n \log m_1})$, $m_1 \geq \Omega(n \log(\sigma_1 n))$ and $\sigma_2 \geq \Omega(n^{5/2}\sqrt{m_1}\sigma_1^2 \log^{3/2}(m_1\sigma_1))$. Let $m = m_1 + m_2$. There exists a probabilistic polynomial time algorithm that given n, m_1, m_2 (in unary) and σ_1, σ_2 as inputs, outputs $\mathbf{G} \in \mathbb{Z}^{m \times m}$ such that:*

- *The top $n \times m$ submatrix of $\mathbf{G}$ is within statistical distance $2^{-\Omega(n)}$ of $\tau = D_{\mathbb{Z}, \sigma_1}^{n \times m_1} \times (D_{\mathbb{Z}^{m_2}, \sigma_2, \boldsymbol{\delta}_1} \times \ldots \times D_{\mathbb{Z}^{m_2}, \sigma_2, \boldsymbol{\delta}_n})^T$ with $\boldsymbol{\delta}_i$ denoting the ith canonical unit vector,*
- *We have $|\det(\mathbf{G})| = 1$ and $\|\mathbf{G}^{-1}\| \leq O(\sqrt{nm_2}\sigma_2)$, with probability $\geq 1 - 2^{-\Omega(n)}$.*

Lemma 5. *Let $n, m_1, m_2, m, \sigma_1, \sigma_2, \tau$ be as in Lemma 4, and $\xi \geq \Omega(\sqrt{nm_2}\sigma_2)$. Let $q \geq 2$, $t \leq n$, $\alpha \geq \Omega(\sqrt{n}/q)$. Let τ_t be the distribution obtained by keeping only the first t rows from a sample from τ. There is a (dimension-preserving) reduction from $\mathsf{faeLWE}_{q,\alpha,m,t}$ to $\mathsf{mheLWE}_{q,2\alpha\xi,m,t,\tau_t}$ that reduces the advantage by at most $2^{-\Omega(n)}$.*

Proof. Let us first describe the reduction. Let $(\mathbf{A}, \boldsymbol{b}) \in \mathbb{Z}_q^m \times \mathbb{Z}_q$ be the input, which is either sampled from the uniform distribution, or from distribution $A_{q,\{0\},\boldsymbol{s}}^t \times A_{q,\alpha,\boldsymbol{s}}^{m-t}$ for some fixed $\boldsymbol{s} \hookleftarrow \mathbb{Z}_q^n$. Our objective is to distinguish between the two scenarios, using an mheLWE oracle. We compute $\mathbf{G}$ as in Lemma 4 and let $\mathbf{U} = \mathbf{G}^{-1}$. We let $\mathbf{Z} \in \mathbb{Z}^{t \times m}$ denote the matrix formed by the top t rows of $\mathbf{G}$, and let $\mathbf{U}' \in \mathbb{Z}^{m \times (m-t)}$ denote the matrix formed by the right $m - t$ columns of $\mathbf{U}$. By construction, we have $\mathbf{Z}\mathbf{U}' = \mathbf{0}$. We define $\mathbf{A}' = \mathbf{U} \cdot \mathbf{A} \bmod q$. We sample $\boldsymbol{f} \hookleftarrow D_{\alpha q(\xi^2 \mathbf{I} - \mathbf{U}'\mathbf{U}'^T)^{1/2}}$ (thanks to Lemma 4 and the choice of ξ, the matrix $\xi^2 \mathbf{I} - \mathbf{U}'\mathbf{U}'^T$ is positive definite). We sample $\boldsymbol{e}'$ from $\{0\}^t \times D_{\alpha q}^{m-t}$ and define $\boldsymbol{b}' = \mathbf{U} \cdot (\boldsymbol{b} + \boldsymbol{e}') + \boldsymbol{f}$. We then sample $\boldsymbol{c} \hookleftarrow D_{\mathbb{Z}^m - \boldsymbol{b}', \sqrt{2}\alpha\xi q}$, and define $\boldsymbol{h} = \mathbf{Z}(\boldsymbol{f} + \boldsymbol{c})$.

Finally, the reduction calls the mheLWE oracle on input $(\mathbf{A}', \boldsymbol{b}' + \boldsymbol{c}, \mathbf{Z}, \boldsymbol{h})$, and outputs the reply.

Correctness is obtained by showing that distribution $A_{q,\{0\},\boldsymbol{s}}^t \times A_{q,\alpha,\boldsymbol{s}}^{m-t}$ is mapped to the mheLWE "LWE" distribution and that the uniform distribution is mapped to the mheLWE "uniform" distribution, up to $2^{-\Omega(n)}$ statistical distances (we do not discuss these tiny statistical discrepancies below). The proof is identical to the reduction analysis in the proof of [20, Lemma 4.7]. □

Theorem 4 is obtained by combining Lemmas 3, 4 and 5.

5 Constructions Based on Paillier

In this section, we show how to remove the main limitation of our DDH-based system which is its somewhat expensive decryption algorithm. To this end, we use Paillier's cryptosystem [51] and the property that, for an RSA modulus $N = pq$, the multiplicative group $\mathbb{Z}_{N^2}^*$ contains a subgroup of order N (generated by $(N + 1)$) in which the discrete logarithm problem is easy. We also rely on the observation [21,22] that combining the Paillier and Elgamal encryption schemes makes it possible to decrypt without knowing the factorization of $N = pq$.

5.1   Computing Inner Products over $\mathbb{Z}$

In the following scheme, key vectors $\boldsymbol{x}$ and message vectors $\boldsymbol{y}$ are assumed to be of bounded norm $\|\boldsymbol{x}\| \leq X$ and $\|\boldsymbol{y}\| \leq Y$, respectively. The bounds X and Y are chosen so that $X \cdot Y < N$, where N is the composite modulus of Paillier's cryptosystem. Decryption allows to recover $\langle \boldsymbol{x}, \boldsymbol{y} \rangle \bmod N$, which is exactly $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ over the integers, thanks to the norm bounds. The security proof further requires that $\ell Y^2 < N$ and we thus assume $X, Y < (N/\ell)^{1/2}$.

Setup$(1^\lambda, 1^\ell, X, Y)$: Choose safe prime numbers $p = 2p' + 1$, $q = 2q' + 1$ with sufficiently large primes $p', q' > 2^{l(\lambda)}$, for some polynomial l, and compute $N = pq > XY$. Then, sample $g' \hookleftarrow \mathbb{Z}_{N^2}^*$ and compute $g = g'^{2N} \bmod N^2$, which generates the subgroup of $(2N)$th residues in $\mathbb{Z}_{N^2}^*$ with overwhelming probability. Then, sample an integer vector $\boldsymbol{s} = (s_1, \ldots, s_\ell)^T \hookleftarrow D_{\mathbb{Z}^\ell, \sigma}$ with discrete Gaussian entries of standard deviation $\sigma > \sqrt{\lambda} \cdot N^{5/2}$ and compute $h_i = g^{s_i} \bmod N^2$. Define

$$\mathsf{mpk} := \left(N, g, \{h_i\}_{i=1}^\ell, Y \right)$$

and $\mathsf{msk} := (\{s_i\}_{i=1}^\ell, X)$. The prime numbers p, p', q, q' are no longer needed.

Keygen$(\mathsf{msk}, \boldsymbol{x})$: To generate a key for the vector $\boldsymbol{x} = (x_1, \ldots, x_\ell) \in \mathbb{Z}^\ell$ with $\|\boldsymbol{x}\| \leq X$, compute $\mathsf{sk}_{\boldsymbol{x}} = \sum_{i=1}^\ell s_i \cdot x_i$ over $\mathbb{Z}$.

Encrypt$(\mathsf{mpk}, \boldsymbol{y})$: To encrypt a vector $\boldsymbol{y} = (y_1, \ldots, y_\ell) \in \mathbb{Z}^\ell$ with $\|\boldsymbol{y}\| \leq Y$, sample $r \hookleftarrow \{0, \ldots, \lfloor N/4 \rfloor\}$ and compute

$$C_0 = g^r \bmod N^2,$$
$$C_i = (1 + y_i N) \cdot h_i^r \bmod N^2, \qquad \forall i \in \{1, \ldots, \ell\}.$$

Return $C_{\boldsymbol{y}} = (C_0, C_1, \ldots, C_\ell) \in \mathbb{Z}_{N^2}^{\ell+1}$.

Decrypt$(\mathsf{mpk}, \mathsf{sk}_{\boldsymbol{x}}, C_{\boldsymbol{y}})$: Given $\mathsf{sk}_{\boldsymbol{x}} \in \mathbb{Z}$, compute

$$C_{\boldsymbol{x}} = \left(\prod_{i=1}^\ell C_i^{x_i} \right) \cdot C_0^{-\mathsf{sk}_{\boldsymbol{x}}} \bmod N^2.$$

Then, compute and output $\log_{(1+N)}(C_{\boldsymbol{x}}) = \frac{C_{\boldsymbol{x}} - 1 \bmod N^2}{N}$.

As in previous constructions (including those of [2]), our security proof requires inner products to be evaluated over $\mathbb{Z}$, although the decryptor technically computes $\langle \boldsymbol{x}, \boldsymbol{y} \rangle \bmod N$. The reason is that, since secret keys are computed over the integers, our security proof only goes through if the adversary is restricted to only obtain secret keys for vectors $\boldsymbol{x}$ such that $\langle \boldsymbol{x}, \boldsymbol{y}_0 \rangle = \langle \boldsymbol{x}, \boldsymbol{y}_1 \rangle$ over $\mathbb{Z}$.

Theorem 5. *The scheme provides full security under the* DCR *assumption.* (The proof is available in the full version of the paper [4]).

5.2   A Construction for Inner Products over $\mathbb{Z}_N$

Here, we show that our first DCR-based scheme can be adapted in order to compute the inner product $\langle \boldsymbol{y}, \boldsymbol{x} \rangle \bmod N$ instead of computing it over $\mathbb{Z}$. To do this, a first difficulty is that, as in our LWE-based system, private keys are computed over the integers and the adversary may query private keys for vectors that are linearly dependent over $\mathbb{Z}_N^\ell$ but independent over $\mathbb{Z}^\ell$. This problem is addressed as previously, by having the authority keep track of all previously revealed private keys. As in our LWE-based construction over $\mathbb{Z}_p$, we also need to increase the size of private keys (by a factor $\approx \ell$) because we have to use a different information-theoretic argument in the last step of the security proof.

Setup$(1^\lambda, 1^\ell)$**:** Choose safe prime numbers $p = 2p'+1$, $q = 2q'+1$ with sufficiently large primes $p', q' > 2^{l(\lambda)}$, for some polynomial l, and compute $N = pq$. Then, sample $g' \hookleftarrow \mathbb{Z}_{N^2}^*$ and compute $g = g'^{2N} \bmod N^2$, which generates the subgroup of $(2N)$th residues in $\mathbb{Z}_{N^2}^*$ with overwhelming probability. Then, sample an integer vector $\boldsymbol{s} = (s_1, \ldots, s_\ell)^T \hookleftarrow D_{\mathbb{Z}^\ell, \sigma}$ with discrete Gaussian entries of standard deviation $\sigma > \sqrt{\lambda}(\sqrt{\ell} N)^{\ell+1}$ and compute $h_i = g^{s_i} \bmod N^2$. Define $\mathsf{msk} := \{s_i\}_{i=1}^\ell$ and

$$\mathsf{mpk} := \left(N, g, \{h_i\}_{i=1}^\ell \right).$$

Keygen$(\mathsf{msk}, \boldsymbol{x}, \mathsf{st})$**:** To generate the jth secret key $\mathsf{sk}_{\boldsymbol{x}}$ for a vector $\boldsymbol{x} \in \mathbb{Z}_N^\ell$ using the master secret key msk and an (initially empty) internal state st, a stateful algorithm is used. At any time, st contains at most ℓ tuples $(\boldsymbol{x}_i, \overline{\boldsymbol{x}}_i, \boldsymbol{z}_{\boldsymbol{x}_i})$ where the $(\overline{\boldsymbol{x}}_i, \boldsymbol{z}_{\boldsymbol{x}_i})$'s are the previously revealed secret keys and the $\boldsymbol{x}_i$ are the corresponding vectors.
  – If $\boldsymbol{x}$ is linearly independent from the $\boldsymbol{x}_i$'s modulo N, set $\overline{\boldsymbol{x}} = \boldsymbol{x} \in \mathbb{Z}^\ell$ (with coefficients in $[0, N)$), $\boldsymbol{z}_{\boldsymbol{x}} = \langle \boldsymbol{s}, \boldsymbol{x} \rangle \in \mathbb{Z}$ and add $(\boldsymbol{x}, \overline{\boldsymbol{x}}, \boldsymbol{z}_{\boldsymbol{x}})$ to st.
  – If $\boldsymbol{x} = \sum_i k_i \boldsymbol{x}_i \bmod N$ for some coefficients $\{k_i\}_{i \leq j-1}$ in $\mathbb{Z}_N$, then compute $\overline{\boldsymbol{x}} = \sum_i k_i \cdot \overline{\boldsymbol{x}}_i \in \mathbb{Z}^\ell$ and $\boldsymbol{z}_{\boldsymbol{x}} = \sum_i k_i \cdot \boldsymbol{z}_{\boldsymbol{x}_i} \in \mathbb{Z}^m$.
  In either case, return $\mathsf{sk}_{\boldsymbol{x}} = (\overline{\boldsymbol{x}}, \boldsymbol{z}_{\boldsymbol{x}})$.

Encrypt$(\mathsf{mpk}, \boldsymbol{y})$**:** To encrypt a vector $\boldsymbol{y} = (y_1, \ldots, y_\ell) \in \mathbb{Z}_N^\ell$, sample $r \hookleftarrow \{0, \ldots, \lfloor N/4 \rfloor\}$ and compute

$$C_0 = g^r \bmod N^2,$$
$$C_i = (1 + y_i N) \cdot h_i^r \bmod N^2, \qquad \forall i \in \{1, \ldots, \ell\}.$$

Return $C_{\boldsymbol{y}} = (C_0, C_1, \ldots, C_\ell) \in \mathbb{Z}_{N^2}^{\ell+1}$.

Decrypt$(\mathsf{mpk}, \mathsf{sk}_{\boldsymbol{x}}, C_{\boldsymbol{y}})$**:** Given $\mathsf{sk}_{\boldsymbol{x}} = (\overline{\boldsymbol{x}}, \boldsymbol{z}_{\boldsymbol{x}}) \in \mathbb{Z}^\ell \times \mathbb{Z}$ with $\overline{\boldsymbol{x}} = (\overline{x}_1, \ldots, \overline{x}_\ell)$, compute

$$C_{\boldsymbol{x}} = \left(\prod_{i=1}^\ell C_i^{\overline{x}_i} \right) \cdot C_0^{-\boldsymbol{z}_{\boldsymbol{x}}} \bmod N^2.$$

Then, compute and output $\log_{(1+N)}(C_{\boldsymbol{x}}) = \frac{C_{\boldsymbol{x}} - 1 \bmod N^2}{N}$.

From a security standpoint, the following result is proved in the full version of the paper [4].

Theorem 6. *The above stateful scheme provides full security under the* DCR *assumption.*

6 Bootstrapping Linear FE to Efficient Bounded FE for All Circuits

In this section, we describe how to compile our Linear FE scheme, denoted by LinFE which computes linear functions modulo p (for us $p = 2$), into a bounded collusion FE scheme for all circuits, denoted by BddFE. The underlying scheme LinFE is assumed to be AD-IND secure, which, by [49], is equivalent to non-adaptive simulation secure NA-SIM, since linear functions are "preimage sampleable". We refer the reader to [49] for more details.

Let $\mathcal{C}$ be a family of polynomial-size circuits. Let $C \in \mathcal{C}$ and let $\mathbf{x}$ be some input. Let $\widetilde{C}(\mathbf{x}, R)$ be a randomized encoding of C that is computable by a constant depth circuit with respect to inputs x and R (see [9]). Then consider a new family of circuits $\mathcal{G}$ defined by:

$$G_{C,\Delta}(x, R_1, \ldots, R_S) = \left\{ \widetilde{C}\Big(x; \underset{a \in \Delta}{\oplus} R_a \Big) : \ C \in \mathcal{C}, \ \Delta \subseteq [S] \right\},$$

for some S to be chosen below. As observed in [39, Sect. 6], circuit $G_{C,\Delta}(\cdot, \cdot)$ is computable by a constant degree polynomial (one for each output bit). Given an FE scheme for $\mathcal{G}$, one may construct a scheme for $\mathcal{C}$ by having the decryptor first recover the output of $G_{C,\Delta}(\mathbf{x}, R_1, \ldots, R_S)$ and then applying the decoder for the randomized encoding to recover $C(\mathbf{x})$.

Note that to support q queries the decryptor must compute q randomized encodings, each of which needs fresh randomness. As shown above, this is handled by hardcoding sufficiently many random elements in the ciphertext and taking a random subset sum of these to generate fresh random bits for each query. As in [39], the parameters are chosen so that the subsets form a cover-free system, so that every random subset yields fresh randomness (with overwhelming probability).

In more details, we let the set S, v, m be parameters to the construction. Let Δ_i for $i \in [q]$ be a uniformly random subset of S of size v. To support q queries, we identify the set $\Delta_i \subseteq S$ with query i. If $v = O(\lambda)$ and $S = O(\lambda \cdot q^2)$ then the sets Δ_i are cover-free with high probability. For details, we refer the reader to [39, Sect. 5]. We now proceed to describe our construction. Let $L \triangleq (\ell + S \cdot m)^3$, where $m \in \text{poly}(\lambda)$ is the size of the random input in the randomized encoding and ℓ is the length of the messages to be encrypted.

BddFE.Setup$(1^\lambda, 1^\ell)$**:** Upon input the security parameter λ and the message space $\mathcal{M} = \{0, 1\}^\ell$, invoke $(\mathsf{mpk}, \mathsf{msk}) = \mathsf{LinFE.Setup}(1^\lambda, 1^L)$ and output it.

BddFE.KeyGen(msk, C)**:** Upon input the master secret key and a circuit C, do:

1. Sample a uniformly random subset $\Delta \subseteq S$ of size v.
2. Express $C(\mathbf{x})$ by $G_{C,\Delta}(\mathbf{x}, R_1, \ldots, R_S)$, which in turn can be expressed as a sequence of degree 3 polynomials $P_1, \ldots, P_k$, where $k \in \mathsf{poly}(\lambda)$.
3. Linearize each polynomial P_i and let P'_i be its vector of coefficients. Note that the ordering of the coefficients can be aribitrary but should be public.
4. Output $\mathsf{BddFE.SK}_C = \{\mathsf{SK}_i = \mathsf{LinFE.KeyGen}(\mathsf{LinFE.msk}, P'_i)\}_{i \in [k]}$.

BddFE.Enc$(\mathbf{x}, \mathsf{mpk})$**:** Upon input the public key and the plaintext $\mathbf{x}$, do:

1. Sample $R_1, \ldots, R_S \leftarrow \{0, 1\}^m$.
2. Compute all symbolic monomials of degree 3 in the variables $x_1, \ldots, x_\ell$ and $R_{i,j}$ for $i \in [S]$, $j \in [m]$. The number of such monomials is $L = (\ell + S \cdot m)^3$. Arrange them according to the public ordering and denote the resulting vector by $\boldsymbol{y}$.
3. Output $\mathsf{CT}_{\mathbf{x}} = \mathsf{LinFE.Enc}(\mathsf{LinFE.mpk}, \boldsymbol{y})$.

BddFE.Dec$(\mathsf{mpk}, \mathsf{CT}_{\mathbf{x}}, \mathsf{SK}_C)$**:** Upon input a ciphertext $\mathsf{CT}_{\mathbf{x}}$ for vector $\mathbf{x}$, and a secret key $\mathsf{SK}_C = \{\mathsf{SK}_i\}_{i \in [k]}$ for circuit C, do the following:

1. Compute $G_{C,\Delta}(\mathbf{x}, R_1, \ldots, R_S) = \{P_i(\mathbf{Y})\}_{i \in [k]} = \{\mathsf{LinFE.Dec}(\mathsf{CT}_{\mathbf{x}}, \mathsf{SK}_i)\}_{i \in [k]}$.
2. Run the decoder for the randomized encoding to recover $C(\mathbf{x})$ from $G_{C,\Delta}(\mathbf{x}, R_1, \ldots, R_S)$.

Correctness follows from the correctness of LinFE and the correctness of randomized encodings.

Security. The definition for q-NA-SIM security is provided in the full version of the paper [4]. We proceed to describe our simulator $\mathsf{Bdd.Sim}$. Let $\mathsf{RE.Sim}$ be the simulator guaranteed by the security of randomized encodings and $\mathsf{LinFE.Sim}$ be the simulator guaranteed by the security of the LinFE scheme.

Simulator Bdd.Sim$\big(\{C_i, C_i(\mathbf{x}), \mathsf{SK}_i\}_{i \in [q^*]}\big)$**:** The simulator $\mathsf{Bdd.Sim}$ receives the secret key queries C_i, the corresponding (honestly generated) secret keys SK_i and the values $C_i(\mathbf{x})$ for $i \in [q^*]$ where $q^* \leq q$, and must simulate the ciphertext $\mathsf{CT}_{\mathbf{x}}$. It proceeds as follows:

1. Sample $\Delta_1, \ldots, \Delta_q \subseteq S$, of size v each.
2. For each $i \in [q^*]$, invoke $\mathsf{RE.Sim}(C_i(x))$ to learn $G_{C_i}(\mathbf{x}, \hat{R}_i)$ for some $\hat{R}_i$ chosen by the simulator. Interpret

$$\hat{R}_i = \underset{a \in \Delta_i}{\oplus} R_a \quad \text{and} \quad G_{C_i, \Delta_i}(\mathbf{x}, R_1, \ldots, R_S) = G_{C_i}(\mathbf{x}, \hat{R}_i) = \big(P_1(\mathbf{Y}), \ldots, P_k(\mathbf{Y})\big).$$

3. Let $\mathsf{CT}_{\mathbf{x}} = \mathsf{LinFE.Sim}\big(\{G_{C_i, \Delta_i}, G_{C_i, \Delta_i}(\mathbf{x}, R_1, \ldots, R_S), \mathsf{SK}_i\}_{i \in [q^*]}\big)$ and output it.

The correctness of $\mathsf{Bdd.Sim}$ follows from the correctness of $\mathsf{RE.Sim}$ and $\mathsf{LinFE.Sim}$.

A last remaining technicality is that the most general version of our construction for FE for inner product modulo p is stateful. This is because a general adversary against LinFE may request keys that are linearly dependent modulo p

but linearly independent over the integers, thus learning new linear relations in the master secret. This forces the simulator (and hence the key generator) to maintain a state.

However, in our application, we can make do with a stateless variant, since all the queries will be linearly independent over $\mathbb{Z}_2$. To see this, note that in the above application of LinFE, each query is randomized by a unique random set Δ_i. Recall that by cover-freeness, the element $\underset{a \in \Delta_i}{\oplus} R_a$ must contain at least one fresh random element, say R^*, which is not contained by $\underset{j \neq i}{\cup} \Delta_j$. Stated a bit differently, if we consider the query vectors of size L, then cover-freeness implies that no query vector lies within the linear span of the remaining queries made by the adversary. For any query Q, there is at least one position $j \in [L]$ so that this position is nonzero in the L vector representing Q but zero for all other vectors. Hence the query vectors are linearly independent over $\mathbb{Z}_2$, for which case, our construction of Sect. 4.2 is stateless.

A   Definitions for Functional Encryption

We now recall the syntax of Functional Encryption, as defined by Boneh et al. [19], and their indistinguishability-based security definition.

Definition 6 [19]**.** *A functionality F defined over $(\mathcal{K}, \mathcal{Y})$ is a function $F : \mathcal{K} \times \mathcal{Y} \to \Sigma \cup \{\bot\}$, where $\mathcal{K}$ is a key space, $\mathcal{Y}$ is a message space and Σ is an output space, which does not contain the special symbol $\bot$.*

Definition 7. *A functional encryption (FE) scheme for a functionality F is a tuple $\mathcal{FE} = (\mathsf{Setup}, \mathsf{Keygen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ of algorithms with the following specifications:*

Setup(1^λ)**:** *Takes as input a security parameter 1^λ and outputs a master key pair $(\mathsf{mpk}, \mathsf{msk})$.*

Keygen(msk, K)**:** *Given the master secret key msk and a key (i.e., a function) $K \in \mathcal{K}$, this algorithm outputs a key sk_K.*

Encrypt(mpk, Y)**:** *On input of a message $Y \in \mathcal{Y}$ and the master public key mpk, this randomized algorithm outputs a ciphertext C.*

Decrypt$(\mathsf{mpk}, \mathsf{sk}_K, C)$**:** *Given the master public key mpk, a ciphertext C and a key sk_K, this algorithm outputs $v \in \Sigma \cup \{\bot\}$.*

We require that, for all $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$, all keys $K \in \mathcal{K}$ and all messages $Y \in \mathcal{Y}$, if $\mathsf{sk}_K \leftarrow \mathsf{Keygen}(\mathsf{msk}, K)$ and $C \leftarrow \mathsf{Encrypt}(\mathsf{mpk}, Y)$, with overwhelming probability, we have $\mathsf{Decrypt}(\mathsf{mpk}, \mathsf{sk}_K, C) = F(K, Y)$ whenever $F(K, Y) \neq \bot$.

In some cases, we will also give a state st as input to algorithm Keygen, so that a stateful authority may reply to key queries in a way that depends on the queries that have been made so far. In that situation, algorithm Keygen may additionally update state st.

INDISTINGUISHABILITY-BASED SECURITY. From a security standpoint, what we expect from a FE scheme is that, given $C \leftarrow$ Encrypt(mpk, Y), the only thing revealed by a secret key sk_K about the underlying Y is the function evaluation $F(K, Y)$. In the natural definition of indistinguishability-based security (see, e.g., [19]), one asks that no efficient adversary be able to differentiate encryptions of Y_0 and Y_1 without obtaining secret keys sk_K such that $F(K, Y_0) \neq F(K, Y_1)$.

A detailed definition of indistinguishability-based security (which initially comes from [19, Sect. 4]) is given in the full version of the paper. It captures *adaptive* security in that the adversary is allowed to choose the messages Y_0, Y_1 in the middle of the game, based on the information obtained so far. Abdalla *et al.* [2] considered a weaker notion, called *selective* security, where the adversary has to declare the messages Y_0, Y_1 before even seeing mpk. In this scenario, the adversary can receive the challenge ciphertext at the same time as the public key.

References

1. Abdalla, M., et al.: Searchable encryption revisited: consistency properties, relation to anonymous IBE, and extensions. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 205–222. Springer, Heidelberg (2005)
2. Abdalla, M., Bourse, F., De Caro, A., Pointcheval, D.: Simple functional encryption schemes for inner products. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 733–751. Springer, Heidelberg (2015)
3. Agrawal, S., Freeman, D.M., Vaikuntanathan, V.: Functional encryption for inner product predicates from learning with errors. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 21–40. Springer, Heidelberg (2011)
4. Agrawal, S., Libert, B., Stehlé, D.: Fully secure functional encryption for inner products, from standard assumptions. Cryptology ePrint Archive: report 2015/608
5. Agrawal, S., Gorbunov, S., Vaikuntanathan, V., Wee, H.: Functional encryption: new perspectives and lower bounds. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 500–518. Springer, Heidelberg (2013)
6. Agrawal, S., Rosen, A.: Online-offline functional encryption for bounded collusions. Cryptology ePrint Archive, Report 2016/361 (2016). http://eprint.iacr.org/
7. Alperin-Sheriff, J., Peikert, C.: Circular and KDM security for identity-based encryption. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 334–352. Springer, Heidelberg (2012)
8. Ananth, P., Brakerski, Z., Segev, G., Vaikuntanathan, V.: From selective to adaptive security in functional encryption. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9216, pp. 657–677. Springer, Heidelberg (2015)
9. Applebaum, B., Ishai, Y., Kushilevitz, E.: Computationally private randomizing polynomials and their applications. Comput. Complex. **15**(2), 115–162 (2006)

10. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation. In: Proceedings of STOC, pp. 1–10. ACM (1988)
11. Bernstein, D.J., Lange, T.: Computing small discrete logarithms faster. In: Galbraith, S., Nandi, M. (eds.) INDOCRYPT 2012. LNCS, vol. 7668, pp. 317–338. Springer, Heidelberg (2012)
12. Bishop, A., Jain, A., Kowalczyk, L.: Function-hiding inner product encryption. In: Iwata, T., et al. (eds.) ASIACRYPT 2015. LNCS, vol. 9452, pp. 470–491. Springer, Heidelberg (2015). doi:10.1007/978-3-662-48797-6_20
13. Boneh, D., Boyen, X.: Efficient selective-ID secure identity-based encryption without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
14. Boneh, D., Boyen, X.: Efficient selective identity-based encryption without random oracles. J. Cryptol. **24**(4), 659–693 (2011)
15. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
16. Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 506–522. Springer, Heidelberg (2004)
17. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. SIAM J. Comput. **32**(3), 586–615 (2003). (electronic)
18. Boneh, D., Gentry, C., Gorbunov, S., Halevi, S., Nikolaenko, V., Segev, G., Vaikuntanathan, V., Vinayagamurthy, D.: Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 533–556. Springer, Heidelberg (2014)
19. Boneh, D., Sahai, A., Waters, B.: Functional encryption: definitions and challenges. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 253–273. Springer, Heidelberg (2011)
20. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: On the classical hardness of learning with errors. In: Proceedings of STOC, pp. 575–584. ACM (2013)
21. Bresson, E., Catalano, D., Pointcheval, D.: A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications. In: Laih, C.-S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 37–54. Springer, Heidelberg (2003)
22. Camenisch, J.L., Shoup, V.: Practical verifiable encryption and decryption of discrete logarithms. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 126–144. Springer, Heidelberg (2003)
23. Cheon, J.H., Han, K., Lee, C., Ryu, H., Stehlé, D.: Cryptanalysis of the multilinear map over the integers. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 3–12. Springer, Heidelberg (2015)
24. Coron, J.-S., Lepoint, T., Tibouchi, M.: Practical multilinear maps over the integers. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 476–493. Springer, Heidelberg (2013)
25. Cramer, R., Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998)
26. Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002)

27. Damgård, I.B.: Towards practical public key systems secure against chosen ciphertext attacks. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 445–456. Springer, Heidelberg (1992)

28. De Caro, A., Iovino, V., Jain, A., O'Neill, A., Paneth, O., Persiano, G.: On the achievability of simulation-based security for functional encryption. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 519–535. Springer, Heidelberg (2013)

29. Dodis, Y., Fazio, N.: Public key trace and revoke scheme secure against adaptive chosen ciphertext attack. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 100–115. Springer, Heidelberg (2002)

30. Ducas, L., Durmus, A., Lepoint, T., Lyubashevsky, V.: Lattice signatures and bimodal Gaussians. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 40–56. Springer, Heidelberg (2013)

31. Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.: An algebraic framework for Diffie-Hellman assumptions. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 129–147. Springer, Heidelberg (2013)

32. Garg, S., Gentry, C., Halevi, S.: Candidate multilinear maps from ideal lattices. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 1–17. Springer, Heidelberg (2013)

33. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. In: Proceedings of FOCS, pp. 40–49 (2013)

34. Garg, S., Gentry, C., Halevi, S., Sahai, A., Waters, B.: Attribute-based encryption for circuits from multilinear maps. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 479–499. Springer, Heidelberg (2013)

35. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Proceedings of STOC, pp. 197–206. ACM (2008)

36. Goldwasser, S., Lewko, A., Wilson, D.A.: Bounded-collusion IBE from key homomorphism. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 564–581. Springer, Heidelberg (2012)

37. Goldwasser, S., Kalai, Y.T., Popa, R.A., Vaikuntanathan, V., Zeldovich, N.: How to run turing machines on encrypted data. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 536–553. Springer, Heidelberg (2013)

38. Goldwasser, S., Tauman Kalai, Y., Popa, R., Vaikuntanathan, V., Zeldovich, N.: Reusable garbled circuits and succinct functional encryption. In: Proceedings of STOC, pp. 555–564. ACM Press (2013)

39. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Functional encryption with bounded collusions via multi-party computation. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 162–179. Springer, Heidelberg (2012)

40. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Attribute-based encryption for circuits. In: Proceedings of STOC, pp. 545–554. ACM Press (2013)

41. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Predicate encryption for circuits from LWE. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9216, pp. 503–523. Springer, Heidelberg (2015)

42. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of ACM-CCS 2006, pp. 89–98. ACM Press (2006)

43. Hu, Y., Jia, H.: Cryptanalysis of GGH map. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9665, pp. 537–565. Springer, Heidelberg (2016). doi:10.1007/978-3-662-49890-3_21

44. Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 146–162. Springer, Heidelberg (2008)
45. Ling, S., Phan, D.H., Stehlé, D., Steinfeld, R.: Hardness of k-LWE and applications in traitor tracing. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 315–334. Springer, Heidelberg (2014)
46. Micciancio, D., Mol, P.: Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 465–484. Springer, Heidelberg (2011)
47. Micciancio, D., Peikert, C.: Trapdoors for lattices: simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012)
48. Nguyên, P.Q., Stern, J.: Merkle-Hellman revisited: a cryptanalysis of the Qu-Vanstone cryptosystem based on group factorizations. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 198–212. Springer, Heidelberg (1997)
49. O'Neill, A.: Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556 (2010). http://eprint.iacr.org/
50. O'Neill, A., Peikert, C., Waters, B.: Bi-deniable public-key encryption. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 525–542. Springer, Heidelberg (2011)
51. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)
52. Pollard, J.: Kangaroos, monopoly and discrete logarithms. J. Cryptol. **13**, 433–447 (2000)
53. Regev, O.: On lattices, learning with errors, random linear codes, cryptography. In: Proceedings of STOC, pp. 84–93. ACM (2005)
54. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM **56**(6), 34 (2009)
55. Sahai, A., Seyalioglu, H.: Worry-free encryption: functional encryption with public keys. In: Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010 (2010)
56. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
57. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
58. Waters, B.: Functional encryption for regular languages. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 218–235. Springer, Heidelberg (2012)