






# Combinatorially Homomorphic Encryption

Yuval Ishai<sup>(✉)</sup>, Eyal Kushnir, and Ron D. Rothblum

Technion, Haifa, Israel

{yuvali,eyal.kushnir,rothblum}@cs.technion.ac.il

**Abstract.** Homomorphic encryption enables public computation over encrypted data. In the past few decades, homomorphic encryption has become a staple of both the theory and practice of cryptography. Nevertheless, while there is a general loose understanding of what it means for a scheme to be homomorphic, to date there is no single unifying minimal definition that captures all schemes. In this work, we propose a new definition, which we refer to as *combinatorially homomorphic encryption*, which attempts to give a broad base that captures the intuitive meaning of homomorphic encryption.

Our notion relates the ability to accomplish some task when given a ciphertext, to accomplishing the same task without the ciphertext, in the context of *communication complexity*. Thus, we say that a scheme is combinatorially homomorphic if there exists a communication complexity problem  $f(x, y)$  (where  $x$  is Alice's input and  $y$  is Bob's input) which requires communication  $c$ , but can be solved with communication less than  $c$  when Alice is given in addition also an encryption  $E_k(y)$  of Bob's input (using Bob's key  $k$ ).

We show that this definition indeed captures pre-existing notions of homomorphic encryption and (suitable variants are) sufficiently strong to derive prior known implications of homomorphic encryption in a conceptually appealing way. These include constructions of (lossy) public-key encryption from homomorphic private-key encryption, as well as collision-resistant hash functions and private information retrieval schemes.

## 1 Introduction

Homomorphic encryption, originally proposed by Rivest, Adleman, and Demoullin [39], is one of the cornerstones of modern cryptography. Roughly speaking, an encryption scheme is homomorphic wrt to a function  $f$  if given an encryption of a message  $m$ , it is possible to generate an encryption of  $f(m)$ , without knowing the secret key. Homomorphic encryption is used extensively in cryptography, whether explicitly, or implicitly via homomorphisms offered by concrete schemes (e.g., based on factoring, discrete log, or lattices). Until 2009, the default interpretation of homomorphic encryption was for  $f$  to be a linear function; this is still a commonly used special case today both in theory and in practice. However, since then, we have seen the development of *fully* homomorphic encryption schemes [11, 18], which are homomorphic wrt to *all* functions  $f$ .

There are many different candidates for homomorphic encryption from the literature (Goldwasser-Micali [22], Benaloh [6], ElGamal [17], Paillier [34], Damgård-Jurik [15], Regev [38] and more) and many different interpretations and precise definitions for what exact form of homomorphism they achieve. However, all definitions that we are aware of (and are discussed in detail next) are either too strict, in the sense that they only capture a few of the candidates, or are too broad, in the sense that they do not draw a clear line between “trivial” and “nontrivial” homomorphism.

Thus, despite being a central notion in cryptography, there is no canonical definition of what it means for an encryption scheme to be homomorphic. The main goal of this work is to introduce such a broad notion (or rather several variants following one theme) that captures and abstracts the intuition underlying the concept of homomorphic encryption and may serve as a default “minimal” interpretation of what homomorphic encryption means.

Let  $(\text{Gen}, \text{Enc}, \text{Dec})$  be a (private-key or public-key) encryption scheme. We proceed to discuss several takes on the notion of homomorphic encryption, and what we find lacking in each.

**Ideal Homomorphism:** A very simple and strong definition of homomorphic encryption may require that a homomorphically evaluated ciphertext, generated by an evaluation of the function  $f$  on the ciphertext  $E_{pk}(m)$ , is distributed similarly<sup>1</sup> to  $E_{pk}(f(m))$ .

This notion is extremely strong (and useful) and is satisfied by a few number theoretic based schemes such as Goldwasser-Micali [22] and Benaloh [6] (ElGamal [17] and Paillier/Damgård-Jurik [15, 34] also offer some form of ideal homomorphism but suffer from caveats that are discussed below). Unfortunately, many other schemes, especially lattice-based ones, do not satisfy it. Moreover, this strong notion is an overkill for many applications.

**Algebraic Homomorphism:** (a.k.a. Linear Homomorphism or Additive Homomorphism) An algebraic perspective taken earlier on (and inspired by the number-theory based schemes available at the time), is to view the plaintext and ciphertext spaces as groups, so that the encryption function is a homomorphism from the former to the latter.<sup>2</sup> Thus, running the group operation on the ciphertexts has the effect of implementing the corresponding group operation on the plaintexts.

Unfortunately, this definition is quite restrictive. In particular, it does not capture homomorphisms that are non-linear such as [9, 20, 28] let alone fully-homomorphic schemes (e.g., [11, 18, 21]). ElGamal with plaintexts implemented as group elements is only homomorphic wrt the underlying cryptographic group, whereas ElGamal with plaintexts in the exponent only supports decryption of small plaintext values. Lattice-based encryption schemes

<sup>1</sup> Several variants of the definition are possible depending on whether the similarity should be perfect, statistical or computational, and also whether it should hold even given additional information such as  $E_{pk}(m)$ , or even given the corresponding secret-key. We ignore these subtleties here.

<sup>2</sup> Indeed, this is the source of the term homomorphic encryption.

such as Regev [38] only support a bounded number of operations that depend on the modulus-to-noise ratio.

**Functional Homomorphism:** A typical modern definition of (public-key) homomorphic encryption states that an encryption scheme  $(\text{Gen}, \text{Enc}, \text{Dec})$  is homomorphic wrt to a function  $f$ , or (more generally) a class  $\mathcal{F}$  of functions, if there exists a poly-time Eval algorithm such that  $\text{Dec}_{sk}(\text{Eval}_{pk}(\text{Enc}_{pk}(m), f)) = f(m)$  for all messages  $m$ , key-pairs  $(pk, sk)$ , and  $f \in \mathcal{F}$ . To avoid trivial solutions, the homomorphic evaluation algorithm is further assumed to be “compact.” This is typically defined to mean that the size of the generated ciphertext or the decryption circuit is smaller than the circuit size of  $f$ .<sup>3</sup> The precise notion of compactness varies both quantitatively (Should the size of the evaluated ciphertext be independent of the circuit? Is a poly-logarithmic or even sub-linear dependence allowed?) and qualitatively (Why circuits? How exactly is circuit complexity measured? What about redundancies in the representation?). In particular, it is unclear what a minimal notion of compactness that suffices for applications should be. Beyond the difficulty with formalizing the common notion of compactness, we point out several additional difficulties with existing definitions of functional homomorphism:

1. Usually, lattice-based schemes only satisfy an approximate notion of this definition as there is a noise associated with each ciphertext, and this noise grows as the homomorphic evaluation is performed, until a point in which the ciphertext is undecryptable.

This can sometimes be avoided by using a large modulus-to-noise ratio, but that is merely hiding the problem under the rug—we do think of the schemes as homomorphic even when the modulus-to-noise ratio is small, but the definition is not flexible enough to capture this.

2. Discrete-log based schemes such as ElGamal, over a cyclic plaintext group of order  $q$ , are often thought of as linearly homomorphic with addition in the group  $\mathbb{Z}_q$ . As briefly mentioned above though, one can only decrypt ciphertexts whose messages are polynomially small as decryption involves a discrete-log operation.

Despite this well-known fact, ElGamal is considered to be additively homomorphic but capturing it within the existing framework is quite messy.

3. Lastly, if one wishes to define homomorphic encryption in general, that is, not specifically wrt some function  $f$ , this approach becomes problematic. For example, simply assuming the existence of *some* function  $f$  such that

---

<sup>3</sup> If compactness is not required, then the homomorphic evaluation can be trivially delegated to the decryptor (e.g., by appending the description of the circuit the ciphertext). Nevertheless, some homomorphic schemes such as [41] or constructions based on garbled circuits [12, 19, 25, 27] are not compact but are circuit private, meaning that the ciphertext does not reveal the evaluated circuit. In this work, we focus on compact homomorphic encryption, which is meaningful even without circuit privacy.

the scheme is functionally homomorphic wrt  $f$  is not very meaningful if  $f$  is the identity function or a constant function. More generally, it is not entirely clear what non-triviality constraints  $f$  needs to satisfy for this notion to be meaningful or useful.

## 1.1 Combinatorially Homomorphic Encryption

Our main contribution is proposing a new definition for homomorphic encryption. Our goal in this definition is threefold: (1) we wish to find a notion that is consistent and truly formalizes the intuitive meaning of homomorphic encryption, drawing precise lines between “trivial” and “nontrivial” homomorphism; (2) for the definition to be sufficiently broad to capture all schemes that are currently thought of as homomorphic (including both number-theory and lattice-based schemes) and (3) for the definition to be sufficiently strong to preserve the known implications of existing notions of homomorphic encryption such as public-key encryption (PKE), collision-resistant hashing (CRH) and private information retrieval (PIR). We believe that positioning homomorphic encryption as a true cryptographic primitive, similarly to “one-way function” or “public-key encryption”, will facilitate a systematic study of its relation with other cryptographic primitives.

We call this new framework *combinatorially homomorphic encryption*, of which we describe several variants. The first variant refers to *communication complexity* [43], which we briefly review. Recall that in *distributional communication complexity* there are two parties, Alice and Bob, who respectively get inputs  $x$  and  $y$ , drawn from some joint distribution. Their goal is to compute some function  $f(x, y)$  while minimizing the number of bits exchanged between them to the extent possible. In our most basic definition (which is sufficient for most of the goals listed above), we focus specifically on *one-way* communication complexity—that is when communication is only allowed from Alice to Bob (and not in the other direction). In other words, the minimal number of bits that Alice needs to send to Bob so that he can compute  $f(x, y)$ . See [29, 36] for a detailed introduction to communication complexity.

The first instantiation of our framework for homomorphic encryption takes the following operational perspective. We say that a scheme is *communication-complexity (CC) homomorphic* if there exists some one-way communication complexity problem  $f$ , which requires communication  $c$ , such that if Alice is given, in addition to  $x$ , a ciphertext  $\text{Enc}_k(y)$  of Bob’s input using Bob’s key  $k$ , then the communication problem can be solved using less than  $c$  bits (and where Alice and Bob both run in polynomial-time). Note that while it is possible to talk about CC-homomorphic encryption with respect to a specific communication complexity problem, our main definition refers to the *existence* of a communication complexity problem for which the notion is non-trivial.

**Definition 1 (Informally Stated, see Sect. 3).** *We say that an encryption scheme (Gen, Enc, Dec) is CC homomorphic if there exists a communication complexity problem  $f$  which requires communication  $c$ , but there exists a polynomial-*

time one-way protocol for solving the problem  $f'((x, \text{Enc}_k(y)), (y, k))$ , defined as  $f'((x, \text{Enc}_k(y)), (y, k)) = f(x, y)$ , with communication less than  $c$ .

The definition can be adapted to the public-key setting in the natural way (i.e.,  $y$  is encrypted under the public key and Bob gets the corresponding private key).

CC homomorphic encryption captures the basic intuitive understanding that homomorphic encryption should enable *useful* computation on encrypted data. Here, Alice can perform such a computation in a way that helps Bob derive the output more efficiently than if Alice had not been given the ciphertext.

We also consider generalizations of this notion in two ways. First, we consider an interactive variant (presented in the full version), in which the homomorphic communication game is allowed to be interactive and the communication complexity lower bound holds in the interactive setting (which is the standard model for communication complexity). Motivated by applications described below, we also consider comparing the “homomorphic communication complexity” to other combinatorial measures of the function  $f$  such as its VC dimension.<sup>4</sup> Lastly, while our basic definition considers distributional communication complexity over efficiently sampleable *product* distributions, it suffices for our results that the conditional marginal distributions are efficiently sampleable.

*Existing Schemes in the Lens of Combinatorially Homomorphic Encryption.* To see that CC homomorphic encryption indeed captures existing schemes, consider an encryption scheme that is linearly homomorphic mod 2, in the standard functional sense. To see that such a scheme is combinatorially homomorphic, consider the inner product communication complexity game in which Alice and Bob get as input random vectors  $x, y \in \{0, 1\}^n$  and Bob’s goal is to compute their inner product  $\langle x, y \rangle = \bigoplus_{i \in [n]} x_i y_i$ . It is well-known that this task requires communication complexity  $\Omega(n)$  (in fact, in the one-way version, this follows directly from the leftover hash lemma). However, if Alice is given in addition to  $x$ , also a bit-by-bit encryption  $\text{Enc}_k(y_1), \dots, \text{Enc}_k(y_n)$  of Bob’s input, then using the linear homomorphism she can compute an encryption of  $\langle x, y \rangle$  and send it to Bob, who can decrypt and retrieve the result. The compactness property of functional homomorphic encryption guarantees that the communication in this new protocol is smaller than the  $\Omega(n)$  lower bound that holds when Alice is not given the encryption of Bob’s input.

The above idea can be generalized to linear homomorphisms over any group, as stated in the following theorem. A simple unifying explanation is that traditional homomorphic schemes from the literature imply PIR, which can be thought of as being CC-homomorphic with respect to the “index” function. In particular, it shows that Goldwasser-Micali [22], Benaloh [6] and Regev [38] fall within our framework.

**Theorem 1 (Informally Stated, see the full version).** *Any linearly homomorphic private-key encryption scheme is combinatorially homomorphic.*

<sup>4</sup> More precisely, we consider the VC dimension of the function family  $\{f_x : \{0, 1\}^n \rightarrow \{0, 1\}\}_x$ , where  $f_x(y) = f(x, y)$ .

To illustrate a concrete instantiation, we show a simple private-key scheme based on Learning with Errors (LWE) that satisfies our definition. The secret key is a random vector  $\mathbf{s} \leftarrow \mathbb{Z}_q^\lambda$ . To encrypt a bit  $b \in \{0, 1\}$ , sample a random  $\mathbf{a} \leftarrow \mathbb{Z}_q^\lambda$  and output  $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e + \lfloor q/2 \rfloor \cdot b)$  as the ciphertext, where  $e \in \mathbb{Z}_q$  comes from a  $B$ -bounded noise distribution. The security of this private-key scheme follows almost tautologically from decisional LWE.

Now consider the communication complexity game in which Alice and Bob get as their respective inputs  $x, y \in \{0, 1\}^n$  and their goal is to compute the inner product. As mentioned above, it is well known that this problem requires communication complexity  $\Omega(n)$ . Suppose however that Alice is given a bit-by-bit encryption of Bob's input. Namely, ciphertexts  $c_1, \dots, c_n$  such that  $c_i = (\mathbf{a}_i, \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i + \lfloor q/2 \rfloor \cdot y_i)$ . Alice can now compute a new ciphertext  $(\mathbf{a}', \sigma')$ , where  $\mathbf{a}' = \sum_i x_i \cdot \mathbf{a}_i$  and  $\sigma' = \sum_i x_i \cdot (\langle \mathbf{a}_i, \mathbf{s} \rangle + e_i + \lfloor q/2 \rfloor \cdot y_i) = \langle \mathbf{a}', \mathbf{s} \rangle + \sum_i x_i e_i + \lfloor q/2 \rfloor \cdot \langle x, y \rangle$  (and all arithmetic is mod  $q$ ). Alice sends this ciphertext to Bob who computes  $\sigma' - \langle \mathbf{a}', \mathbf{s} \rangle = \sum_i x_i e_i + \lfloor q/2 \rfloor \cdot \langle x, y \rangle$ . As long as  $\sum_i x_i e_i < q/4$  (which holds if  $B \cdot n < q/4$ ), then Bob can now correctly round and obtain  $\langle x, y \rangle$ . If the communication in this game (which is  $(\lambda + 1) \cdot \log(q)$ ) is smaller than the communication complexity lower bound of  $\Omega(n)$ , then this basic private-key scheme is CC homomorphic.<sup>5</sup>

Jumping ahead, one of our main applications is a construction of *public-key encryption* from any CC homomorphic *private-key* encryption (which extends the [40] construction of public-key encryption from linearly homomorphic encryption). Thus, the above construction yields a public-key encryption scheme from LWE which, we believe, cleanly abstracts Regev's [38] celebrated public-key scheme. Furthermore, our work is the first one to offer a qualitative notion of homomorphism, where each choice of parameters (including secret distribution and noise distribution) can be classified as either being combinatorially homomorphic or not.

Note that the definition of CC homomorphic encryption is sufficiently flexible to allow for variations of linear homomorphisms, and even for non-linear homomorphisms, that may be difficult to capture otherwise. All one needs to do is to adapt the communication complexity game to capture the specific functionality that is offered by the scheme and show the corresponding communication complexity lower bound (which is usually not difficult).

Consider, for example, the ElGamal cryptosystem [17] with plaintexts in the exponent, which is widely considered to be homomorphic, yet is not captured by the standard linearly homomorphic encryption definition (since decryption involves a discrete-log operation). The scheme works over a cyclic group  $\mathbb{G}$  of order  $q$  with generator  $g$ . The secret key is a random  $s \leftarrow \mathbb{Z}_q$ . To encrypt a bit  $b \in \{0, 1\}$ , sample a random  $r \leftarrow \mathbb{Z}_q$  and output  $(g^r, g^{s \cdot r + b})$ . To decrypt a ciphertext  $(c_0, c_1)$ , compute  $z = c_1 \cdot c_0^{-s}$  and output 0 if  $z = 1$  and 1 otherwise.

---

<sup>5</sup> The homomorphic private-key to public-key transformation of Rothblum [40] can also be viewed as morally giving an abstraction of Regev's scheme, but the actual formal definition of homomorphic encryption used in [40] is not technically achieved by the above private-key scheme.

The security of this private-key scheme follows from the decisional Diffie-Hellman assumption.

To show that the above encryption scheme is CC-homomorphic we will use the well-known Disjointness communication complexity problem, where Alice and Bob are given sets  $x, y \subseteq [n]$  respectively, and need to determine whether their sets are disjoint. Babai *et al.* [4] showed that the disjointness problem has communication complexity  $\Omega(\sqrt{n})$  (on a specific product distribution<sup>6</sup>). Suppose however that Alice is given bit-by-bit encryptions  $c_1, \dots, c_n$  of Bob's input (the input sets  $x$  and  $y$  can be represented as indicator vectors so that  $c_i = (g^{r_i}, g^{r_i \cdot s + y_i})$ ). Alice can then compute  $(\prod_{i \in \mathcal{I}(x)} g^{r_i}, \prod_{i \in \mathcal{I}(x)} g^{r_i \cdot s + y_i}) = (g^{r'}, g^{r' \cdot s + \sum_{i \in \mathcal{I}(x)} y_i})$ , where  $\mathcal{I}(x) = \{i : x_i = 1\}$ . Alice can send this ciphertext to Bob who can compute  $z = g^{r' \cdot s + \sum_{i \in \mathcal{I}(x)} y_i} \cdot (g^{r'})^{-s} = g^{\sum_{i \in \mathcal{I}(x)} y_i}$ . It holds that  $z = 1$  if and only if the sets are disjoint. Therefore, if the communication complexity of this protocol (which is  $2 \log(q)$ ) is smaller than the communication complexity lower bound (which is  $\sqrt{n}$ ), then the private-key scheme is CC-homomorphic.

The above idea can be generalized to capture any encryption scheme that is homomorphic with respect to the OR operation, as stated in the following theorem.

**Theorem 2 (Informally Stated, see the full version).** *Any OR-homomorphic private-key encryption scheme is combinatorially homomorphic.*

We also show a specific instantiation of our scheme using low-noise LPN (i.e., when the absolute noise is roughly  $\log^2(\lambda)$ ). Using our framework in combination with the applications listed below, we can re-derive recent results on LPN (due to [7, 10]) in a way that we find to be conceptually simpler.

*Applications.* As our main technical results, we show that suitable variations of our basic notion of combinatorially homomorphic encryption suffice to derive some of the key applications that are known from (say) standard linearly homomorphic encryption.

Our first main result shows how to transform any combinatorially homomorphic *private-key* encryption into a public-key one. This generalizes the work of Rothblum [40], who gave such a transformation for linearly homomorphic private-key encryption. As a matter of fact, we obtain the stronger notion of *lossy* public-key encryption [5, 35] (which is equivalent to semi-honest two-message statistical oblivious transfer [23]).

**Theorem 3 (Informally Stated, see Theorem 8).** *If there exists a combinatorially homomorphic private-key encryption scheme then there exists a lossy public-key encryption scheme.*

<sup>6</sup> In fact, Razborov [37] showed an input distribution on which the communication complexity of disjointness is  $\Omega(n)$ . However, since this input distribution is not a *product* distribution, using involves slightly more involved techniques (see further discussion in Sect. 3.1).

We remark that the security property required from the private-key scheme is very mild (and in particular is weaker than CPA security). Specifically, we merely need a weak notion of “entropic security” (see Definition 7) which, loosely speaking, requires that the distributions  $(y, \text{Enc}_k(y))$  and  $(y, \text{Enc}_k(y'))$  are computationally indistinguishable, where  $y, y'$  are independent samples drawn from Bob’s input distribution in the communication game.

As it is instructive to understanding the power of CC homomorphic encryption, we briefly sketch a simplified proof of Theorem 3 next. The public key of the scheme is  $(y, \text{Enc}_k(y))$ , where  $y$  is a random input for Bob in the communication game, and  $k$  is the private key of the private-key scheme. To encrypt a bit  $b$ , a random input  $x$  for Alice is sampled, and the ciphertext is Alice’s message in the “homomorphic” protocol  $m_A$ , as well as  $f(x, y) \oplus b$ . To decrypt, we run Bob on input  $((y, k), m_A)$  to compute  $f(x, y)$ , and then we can retrieve the message bit  $b$ . Correctness follows from the correctness of the homomorphic protocol. As for security, using the entropic security of the private-key scheme, we can switch the public key  $(y, \text{Enc}_k(y))$  to the lossy public key  $(y, \text{Enc}_k(y'))$ . Thus, the adversary’s goal now is essentially to compute  $f(x, y)$  given  $(y, \text{Enc}_k(y'))$  and  $m_A$ .

Assume that this is possible. Then we can derive a more efficient communication complexity protocol for computing  $f$  in the standard setting, in which Alice gets only  $x$  and Bob gets only  $y$ . Alice and Bob sample a key  $k$  and a ciphertext  $\text{Enc}_k(y')$  using shared randomness.<sup>7</sup> Then, Alice generates a message  $m_A$  from the homomorphic protocol and sends it to Bob, who can then run the adversary on input  $((y, c), m_A)$  to compute  $f(x, y)$ . Since we required that Alice’s message in the homomorphic protocol is shorter than the communication complexity of  $f$ , we derive a contradiction. Note that this argument immediately gives the stronger notion of *lossy* encryption.

This basic result can be generalized to interactive combinatorially homomorphic encryption in which case we derive a key agreement protocol (which can be thought of as an interactive analog of public-key encryption).

**Theorem 4 (Informally Stated, see the full version).** *If there exists an interactive combinatorially homomorphic encryption scheme then there exists a key agreement protocol.*

Ishai, Kushilevitz and Ostrovsky [26] showed how to construct a *collision-resistant hash function* (CRH) from any linearly homomorphic encryption scheme. Recall that a CRH is a collection of shrinking hash functions so that no polynomial-time adversary can find a collision, given the description of a random function from the collection. We generalize the [26] result and construct CRH from any CC homomorphic encryption.

**Theorem 5 (Informally Stated, see Theorem 9).** *If there exists a combinatorially homomorphic encryption scheme (satisfying a mild non-triviality constraint) then there exists a collision-resistant hash function.*

<sup>7</sup> As usual in distributional communication complexity, this shared randomness can be eliminated by non-uniformly fixing the best choice.



(The mild non-triviality constraint that we require is that the communication complexity problem is defined wrt a function  $f$  such that the function family  $\{f_y : \{0, 1\}^n \rightarrow \{0, 1\}\}_y$ , where  $f_y(x) = f(x, y)$ , is a universal hash function family).

As in [26], for this application, we do not need the decryption algorithm to be efficient, and a more general notion of “CC homomorphic commitment” (in which Bob can be inefficient in the communication game) suffices.

Next, we revisit the Kushilevitz-Ostrovsky [30] construction of *private information retrieval* (PIR) scheme from a linearly homomorphic encryption scheme.<sup>8</sup> Recall that a PIR scheme is a two-party protocol between a server, which is given a database  $x \in \{0, 1\}^n$ , and a client who is given as input an index  $i \in [n]$ . The goal is for the client to reconstruct  $x_i$  whereas the index  $i$  is computationally hidden from the server (both parties are assumed to be polynomial-time). We say that the PIR scheme is non-trivial if the communication complexity is less than  $n$ .<sup>9</sup>

We generalize the [30] construction and derive PIR from combinatorially homomorphic encryption. For this result, we need the communication in the homomorphic variant of the communication game to be shorter than before. Specifically, rather than beating the communication complexity lower bound for the game, it should beat its *VC dimension*. We refer to schemes satisfying this (intuitively stronger) notion as *VC homomorphic*.

**Theorem 6 (Informally Stated, see the full version).** *Assume that there exists a VC homomorphic encryption scheme then there exists a non-trivial PIR scheme.*

*Applications from Learning Parity with Noise.* As noted above, we can capture a low noise variant of LPN (specifically with an absolute noise level of roughly  $\log^2(n)$ ) in our framework, via a simple construction. Using Theorem 5, we can use LPN with this noise level to obtain CRH, thereby giving a conceptually simple derivation of recent results [10, 44]. Similarly, using Theorem 3 we get a simple construction of semi-honest 2-message statistical OT from LPN. This can be viewed as an abstraction of a recent result of Bitansky and Freizeit [7]. We emphasize though that [7] use the semi-honest construction only as a stepping stone towards a construction that achieves security against malicious receivers (but additionally requires a Nisan-Wigderson style derandomization assumption).

---

<sup>8</sup> The [30] construction is based on the Quadratic Residuosity assumption, but is easy to generalize to compact linearly homomorphic encryption (for a suitable notion of compactness), see [31, 42].

<sup>9</sup> While a PIR scheme with communication, say,  $n - 1$  does not seem directly useful, it is sufficient for deriving some important consequences of PIR such as CRH [26], oblivious transfer [14], lossy encryption [23] and SZK hardness [32].

## 1.2 Related Work

As previously mentioned, Rothblum [40] showed that any linearly homomorphic encryption that satisfies a mild compactness property can be used to construct a public-key encryption scheme. His proof relies on the Leftover Hash Lemma and can be streamlined using our framework (see discussion in Sect. 1.1).

Alamati *et al.* [1, 2] study the possibility of constructing Cryptomania primitives (such as CRH and PKE) based on Minicrypt primitives that are equipped with certain algebraic structures. Their work is limited to primitives with group homomorphism over the input or output spaces. In particular, like [40], their work does not consider non-linear homomorphisms.

Bogdanov and Lee [8] study the limits of security for homomorphic encryption. Along the way, they introduce a notion of sensitivity for homomorphically evaluated functions. While this notion suffices for their applications, it does not seem to be a minimal notion of non-triviality for functional homomorphisms.

Cohen and Naor [13] study a different connection between communication complexity and cryptography, and in particular, show that the existence of non-trivial communication complexity protocols in which the inputs are drawn from efficiently sampleable distributions imply cryptographic primitives (such as distribution collision-resistant hash functions).

## 2 Preliminaries

For a distribution  $D$ , we denote by  $x \leftarrow D$  the process of sampling from  $D$ . For any joint distribution  $(X, Y)$  we will denote by  $x \leftarrow \text{Proj}_1(X, Y)$  or  $y \leftarrow \text{Proj}_2(X, Y)$  sampling from  $(X, Y)$  and keeping only the first or the second element of the pair, respectively. A function  $\mu : \mathbb{N} \rightarrow [0, 1]$  is *negligible* if for every polynomial  $p$  and sufficiently large  $\lambda$  it holds that  $\mu(\lambda) \leq 1/p(\lambda)$ . All logarithms considered in this paper are in base 2.

**Definition 2 (Statistical Distance).** *Let  $X$  and  $Y$  be two distributions over a finite domain  $U$ . The statistical distance between  $X$  and  $Y$  is defined as follows.*

$$\text{SD}(X, Y) = \max_{f: U \rightarrow \{0,1\}} \left| \Pr[f(X) = 1] - \Pr[f(Y) = 1] \right|.$$

*If  $\text{SD}(X, Y) \leq \epsilon$  we say that  $X$  is  $\epsilon$ -close to  $Y$ .*

Next, we define computational indistinguishability, which can be thought of as a computational analog of the statistical distance.

**Definition 3 (Computational Indistinguishability).** *We say that two distribution ensembles  $X = (X_\lambda)_{\lambda \in \mathbb{N}}$  and  $Y = (Y_\lambda)_{\lambda \in \mathbb{N}}$  are computationally indistinguishable, and denote it by  $X \approx_c Y$ , if for every probabilistic polynomial-size distinguisher  $\mathcal{D}$  there exists a negligible function  $\mu$  such that for every  $\lambda \in \mathbb{N}$ ,*

$$\left| \Pr[\mathcal{D}(X_\lambda) = 1] - \Pr[\mathcal{D}(Y_\lambda) = 1] \right| \leq \mu(\lambda).$$

## 2.1 Communication Complexity

Communication complexity (CC), introduced by Yao [43], provides a mathematical model for the study of communication between two or more parties. It has proven to be a powerful tool in a surprising variety of fields such as circuit complexity, streaming, and quantum computing. We refer to the books by Kushilevitz and Nisan [29] and by Rao and Yehudayoff [36] for a comprehensive introduction. We now turn to recall several CC-related definitions that will be used in this paper.

Let  $f$  be a 2-argument function. Consider the setting of two communicating parties, Alice and Bob, who are given inputs  $x$  and  $y$  respectively, and wish to *cooperatively* compute the value of  $f(x, y)$  (without loss of generality we will require that only Bob outputs this value). The communication between them is conducted according to some fixed deterministic protocol  $\pi$ . The output of the protocol (i.e., Bob's output) on inputs  $x$  and  $y$  is denoted by  $\pi(x, y)$ .

*Distributional Communication Complexity.* We allow the protocol to err with a small probability on some input distribution. Namely,

**Definition 4 (Protocol Correctness).** *Given a function  $f : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$  and a joint input distribution  $(X, Y)$ , we say that a deterministic protocol  $\pi$  computes  $f$  with error  $\epsilon$  on  $(X, Y)$  if*

$$\Pr \left[ \pi(x, y) \neq f(x, y) : (x, y) \leftarrow (X, Y) \right] \leq \epsilon.$$

*Interchangeably, we can say that the protocol  $\pi$  computes  $f$  with correctness  $1 - \epsilon$  on  $(X, Y)$ .*

The communication complexity of a protocol  $\pi$  on inputs  $x$  and  $y$  is defined to be the number of bits exchanged by the parties while running the protocol on these inputs. The length of a protocol  $\pi$  on input distribution  $(X, Y)$ , denoted by  $\text{CC}[\pi, (X, Y)]$ , is defined to be the maximal communication complexity of the protocol on any input in the support of the distribution (notice that this measure is well-defined since these sets are finite).

The  $\epsilon$ -error distributional communication complexity of  $f$  on  $(X, Y)$  is the minimal length of any deterministic protocol computing  $f$  with error  $\epsilon$  with respect to  $(X, Y)$ . That is,

**Definition 5 (Distributional Communication Complexity).** *Given a function  $f$  and a joint input distribution  $(X, Y)$  we define the  $\epsilon$ -error  $(X, Y)$ -distributional communication complexity of  $f$  as follows.*

$$\mathcal{D}^{A \leftrightarrow B}(f, (X, Y), \epsilon) := \min_{\substack{\pi \text{ computes } f \\ \text{with error } \epsilon \\ \text{on } (X, Y)}} \text{CC}[\pi, (X, Y)].$$

The *one-way  $\epsilon$ -error  $(X, Y)$ -distributional communication complexity* of  $f$ , denoted by  $\mathcal{D}^{A \rightarrow B}(f, (X, Y), \epsilon)$ , is defined similarly but limited to one-round protocols that consist of only one message - from Alice to Bob.

*Discrepancy.* The discrepancy method is a common technique for proving lower bounds on distributional communication complexity. We now define the discrepancy of a function with respect to an input distribution.

**Definition 6 (Discrepancy).** *Given a function  $f : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$  and a joint input distribution  $(X, Y)$  we define the discrepancy of  $f$  on a rectangle  $R = S \times T \subseteq (X, Y)$ , denoted here by  $\text{Disc}(f, (X, Y); R)$ , as follows.*

$$\text{Disc}(f, (X, Y); R) := \left| \Pr \left[ (x, y) \in R \wedge f(x, y) = 1 \right] - \Pr \left[ (x, y) \in R \wedge f(x, y) = 0 \right] \right|,$$

where  $(x, y) \leftarrow (X, Y)$ . The discrepancy of  $f$  on  $(X, Y)$  is defined as

$$\text{Disc}(f, (X, Y)) := \max_R \text{Disc}(f, (X, Y); R).$$

A well-known theorem (see, e.g., [36, Theorem 5.2]) shows that the discrepancy can be used to lower bound distributional communication complexity.

**Theorem 7.** *For any function  $f : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$ , a joint input distribution  $(X, Y)$  and an error rate  $\epsilon \in (0, \frac{1}{2})$  we have that*

$$\mathcal{D}^{A \rightarrow B}(f, (X, Y), \epsilon) \geq \log \left( \frac{1 - 2\epsilon}{\text{Disc}(f, (X, Y))} \right)$$

## 2.2 Encryption

In this subsection, we describe the various notions of encryption that will be used throughout this work.

**Definition 7 ( $\mathcal{M}$ -Entropic Secure Private-Key Encryption).** *Let  $\mathcal{M} = (\mathcal{M}_\lambda)_{\lambda \in \mathbb{N}}$  be a message space. An  $\mathcal{M}$ -entropic secure private-key encryption scheme  $\mathcal{E} = (\text{Gen}, \text{Enc}, \text{Dec})$ , with correctness error  $\epsilon = \epsilon(\lambda)$ , is a triplet of probabilistic polynomial-time algorithms with the following syntax.*

- **Key generation.** *Given a security parameter  $1^\lambda$ , the algorithm  $\text{Gen}$  outputs a key  $k$ .*
- **Encryption.** *Given a message  $m \in \mathcal{M}_\lambda$  and a key  $k$ , the algorithm  $\text{Enc}$  outputs a ciphertext  $c$ .*
- **Decryption.** *Given a ciphertext  $c$  and a key  $k$ , the algorithm  $\text{Dec}$  outputs a message  $m$ .*

We require  $\mathcal{E}$  to satisfy the following properties.

- **Correctness.** *For any  $\lambda \in \mathbb{N}$  and message  $m \in \mathcal{M}_\lambda$  it holds that  $\Pr [\text{Dec}_k(c) = m] \geq 1 - \epsilon(\lambda)$ , where  $k \leftarrow \text{Gen}(1^\lambda)$  and  $c \leftarrow \text{Enc}_k(m)$ .*
- **$\mathcal{M}$ -entropic security.**  *$(m, \text{Enc}_k(m))_{\lambda \in \mathbb{N}} \approx_c (m, \text{Enc}_k(m'))_{\lambda \in \mathbb{N}}$ , where  $m$  and  $m'$  are two independent messages sampled from  $\mathcal{M}$ .*

We remark that the notion of entropic security defined above is morally weaker than notions such as CPA security since (1) the adversary is not given access to an encryption oracle and (2) security needs to hold only wrt messages arising from the given distribution (rather than “worst-case” messages).

**Definition 8 (CPA-Secure Private-Key Encryption).** A chosen-plaintext attack (CPA) secure private-key encryption scheme  $\mathcal{E} = (\text{Gen}, \text{Enc}, \text{Dec})$  with message length  $\ell = \ell(\lambda)$  and correctness error  $\epsilon = \epsilon(\lambda)$ , is defined similarly to Definition 7 but the entropic security requirement is replaced with the following:

- **CPA Security.** Consider the following security game.
  1. The challenger samples a key  $k \leftarrow \text{Gen}(1^\lambda)$ .
  2. The adversary chooses a message  $m$  of length  $\ell(\lambda)$  and receives  $\text{Enc}_k(m)$  from the challenger. This step is repeated for a polynomial number of times.
  3. The adversary chooses two challenge message  $m_0, m_1$  of length  $\ell(\lambda)$  and receives from the challenger  $\text{Enc}_k(m_b)$ .
  4. The adversary outputs a bit  $b' \in \{0, 1\}$ .

For any probabilistic polynomial-size adversary  $\mathcal{A}$ , we denote by  $\text{CPA}_{\mathcal{A}}^b(1^\lambda)$  the output of  $\mathcal{A}$  in the game above, and we require that there exists a negligible function  $\mu$  such that for any  $\lambda \in \mathbb{N}$ ,

$$|\Pr[\text{CPA}_{\mathcal{A}}^0(1^\lambda) = 1] - \Pr[\text{CPA}_{\mathcal{A}}^1(1^\lambda) = 1]| \leq \mu(\lambda).$$

We will next define a variant of lossy encryption [5, 35], which is equivalent to a 2-message (semi-honest) statistical OT [35].

**Definition 9 (Lossy Encryption).** Let  $\nu = \nu(\lambda)$  and  $\epsilon = \epsilon(\lambda)$ . A  $\nu$ -lossy bit-encryption scheme  $\mathcal{E} = (\text{Gen}, \text{Enc}, \text{Dec}, \text{LossyGen})$  with correctness error  $\epsilon$ , is a quadruple of polynomial-time algorithms with the following syntax,

- **Key generation.** Given a security parameter  $1^\lambda$ , the algorithm  $\text{Gen}$  outputs a secret key  $sk$  and a public key  $pk$ .
- **Encryption.** Given a bit  $b$  and a public key  $pk$ , the algorithm  $\text{Enc}$  outputs a ciphertext  $c$ .
- **Decryption.** Given a ciphertext  $c$  and a secret key  $sk$ , the algorithm  $\text{Dec}$  outputs a bit  $b$ .
- **Lossy key generation.** Given a security parameter  $1^\lambda$ , the algorithm  $\text{LossyGen}$  outputs a lossy key  $lk$ .

We require  $\mathcal{E}$  to satisfy the following properties.

- **Correctness.** For any  $\lambda \in \mathbb{N}$  and bit  $b$  it holds that  $\Pr[\text{Dec}_{sk}(c) = b] \geq 1 - \epsilon(\lambda)$ , where  $(sk, pk) \leftarrow \text{Gen}(1^\lambda)$  and  $c \leftarrow \text{Enc}_{pk}(b)$ .
- **Key indistinguishability.**  $(\text{Proj}_2(\text{Gen}(1^\lambda)))_{\lambda \in \mathbb{N}} \approx_c (\text{LossyGen}(1^\lambda))_{\lambda \in \mathbb{N}}$ .
- **Lossiness of lossy keys.** For any  $\lambda \in \mathbb{N}$ , we have that  $(lk, \text{Enc}_{lk}(0))$  is  $\nu(\lambda)$ -close in statistical distance to  $(lk, \text{Enc}_{lk}(1))$ , where  $lk \leftarrow \text{LossyGen}(1^\lambda)$ .

If not otherwise specified, by default, we take the parameters  $\nu$  and  $\epsilon$  to be negligible in parameter  $\lambda$ . One can also consider relaxed notions of lossy encryption, where either the correctness error is high—namely,  $\epsilon(\lambda) = \frac{1}{2} - \frac{1}{p(\lambda)}$ , for some polynomial  $p$ —or the statistical distance between encryptions under a lossy key is large—namely,  $\nu(\lambda) = 1 - \frac{1}{p(\lambda)}$ , for some polynomial  $p$ . Next, we will show that both variants are equivalent to the standard definition. We note however that if both the correctness *and* lossiness are close to  $1/2$ , then amplification is not known (see [16, 24] for further discussion and relation to the circuit polarization problem).

*Claim (Weak-Correctness Lossy Encryption implies Lossy Encryption).* Assume there exists a lossy encryption scheme with correctness error  $\frac{1}{2} - \frac{1}{p(\lambda)}$ , for some polynomial  $p$ , then there exists a lossy encryption scheme (Definition 9).

*Claim (Weak-Lossiness Lossy Encryption implies Lossy Encryption).* Assume there exists a  $(1 - \frac{1}{p(\lambda)})$ -lossy encryption scheme, for some polynomial  $p$ , then there exists a lossy encryption scheme (Definition 9).

The proofs of Sect. 2.2 and Sect. 2.2 are given in the full version.

### 2.3 Collision Resistant Hash Function

**Definition 10 (Collision Resistant Hash Function).** A collision resistant function with input length  $\ell(n)$  and output length  $\ell'(n) < \ell(n)$  is defined by a pair of algorithms (Gen, Eval) with the following syntax,

- **Key generation.** Given  $1^\lambda$  the probabilistic polynomial-time algorithm Gen outputs an index  $s$ .
- **Evaluation.** Given index  $s$  and input  $x$  of length  $\ell(\lambda)$ , the polynomial-time algorithm Eval outputs  $y \in \{0, 1\}^{\ell'(\lambda)}$ .

For any  $\lambda \in \mathbb{N}$ ,  $s \leftarrow \text{Gen}(1^\lambda)$  and  $x \in \{0, 1\}^{\ell(\lambda)}$  we define  $h_s(x) := \text{Eval}(s, x)$ .

We require the scheme to satisfy the following collision resistance property.

- **Collision resistance.** for every probabilistic polynomial-size adversary  $\mathcal{A}$  there exists a negligible function  $\mu$  such that for any  $\lambda \in \mathbb{N}$ ,

$$\Pr \left[ x \neq x' \wedge h_s(x) = h_s(x') : \begin{array}{l} s \leftarrow \text{Gen}(1^\lambda), \\ (x, x') \leftarrow \mathcal{A}(s) \end{array} \right] \leq \mu(\lambda).$$

## 3 Combinatorially Homomorphic Encryption

First, we define an extension of a function ensemble and an input distribution ensemble with respect to a private key encryption scheme. These will be used throughout the following sections.

Let  $f$  be an ensemble of 2-argument functions. Let  $(X, Y)$  be an ensemble of input distributions, where  $X = (X_\lambda)_{\lambda \in \mathbb{N}}$  and  $Y = (Y_\lambda)_{\lambda \in \mathbb{N}}$ . Let  $\mathcal{E} =$

(Gen, Enc, Dec) be a private-key encryption scheme (see Definition 7). We extend  $f$  and  $(X, Y)$  by defining for every  $\lambda \in \mathbb{N}$ ,

$$\text{Ext}_{\mathcal{E}}(X_{\lambda}, Y_{\lambda}) := \left\{ \begin{array}{l} (x, y) \leftarrow (X_{\lambda}, Y_{\lambda}) \\ ((x, c), (y, k)) : \begin{array}{l} k \leftarrow \text{Gen}(1^{\lambda}) \\ c \leftarrow \text{Enc}_k(y) \end{array} \end{array} \right\},$$

$$\text{Ext}_{\mathcal{E}}(f_{\lambda}) : ((x, c), (y, k)) \mapsto f_{\lambda}(x, y).$$

We denote  $\text{Ext}_{\mathcal{E}}(X, Y) := (\text{Ext}_{\mathcal{E}}(X_{\lambda}, Y_{\lambda}))_{\lambda \in \mathbb{N}}$  and  $\text{Ext}_{\mathcal{E}}(f) := (\text{Ext}_{\mathcal{E}}(f_{\lambda}))_{\lambda \in \mathbb{N}}$ .

### 3.1 CC-Homomorphic Encryption

We now introduce our new homomorphic encryption definition. Informally, an encryption scheme  $\mathcal{E}$  is combinatorially homomorphic if there exists a polynomial-time communication protocol for  $\text{Ext}_{\mathcal{E}}(f)$  that utilizes the homomorphic properties of  $\mathcal{E}$  to achieve communication cost that is lower than the standard communication complexity of  $f$ , on a specific input distribution.

We put forward two variants of the definition. Namely, CC-homomorphism in the *perfect correctness regime*, where we require the “homomorphic protocol” for  $\text{Ext}_{\mathcal{E}}(f)$  to have perfect correctness, and CC-homomorphism in the *balanced regime*, where we allow imperfect correctness, but require that the function  $f$  be *balanced*, that is, that  $\Pr[f(x, y) = 0 : (x, y) \leftarrow (X, Y)] = \frac{1}{2}$ . In addition to these two variants, we present an even more general setting in the full version, based on an average-case adaptation of the distributional communication complexity definition.

Our definitions will require the input distribution to be *efficiently sampleable*, defined as follows.

**Definition 11 (Efficiently Sampleable Distribution).** *We say that a distribution ensemble  $(X, Y)$  is efficiently sampleable if there exists a probabilistic polynomial-time sampling algorithm that given  $1^{\lambda}$  outputs a random element from  $(X_{\lambda}, Y_{\lambda})$ .*

**Definition 12 (Communication Complexity Homomorphic Encryption in the Perfect Correctness Regime).** *A private-key encryption scheme  $\mathcal{E}$  (Definition 7) is communication-complexity homomorphic (or CC-homomorphic) in the perfect correctness regime, if there exists a function ensemble  $f$ , an efficiently sampleable product distribution ensemble  $(X, Y)$  and a function  $c = c(\lambda)$  such that,*

- *There exists a polynomial-time one-way protocol that computes  $\text{Ext}_{\mathcal{E}}(f)$  with perfect correctness on input distribution  $\text{Ext}_{\mathcal{E}}(X, Y)$ , using  $c(\lambda)$  bits of communication,*
- *Any unbounded one-way protocol that computes  $f$  on  $(X, Y)$ , using  $c(\lambda)$  bits of communication has correctness at most  $1 - \frac{1}{p(\lambda)}$ , for some polynomial  $p$ .*

*Remark 1.* A natural relaxation of the definition allows a negligible failure probability in the homomorphic communication protocol. However, jumping ahead, having perfect correctness here will be useful as it will also lead to perfect correctness in some of our applications (e.g., lossy encryption, see Theorem 8).

*Remark 2.* Instead of requiring that  $(X, Y)$  is an ensemble of *product* distributions, it is sufficient to require it to be an ensemble of joint distributions such that the conditional distributions  $X|Y$  are efficiently sampleable.

**Definition 13 (Communication Complexity Homomorphic Encryption in the Balanced Regime).** *A private-key encryption scheme  $\mathcal{E}$  (Definition 7) is communication-complexity homomorphic (or CC-homomorphic) in the balanced regime, if there exists a function ensemble  $f$ , an efficiently sampleable product distribution ensemble  $(X, Y)$  and a function  $c = c(\lambda)$  such that,*

- $\Pr [f(x, y) = 0 : (x, y) \leftarrow (X, Y)] = \frac{1}{2}$ ,
- *There exists a polynomial-time one-way protocol that computes  $\text{Ext}_{\mathcal{E}}(f)$  with correctness at least  $\frac{1}{2} + \frac{1}{p(\lambda)}$ , for some polynomial  $p$ , on  $\text{Ext}_{\mathcal{E}}(X, Y)$  using  $c$  bits of communication,*
- *There exists a negligible function  $\mu$  such that any unbounded one-way protocol that computes  $f$  on input distribution  $(X, Y)$  using  $c$  bits of communication has correctness at most  $\frac{1}{2} + \mu(\lambda)$ , for any sufficiently large  $\lambda$ .*

## 4 Applications

In this section, we demonstrate applications of our new notions of homomorphic encryption. In Sect. 4.1 we construct Lossy Encryption. In Sect. 4.2 we construct a Collision Resistant Hash function.

### 4.1 Lossy Encryption

In this section, we show how to use CC-homomorphic encryption to construct lossy public-key encryption.

**Theorem 8 (CC-homomorphic Encryption Implies Lossy Encryption).** *Assume there exists a CC-homomorphic encryption scheme in either the perfect correctness regime (see Definition 12) or the balanced regime (see Definition 13), then there exists a lossy encryption scheme.*

We will prove Theorem 8 in the balanced regime (Definition 13). The proof in the perfect correctness regime (Definition 12) is similar, but produces a  $(1 - \frac{1}{p(\lambda)})$ -lossy encryption, for some polynomial  $p$ , with perfect correctness that can be amplified to full-fledged lossy encryption scheme using Sect. 2.2.

*Proof (Proof of Theorem 8.)* Let  $\mathcal{E} = (\text{Gen}, \text{Enc}, \text{Dec})$  be a  $Y$ -entropic secure CC-homomorphic encryption scheme with respect to function ensemble  $f$  and input product distribution ensemble  $(X, Y)$  such that  $\Pr [f(x, y) = 0 : (x, y) \leftarrow (X, Y)]$



$= \frac{1}{2}$ . Let  $\pi$  be a polynomial-time one-way protocol computing the extended function ensemble  $\text{Ext}_{\mathcal{E}}(f)$  with error less than  $\frac{1}{2} - \frac{1}{\tau(\lambda)}$  on  $\text{Ext}_{\mathcal{E}}(X, Y)$ , for some polynomial  $\tau$ , with communication cost  $c = c(\lambda)$ , such that any unbounded protocol for  $f$  with error  $\frac{1}{2} - \frac{1}{p(\lambda)}$  on  $(X, Y)$ , for some polynomial  $p$ , requires strictly more than  $c$  bits of communication.

For the following, given input  $((x, c), (y, k))$  from  $\text{Ext}_{\mathcal{E}}(X, Y)$ , we denote by  $\text{Alice}(x, c)$  the message Alice generates in the protocol and we denote by  $\text{Bob}(y, k, m_A)$  the output of Bob after receiving a message  $m_A$  from Alice. Consider the following scheme  $(\text{Gen}^*, \text{Enc}^*, \text{Dec}^*, \text{LossyGen}^*)$ .

- **Key generation.** Given a security parameter  $1^\lambda$  the probabilistic polynomial-time algorithm  $\text{Gen}^*$  samples a key  $k \leftarrow \text{Gen}(1^\lambda)$  and an element  $y \leftarrow Y$ , and outputs the public key  $pk = (y, \text{Enc}_k(y))$  and the secret key  $sk = (y, k)$ .
- **Encryption.** Given the public key  $pk = (y, c)$  and a bit  $b$ , the probabilistic polynomial-time algorithm  $\text{Enc}^*$  samples  $x \leftarrow X$  that satisfies  $f(x, y) = b$  (by rejection sampling) and outputs  $m_A = \text{Alice}(x, c)$ .
- **Decryption.** Given the secret key  $sk = (y, k)$  and a ciphertext  $m_A$ , the deterministic polynomial-time algorithm  $\text{Dec}^*$  outputs  $\text{Bob}(y, k, m_A)$ .
- **Lossy Key generation.** Given a security parameter  $1^\lambda$  the probabilistic polynomial-time algorithm  $\text{LossyGen}^*$  samples a key  $k \leftarrow \text{Gen}(1^\lambda)$  and elements  $y, y' \leftarrow Y$ , and outputs the lossy key  $lk = (y, \text{Enc}_k(y'))$ .

*Claim.* The scheme satisfies correctness (see Definition 9).

*Proof.* For any  $\lambda \in \mathbb{N}$ ,

$$\begin{aligned} \Pr \left[ \text{Dec}_{sk}^*(\text{Enc}_{pk}^*(b)) \neq b \right] &\stackrel{(1)}{=} \Pr \left[ \text{Bob}(y, k, \text{Alice}(x, c)) \neq f(x, y) : \begin{array}{l} (x, y) \leftarrow (X, Y) \\ \text{s.t. } f(x, y) = b \end{array} \right] \\ &\stackrel{(2)}{=} \Pr \left[ \text{Bob}(y, k, \text{Alice}(x, c)) \neq f(x, y) : (x, y) \leftarrow (X, Y) \right] \\ &\stackrel{(3)}{\leq} \frac{1}{2} - \frac{1}{\tau(\lambda)}, \end{aligned}$$

where  $b \leftarrow \{0, 1\}$ ,  $(sk, pk) \leftarrow \text{Gen}^*(1^\lambda)$ ,  $k \leftarrow \text{Gen}(1^\lambda)$  and  $c \leftarrow \text{Enc}_k(y)$ , and where (1) is by the definition of the scheme, (2) is since  $\Pr[f(x, y) = 0 : (x, y) \leftarrow (X, Y)] = \frac{1}{2}$ , and therefore sampling  $b \leftarrow \{0, 1\}$  and then sampling from  $(X, Y)$  conditioned on  $f(x, y) = b$  is the same as sampling directly from  $(X, Y)$ , and (3) is since the protocol  $\pi$  computes  $\text{Ext}_{\mathcal{E}}(f)$  on  $\text{Ext}_{\mathcal{E}}(X, Y)$  with error less than  $B - \frac{1}{\tau(\lambda)}$ , and since  $((x, c), (y, k))$  is sampled similarly to a random sample from  $\text{Ext}_{\mathcal{E}}(X, Y)$ .

*Claim.* The scheme satisfies key indistinguishability (see Definition 9).

*Proof.* We have that for any fixed  $y$  and  $y'$  sampled from  $Y$ ,

$$\left( \text{Proj}_2(\text{Gen}^*(1^\lambda)) \right)_{\lambda \in \mathbb{N}} = (y, c)_{\lambda \in \mathbb{N}} \approx_c (y, c')_{\lambda \in \mathbb{N}} = (\text{LossyGen}^*(1^\lambda))_{\lambda \in \mathbb{N}},$$

where  $k \leftarrow \text{Gen}(1^\lambda)$ ,  $c \leftarrow \text{Enc}_k(y)$  and  $c' \leftarrow \text{Enc}_k(y')$ , and where the equalities are by the definition of the scheme and the computational indistinguishability is by the  $Y$ -entropic security of  $\mathcal{E}$ .

*Claim.* The scheme satisfies lossiness of lossy keys (see Definition 9).

*Proof.* We will show that given an unbounded distinguisher for encryptions under a lossy key, with non-negligible distinguishing advantage, one can construct a one-way protocol in the standard distributional communication complexity model (Sect. 2.1) that computes  $f$  with correctness  $\frac{1}{2} + \frac{1}{p(\lambda)}$  on  $(X, Y)$ , for some polynomial  $p$ , with communication cost  $c$ . Such a protocol cannot exist by our assumption that  $\mathcal{E}$  is CC-homomorphic in the balanced regime with respect to  $f$  and  $(X, Y)$  (see Definition 13).

Assume towards a contradiction that there exists a (computationally unbounded) distinguisher  $\mathcal{D}$  and a polynomial  $p$  such that for infinitely many  $\lambda \in \mathbb{N}$ ,

$$\Pr \left[ \mathcal{D}(lk, \text{Enc}_{lk}^*(b)) = b : b \leftarrow \{0, 1\}, lk \leftarrow \text{LossyGen}^*(1^\lambda) \right] \geq \frac{1}{2} + \frac{1}{p(\lambda)}.$$

By the definitions of  $\text{LossyGen}^*$  and  $\text{Enc}^*$  we have that for infinitely many  $\lambda \in \mathbb{N}$ ,

$$\Pr \left[ \mathcal{D}(y, c, \text{Alice}(x, c)) = f(x, y) \right] \geq \frac{1}{2} + \frac{1}{p(\lambda)},$$

where  $x \leftarrow X$ ,  $y, y' \leftarrow Y$ ,  $k \leftarrow \text{Gen}(1^\lambda)$  and  $c \leftarrow \text{Enc}_k(y')$ .

We start by constructing a protocol in the standard distributional communication complexity model (Sect. 2.1) that uses shared randomness which we will eliminate later. Consider the following unbounded one-way protocol  $\pi^*$  between parties  $\text{Alice}^*$  and  $\text{Bob}^*$  who are given inputs  $x$  and  $y$  sampled from  $(X, Y)$  and have access to shared random coins.

1.  $\text{Alice}^*$  and  $\text{Bob}^*$  sample a key  $k \leftarrow \text{Gen}(1^\lambda)$ , an element  $y' \leftarrow Y$  and an encryption  $c \leftarrow \text{Enc}_k(y')$  using the shared random coins.
2.  $\text{Alice}^*$  sends  $m_A = \text{Alice}(x, c)$  to  $\text{Bob}^*$ .
3.  $\text{Bob}^*$  runs  $\mathcal{D}$  on  $(y, c, m_A)$  and outputs its answer.

We denote by  $\pi^*(x, y; r)$  the output of the protocol on inputs  $(x, y)$  and random coins  $r$ . infinitely many  $\lambda \in \mathbb{N}$ ,

$$\Pr \left[ \pi^*(x, y; r) = f(x, y) : \begin{matrix} (x, y) \leftarrow (X, Y) \\ r \leftarrow \{0, 1\}^* \end{matrix} \right] = \Pr \left[ \mathcal{D}(y, c, \text{Alice}(x, c)) = f(x, y) \right] \geq \frac{1}{2} + \frac{1}{p(\lambda)},$$

where  $x \leftarrow X$ ,  $y, y' \leftarrow Y$ ,  $k \leftarrow \text{Gen}(1^\lambda)$  and  $c \leftarrow \text{Enc}_k(y')$ .

The above statement holds over a random choice of  $r$ . However, by an averaging argument, for infinitely many  $\lambda \in \mathbb{N}$  there exists a fixed randomness  $r^*$  such that

$$\Pr \left[ \pi^*(x, y; r^*) = f(x, y) : (x, y) \leftarrow (X, Y) \right] \geq \frac{1}{2} + \frac{1}{p(\lambda)}.$$

To conclude, we have that  $\pi^*$  with fixed random coins  $r^*$  is an unbounded one-way protocol that computes  $f$  with error less than  $\frac{1}{2} - \frac{1}{p(\lambda)}$  on  $(X, Y)$  with communication cost  $|\text{Alice}(x, c)| = c$ , which is a contradiction to the assumption that such a protocol cannot exist.

## 4.2 Collision Resistant Hash Function

Next, we use a variant of CC-homomorphic encryption to construct a collision resistant hash function. First, we define an *efficient encoding* algorithm for a set  $X$ .

**Definition 14 (Efficient Encoding).** Let  $X = (X_\lambda)_{\lambda \in \mathbb{N}}$  be an ensemble of finite sets. We say that  $X$  supports an efficient encoding with input length  $\ell = \ell(\lambda)$  if there exists an efficiently computable (polynomial-time) injective function  $\text{Encode} : \{0, 1\}^\ell \rightarrow X_\lambda$ .

Our CRH construction will require a function  $f$  and input distribution  $(X, Y)$  such that the ensemble  $f_Y = (f_\lambda)_{\lambda \in \mathbb{N}}$ , where  $f_\lambda := \{f(\cdot, y) : y \in Y_\lambda\}$ , is a universal hash function family. We put forward the definition.

**Definition 15 (Universal Hash Function Family).** A set  $H$  of functions from  $X$  to  $\{0, 1\}$  is a universal hash function family if for every distinct  $x_1, x_2 \in X$  the hash function family  $H$  satisfies the following constraint.

$$\Pr \left[ h(x_1) = h(x_2) : h \leftarrow H \right] \leq \frac{1}{2}.$$

**Theorem 9 (CC-Homomorphic Encryption Implies CRH).** Assume there exists a CC-homomorphic encryption scheme (Definitions 12 and 13) with respect to function  $f$ , input distribution  $(X, Y)$  and parameter  $c$  that satisfies the following conditions.

- The function ensemble  $\left( \{f(\cdot, y) : y \in Y_\lambda\} \right)_{\lambda \in \mathbb{N}}$  is a universal hash function family.
- The polynomial-time protocol for  $\text{Ext}_\mathcal{E}(f)$  is correct on any input from  $\text{Ext}_\mathcal{E}(X, Y)$  w.p.  $\frac{1}{2} + \frac{1}{p(\lambda)}$ , for some polynomial  $p$ ,
- The ensemble  $X$  supports an efficient encoding with input length  $\ell(\lambda) \geq c(\lambda)$  for any sufficiently large  $\lambda$ .

Then, there exists a collision resistant hash function (Definition 10).

*Remark 3.* As a matter of fact, similarly to [26], a relaxed notion of encryption with an inefficient decryption algorithm (in other words, a commitment scheme) is sufficient.

We will prove Theorem 9 in the balanced regime (Definition 13), but it can also be adapted to the perfect correctness regime (Definition 12).

*Proof (Proof of Theorem 9).* Let  $f$  be a function ensemble and  $(X, Y)$  be an input distribution ensemble such that  $\left(\{f(\cdot, y) : y \in Y_\lambda\}\right)_{\lambda \in \mathbb{N}}$  is a universal hash function family and such that  $X$  supports an efficient encoding with input length  $\ell = \ell(\lambda)$ . Let  $\mathcal{E} = (\text{Gen}, \text{Enc}, \text{Dec})$  be a  $Y$ -entropic secure encryption scheme. Let  $\pi$  be a polynomial-time one-way protocol computing the extended function ensemble  $\text{Ext}_{\mathcal{E}}(f)$  with correctness  $\frac{1}{2} + \frac{1}{p(\lambda)}$  on any input from  $\text{Ext}_{\mathcal{E}}(X, Y)$ , for some polynomial  $p$ , with communication cost  $\ell'(\lambda) < \ell(\lambda)$ .

Consider the following scheme  $(\text{Gen}^*, \text{Eval}^*)$ .

- **Key generation.** Given security parameter  $1^\lambda$ , the probabilistic polynomial-time algorithm  $\text{Gen}^*$  samples  $y \leftarrow Y$ ,  $k \leftarrow \text{Gen}(1^\lambda)$  and  $s \leftarrow \text{Enc}_k(y)$  and outputs  $s$ .
- **Evaluation.** Given index  $s$  and input  $m \in \{0, 1\}^{\ell(\lambda)}$ , the polynomial-time algorithm  $\text{Eval}^*$  outputs  $\text{Alice}(\text{Encode}(m), s)$ .

We first show that the scheme indeed compresses. Indeed, for any  $\lambda \in \mathbb{N}$ ,  $s \leftarrow \text{Gen}^*(1^\lambda)$  and  $m \in \{0, 1\}^{\ell(\lambda)}$ ,

$$|h_s(m)| = \left| \text{Alice}(\text{Encode}(m), s) \right| \leq \ell'(\lambda) < \ell(\lambda).$$

Assume towards a contradiction that the scheme is not collision resistant. Therefore, there exists a probabilistic polynomial-size adversary  $\mathcal{A}$  and a polynomial  $q$  such that for infinitely many  $\lambda \in \mathbb{N}$ ,

$$\Pr \left[ m \neq m' \wedge h_s(m) = h_s(m') : \begin{array}{l} s \leftarrow \text{Gen}^*(1^\lambda), \\ (m, m') \leftarrow \mathcal{A}(s) \end{array} \right] = \frac{1}{q(\lambda)}.$$

Consider the distinguisher  $\mathcal{D}$  for the  $Y$ -entropic security of  $\mathcal{E}$ . Given  $(y_0, c)$ , where  $k \leftarrow \text{Gen}(1^\lambda)$ ,  $y_0, y_1 \leftarrow Y$ ,  $b \leftarrow \{0, 1\}$  and  $c \leftarrow \text{Enc}_k(y_b)$ , the distinguisher  $\mathcal{D}$  computes  $(m, m') \leftarrow \mathcal{A}(c_b)$ . It then checks that  $m \neq m'$ , that  $h_c(m) = h_c(m')$  and that  $f(\text{Encode}(m), y_0) = f(\text{Encode}(m'), y_0)$ . If all checks pass, it outputs 1. Otherwise, it outputs a random bit. For the following, we denote  $x := \text{Encode}(m)$ ,  $x' := \text{Encode}(m')$ .

We first consider the case where  $b = 0$ . Given  $k \leftarrow \text{Gen}(1^\lambda)$ ,  $y_0 \leftarrow Y$ ,  $c \leftarrow \text{Enc}_k(y_0)$  and  $(m, m') \leftarrow \mathcal{A}(c)$ , we define the following events,

1. The event  $E_1$  where  $f(x, y_0) = f(x', y_0)$ .
2. The event  $E_2$  where  $m \neq m'$  and  $h_c(m) = h_c(m')$ .
3. The event  $E_3$  where  $\pi((x, c), (y_0, k)) = \pi((x', c), (y_0, k))$ .
4. The event  $E_4$  where the protocol  $\pi$  is correct on both  $((x, c), (y_0, k))$  and  $((x', c), (y_0, k))$ , or is wrong on both of them.

First, since  $\pi$  is correct on any input w.p. at least  $\frac{1}{2} + \frac{1}{p(\lambda)}$ , there exists a function  $\tau : \mathbb{N} \rightarrow \mathbb{N}$  such that  $\pi$  is correct on any input w.p. exactly  $\frac{1}{2} + \frac{1}{\tau(\lambda)}$ , and  $\tau(\lambda) \leq p(\lambda)$  for any  $\lambda \in \mathbb{N}$ . Therefore,

$$\Pr[E_4] = \left(\frac{1}{2} + \frac{1}{\tau(\lambda)}\right)^2 + \left(\frac{1}{2} - \frac{1}{\tau(\lambda)}\right)^2 = \frac{1}{2} + \frac{2}{\tau^2(\lambda)} \geq \frac{1}{2} + \frac{2}{p^2(\lambda)}. \quad (1)$$

Furthermore, we have that,

$$\begin{aligned}
 \Pr [E_1|E_2] &\stackrel{(1)}{=} \Pr [E_1|E_2 \wedge E_3] \\
 &\geq \Pr [E_1 \wedge E_4|E_2 \wedge E_3] \\
 &\stackrel{(2)}{=} \Pr [E_1|E_2 \wedge E_3 \wedge E_4] \cdot \Pr [E_4] \\
 &\stackrel{(3)}{=} \Pr [E_4],
 \end{aligned} \tag{2}$$

where (1) is since assuming  $E_2$  happened, we have that  $\text{Alice}(x, c) = h_c(m) = h_c(m') = \text{Alice}(x', c)$ , and therefore, since  $\pi$  is a deterministic one-way protocol, we have that  $\pi((x, c), (y_0, k)) = \pi((x', c), (y_0, k))$ , (2) is by conditional probability, and (3) is since if the protocol outputs the same output on both inputs and is correct on both of them or wrong on both of them, then  $f(x, y_0) = f(x', y_0)$ .

Finally, for infinitely many  $\lambda \in \mathbb{N}$  we have that,

$$\begin{aligned}
 \Pr [\mathcal{D}(y_0, c) = 1] &\stackrel{(1)}{=} \Pr [E_1 \wedge E_2] + \frac{1}{2} \cdot (1 - \Pr [E_1 \wedge E_2]) \\
 &= \frac{1}{2} + \frac{1}{2} \cdot \Pr [E_1 \wedge E_2] \\
 &= \frac{1}{2} + \frac{1}{2} \Pr [E_1|E_2] \cdot \Pr [E_2] \\
 &\stackrel{(2)}{=} \frac{1}{2} + \frac{1}{2q(\lambda)} \Pr [E_1|E_2] \\
 &\stackrel{(3)}{\geq} \frac{1}{2} + \frac{1}{2q(\lambda)} \cdot \left( \frac{1}{2} + \frac{2}{p^2(\lambda)} \right),
 \end{aligned}$$

where  $k \leftarrow \text{Gen}(1^\lambda)$ ,  $y_0 \leftarrow Y$ ,  $c \leftarrow \text{Enc}_k(y_0)$  and  $(m, m') \leftarrow \mathcal{A}(c)$ , and where (1) is by the definition of  $\mathcal{D}$ , (2) is since  $\mathcal{D}$  simulates for the adversary  $\mathcal{A}$  a proper collision resistant game, and event  $E_2$  is the event where  $\mathcal{A}$  wins in this game, which happens w.p.  $1/q(\lambda)$ , and (3) is by Eqs. (1) and (2).

On the other hand, for the case where  $b = 1$ , we have that for any  $\lambda \in \mathbb{N}$ ,

$$\begin{aligned}
 \Pr [\mathcal{D}(y_0, c) = 1] &\stackrel{(1)}{=} \frac{1}{2} + \frac{1}{2q(\lambda)} \Pr [f(x, y_0) = f(x', y_0) | m \neq m' \wedge h_s(m) = h_s(m')] \\
 &\leq \frac{1}{2} + \frac{1}{2q(\lambda)} \Pr [f(x, y_0) = f(x', y_0)] \\
 &\stackrel{(2)}{=} \frac{1}{2} + \frac{1}{2q(\lambda)} \cdot \frac{1}{2},
 \end{aligned}$$

where  $k \leftarrow \text{Gen}(1^\lambda)$ ,  $y_0, y_1 \leftarrow Y$ ,  $c \leftarrow \text{Enc}_k(y_1)$  and  $(m, m') \leftarrow \mathcal{A}(c)$ , and where (1) follows by similar reasoning as in the case where  $b = 0$  and (2) is since  $x$  and  $x'$  are independent of  $y_0$  and since  $f_Y$  is a universal hash family, and therefore the probability that  $f(x, y_0) = f(x', y_0)$  is  $1/2$ .

Therefore, for infinitely many  $\lambda \in \mathbb{N}$ ,

$$\begin{aligned}
 \left| \Pr [\mathcal{D}(y_0, c_0) = 1] - \Pr [\mathcal{D}(y_0, c_1) = 1] \right| &\geq \left( \frac{1}{2} + \frac{1}{2q(\lambda)} \cdot \left( \frac{1}{2} + \frac{2}{p^2(\lambda)} \right) \right) - \left( \frac{1}{2} + \frac{1}{2q(\lambda)} \cdot \frac{1}{2} \right) \\
 &= \frac{2}{2q(\lambda) \cdot p^2(\lambda)},
 \end{aligned}$$

where  $k \leftarrow \text{Gen}(1^\lambda)$ ,  $y_0, y_1 \leftarrow Y$  and  $c_b \leftarrow \text{Enc}_k(y_b)$  for  $b \in \{0, 1\}$ , in contradiction to the assumption that  $\mathcal{E}$  is  $Y$ -entropic secure.

## 5 Instantiations

### 5.1 Low Noise LPN

In this section we will construct a CC-homomorphic encryption scheme from low noise LPN, thereby giving a conceptually simple derivation of recent results [7, 10, 44]. We first present the learning parity with noise assumption. For  $\mu \in [0, 1]$  we denote by  $\text{Ber}_\mu$  the Bernoulli distribution with mean  $\mu$ .

**Definition 16 (Learning Parity with Noise Assumption).** *For noise rate  $\mu = \mu(\lambda) \in (0, \frac{1}{2})$ , the  $\text{LPN}_\mu$  assumption is that for any  $m(\lambda) = \lambda^{O(1)}$ ,*

$$(A, As + e)_{\lambda \in \mathbb{N}} \approx_c (A, u)_{\lambda \in \mathbb{N}},$$

where  $A \leftarrow \mathbb{F}_2^{m \times \lambda}$ ,  $s \leftarrow \mathbb{F}_2^\lambda$ ,  $e \leftarrow \text{Ber}_\mu^m$  and  $u \leftarrow \mathbb{F}_2^m$ .

**Theorem 10 (CC-homomorphic Encryption from Low Noise LPN).** *Assuming  $\text{LPN}_{\frac{\log^2 \lambda}{\lambda}}$  (Definition 16) there exists a CC-homomorphic encryption scheme in the balanced regime (Definition 13).*

In fact, we will construct a CC-homomorphic encryption scheme that satisfies the conditions of Theorem 9, thus deriving the following two theorems.

**Theorem 11 (Lossy Encryption from Low Noise LPN).** *Assuming  $\text{LPN}_{\frac{\log^2 \lambda}{\lambda}}$  (Definition 16) there exists a lossy encryption scheme (Definition 9).*

**Theorem 12 (CRH from Low Noise LPN).** *Assuming  $\text{LPN}_{\frac{\log^2 \lambda}{\lambda}}$  (Definition 16) there exists a collision resistant hash function (Definition 10).*

Theorems 11 and 12 follows directly from Theorems 8 to 10. We note however that we do not know how to use LPN to derive a similar result to Alekhnovich’s scheme [3] via our framework. Indeed, the stronger conclusions implied by our framework (lossy encryption, CRH) are not known from the flavor of LPN used by Alekhnovich.

We now describe a private-key encryption scheme  $\mathcal{E} = (\text{Gen}, \text{Enc}, \text{Dec})$  based on low noise LPN.

- **Key generation.** Given a security parameter  $1^\lambda$ , the probabilistic algorithm  $\text{Gen}$  outputs a private key  $s \leftarrow \mathbb{F}_2^\lambda$ .
- **Encryption.** Given a message  $y \in \mathbb{F}_2^{\lambda^2}$  and a private key  $s$ , the probabilistic algorithm  $\text{Enc}$  samples a random matrix  $A \leftarrow \mathbb{F}_2^{\lambda^2 \times \lambda}$  and a random noise  $e \leftarrow \text{Ber}_{\frac{\lambda^2}{\lambda}}$ , and outputs a ciphertext  $(A, A \cdot s + e + y)$ .
- **Decryption.** Given a ciphertext  $(A, b)$ , the deterministic algorithm  $\text{Dec}$  outputs  $b - A \cdot s$ .

We define the following homomorphic operation that supports ciphertext-plaintext multiplication.

- **Ciphertext-plaintext multiplication.** Given a plaintext  $x \in \mathbb{F}_2^{\lambda^2}$  and a ciphertext  $(A, b)$ , where  $A \in \mathbb{F}_2^{\lambda^2 \times \lambda}$  and  $b \in \mathbb{F}_2^{\lambda^2}$ , the deterministic algorithm `PlainMult` outputs  $(x^\top \cdot A, x^\top \cdot b)$ .

We will show that  $\mathcal{E}$  is CC-homomorphic with respect to the inner product functionality  $f = (f_\lambda(x, y) = x^\top y)_{\lambda \in \mathbb{N}}$  over the uniform input distribution  $(X, Y)$  where  $X$  and  $Y$  contain vectors in  $\mathbb{F}_2^{\lambda^2}$ , while  $X_\lambda$  is restricted to vectors with Hamming weight  $\frac{2\lambda}{\log \lambda}$ . Looking ahead, we will construct a polynomial-time protocol for  $\text{Ext}_\mathcal{E}(f)$  with correctness  $\frac{1}{2} + \frac{1}{p(\lambda)}$  on  $\text{Ext}_\mathcal{E}(X, Y)$ , for some polynomial  $p$ , that uses  $c = c(\lambda) = \lambda + 1$  bits of communication. Furthermore, we will show that there exists a negligible function  $\mu$  such that any unbounded one-way protocol that computes  $f$  on  $(X, Y)$  using  $c$  bits of communication has correctness at most  $\frac{1}{2} + \mu(\lambda)$ , for any sufficiently large  $\lambda$ .

Notice that

$$\Pr [f(x, y) = 0 : (x, y) \leftarrow (X, Y)] = \frac{1}{2},$$

and that  $(\{f(\cdot, y) : y \in Y_\lambda\})_{\lambda \in \mathbb{N}}$  is a universal hash function family. Furthermore, the ensemble  $X$  supports an efficient encoding with input length  $2\lambda \geq c$ , for any sufficiently large  $\lambda$ . Namely, given a vector  $m \in \mathbb{F}_2^{2\lambda}$  we map every  $\log \lambda$  bits of  $m$  to a unit vector in  $\mathbb{F}_2^\lambda$ . Then, we concatenate these unit vectors to a vector in  $\mathbb{F}_2^{\lambda^2}$  with Hamming weight  $\frac{2\lambda}{\log \lambda}$ .

First, we will show that the private-key encryption scheme  $\mathcal{E}$  is  $Y$ -entropic secure (Definition 7).

*Claim (Y-Entropic Security of  $\mathcal{E}$ ).* Assuming  $\text{LPN}_{\frac{\log^2 \lambda}{\lambda}}$  (Definition 16), for every  $\lambda \in \mathbb{N}$  and  $y, y' \leftarrow \mathbb{F}_2^{\lambda^2}$  we have that,

$$(y, \text{Enc}_s(y))_{\lambda \in \mathbb{N}} \approx_c (y, \text{Enc}_s(y'))_{\lambda \in \mathbb{N}},$$

where  $s \leftarrow \text{Gen}(1^\lambda)$ .

*Proof.* For any fixed  $y, y' \in \mathbb{F}_2^{\lambda^2}$ ,

$$(y, \text{Enc}_s(y'))_{\lambda \in \mathbb{N}} = (y, (A, A \cdot s + e + y'))_{\lambda \in \mathbb{N}} \approx_c (y, (A, u + y'))_{\lambda \in \mathbb{N}} = (y, (A, u))_{\lambda \in \mathbb{N}} \quad (*)$$

where  $u \leftarrow \mathbb{F}_2^{\lambda^2}$ ,  $s \leftarrow \mathbb{F}_2^\lambda$ ,  $A \leftarrow \mathbb{F}_2^{\lambda^2 \times \lambda}$  and  $e \leftarrow \text{Ber}_{\frac{\log^2 \lambda}{\lambda}}$ , and where  $(*)$  holds by the  $\text{LPN}_{\frac{\log^2 \lambda}{\lambda}}$  assumption.

Now, consider the following polynomial-time one-way protocol for the extended function ensemble  $\text{Ext}_\mathcal{E}(f)$ . Given inputs  $x$  and  $c = \text{Enc}_k(y)$ , Alice computes  $m_A = \text{PlainMult}(x, c)$  and sends it to Bob, who outputs  $\text{Dec}_k(m_A)$ .

The communication cost of this protocol is  $c(\lambda) = |m_A| = \lambda + 1$ . We show the correctness probability of the protocol using the Piling-Up Lemma.

**Lemma 1 (The Piling-Up Lemma [33]).** *Let  $e_1, \dots, e_k \in \mathbb{F}_2$  be i.i.d. random variables such that  $\Pr[e_i = 1] = \epsilon$ , then*

$$\Pr \left[ \bigoplus_{i=1}^k e_i = 0 \right] = \frac{1}{2} + \frac{1}{2}(1 - 2\epsilon)^k.$$

*Claim (Protocol Correctness).* For every  $\lambda \in \mathbb{N}$ ,  $x \in X$  and  $y \in Y$  we have that

$$\Pr \left[ \text{Dec}_s \left( \text{PlainMult}(x, \text{Enc}_s(y)) \right) = x^\top \cdot y : s \leftarrow \text{Gen}(1^\lambda) \right] > \frac{1}{2} + \frac{1}{2\lambda^8}.$$

*Proof.* By the definition of  $\mathcal{E}$  it's enough to show that  $\Pr[x^\top \cdot e = 0] > \frac{1}{2} + \frac{1}{2\lambda^8}$ . By Lemma 1 we have that

$$\Pr[x^\top \cdot e = 0] = \Pr \left[ \bigoplus_{i=1}^{\frac{2\lambda}{\log \lambda}} e_i = 0 \right] \geq \frac{1}{2} + \frac{1}{2} (1 - 2 \frac{\log^2 \lambda}{\lambda})^{\frac{2\lambda}{\log \lambda}} \geq \frac{1}{2} + \frac{1}{2} \cdot 2^{-4 \frac{\log^2 \lambda}{\lambda} \frac{2\lambda}{\log \lambda}} = \frac{1}{2} + \frac{1}{2\lambda^8},$$

where the second inequality holds since  $1 - x \geq 2^{-2x}$  for  $x \leq \frac{1}{2}$ .

Finally, we will show that for the negligible function  $\mu = 2^{-\lambda}$  we have that any unbounded one-way protocol that computes  $f$  on input distribution  $(X, Y)$  using  $c(\lambda) = \lambda + 1$  bits of communication has correctness at most  $\frac{1}{2} + \mu(\lambda)$ , for any sufficiently large  $\lambda$ .

*Claim (Distributional Communication Complexity Lower Bound for  $f$ ).* For any  $\lambda \in \mathbb{N}$ ,

$$\mathcal{D}^{A \rightarrow B}(f, (X, Y), \frac{1}{2} - 2^{-\lambda}) = 2\lambda$$

*Proof.* Take  $\lambda \in \mathbb{N}$ . Let  $H$  be a matrix such that  $H(x, y) = (-1)^{\langle x, y \rangle}$ . It is easy to check that the matrix  $H$  satisfies  $HH^\top = H^\top H = 2^{\lambda^2} I$ . Therefore,  $\|H\| = \sqrt{2^{\lambda^2}}$ . Let  $R = S \times T$  be a rectangle on  $(X_\lambda, Y_\lambda)$ . We have that

$$\begin{aligned} \text{Disc}(f_\lambda; S \times T) &\stackrel{(1)}{=} \left| \sum_{(x,y) \in S \times T} \Pr[x, y \in (X, Y)] (-1)^{\langle x, y \rangle} \right| \\ &\stackrel{(2)}{\leq} \left| \sum_{(x,y) \in S \times T} \frac{1}{\left(\frac{\lambda^2}{\log \lambda}\right)} \frac{1}{2\lambda^2} H(x, y) \right| \\ &= \frac{1}{\left(\frac{\lambda^2}{\log \lambda}\right)} \frac{1}{2\lambda^2} |\mathbf{1}_S \cdot H \cdot \mathbf{1}_T| \\ &\stackrel{(3)}{\leq} \frac{1}{\left(\frac{\lambda^2}{\log \lambda}\right)} \frac{1}{2\lambda^2} \|\mathbf{1}_S\| \cdot \|H\| \cdot \|\mathbf{1}_T\| \\ &\stackrel{(4)}{\leq} \frac{1}{\left(\frac{\lambda^2}{\log \lambda}\right)} \frac{1}{2\lambda^2} \sqrt{\left(\frac{\lambda^2}{\log \lambda}\right)} \cdot 2^{\frac{\lambda^2}{2}} \cdot 2^{\frac{\lambda^2}{2}} \\ &= \frac{1}{\sqrt{\left(\frac{\lambda^2}{\log \lambda}\right)}}, \end{aligned}$$



where (1) is by definition, (2) is since  $X_\lambda$  and  $Y_\lambda$  are independent and distributed uniformly over vectors with Hamming weight  $\frac{2\lambda}{\log \lambda}$  in  $\mathbb{F}_2^{\lambda^2}$  and over  $\mathbb{F}_2^{\lambda^2}$  respectively, (3) is by Cauchy-Schwarz and (4) is since  $\|H\| = \sqrt{2^{\lambda^2}}$  and since  $S$  and  $T$  can contain at most  $\binom{\lambda^2}{\frac{2\lambda}{\log \lambda}}$  and  $2^{\lambda^2}$  elements respectively. Therefore, by Theorem 7 we have for error-rate  $\epsilon(\lambda) = \frac{1}{2} - 2^{-\lambda}$  the following,

$$\begin{aligned} \mathcal{D}^{A \rightarrow B}(f) &\geq \log \left( \frac{1 - 2\epsilon(\lambda)}{\text{Disc}(f, (X, Y))} \right) \\ &\geq \frac{1}{2} \log \left( \frac{\lambda^2}{\frac{2\lambda}{\log \lambda}} \right) - \lambda \\ &\stackrel{(*)}{=} \frac{\lambda}{\log \lambda} \cdot \log \left( \frac{1}{2} \lambda \log \lambda \right) - \lambda \\ &\geq 2\lambda \end{aligned}$$

where (\*) is since  $\binom{n}{k} \geq \left(\frac{n}{k}\right)^k$  for any  $n$  and  $k$ .

**Acknowledgments.** We thank Aayush Jain and the TCC reviewers for their helpful comments. Y. Ishai was supported in part by ERC Project NTSC (742754), BSF grant 2018393, and ISF grant 2774/20. R. Rothblum is funded by the European Union (ERC, FASTPROOF, 101041208). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

## References

1. Alamati, N., Montgomery, H., Patranabis, S.: Symmetric primitives with structured secrets. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019. LNCS, vol. 11692, pp. 650–679. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-26948-7\\_23](https://doi.org/10.1007/978-3-030-26948-7_23)
2. Alamati, N., Montgomery, H., Patranabis, S., Roy, A.: Minicrypt primitives with algebraic structure and applications. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019. LNCS, vol. 11477, pp. 55–82. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-17656-3\\_3](https://doi.org/10.1007/978-3-030-17656-3_3)
3. Alekhovich, M.: More on average case vs approximation complexity. In: Proceedings of the 44th Symposium on Foundations of Computer Science, FOCS 2003, 11–14 October 2003, Cambridge, MA, USA, pp. 298–307. IEEE Computer Society (2003). <https://doi.org/10.1109/SFCS.2003.1238204>
4. Babai, L., Frankl, P., Simon, J.: Complexity classes in communication complexity theory. In: 27th Annual Symposium on Foundations of Computer Science, SFCS 1986, pp. 337–347. IEEE (1986)
5. Bellare, M., Hofheinz, D., Yilek, S.: Possibility and impossibility results for encryption and commitment secure under selective opening. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 1–35. Springer, Heidelberg (2009). [https://doi.org/10.1007/978-3-642-01001-9\\_1](https://doi.org/10.1007/978-3-642-01001-9_1)

6. Benaloh, J.: Dense probabilistic encryption. In: Selected Areas of Cryptography, May 1994. <https://www.microsoft.com/en-us/research/publication/dense-probabilistic-encryption/>
7. Bitansky, N., Freizeit, S.: Statistically sender-private OT from LPN and derandomization. In: Dodis, Y., Shrimpton, T. (eds.) Advances in Cryptology, CRYPTO 2022. LNCS, vol. 13509. Springer, Cham (2022). [https://doi.org/10.1007/978-3-031-15982-4\\_21](https://doi.org/10.1007/978-3-031-15982-4_21)
8. Bogdanov, A., Lee, C.H.: Limits of provable security for homomorphic encryption. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8042, pp. 111–128. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-40041-4\\_7](https://doi.org/10.1007/978-3-642-40041-4_7)
9. Boneh, D., Goh, E.-J., Nissim, K.: Evaluating 2-DNF formulas on ciphertexts. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 325–341. Springer, Heidelberg (2005). [https://doi.org/10.1007/978-3-540-30576-7\\_18](https://doi.org/10.1007/978-3-540-30576-7_18)
10. Brakerski, Z., Lyubashevsky, V., Vaikuntanathan, V., Wichs, D.: Worst-case hardness for LPN and cryptographic hashing via code smoothing. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019. LNCS, vol. 11478, pp. 619–635. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-17659-4\\_21](https://doi.org/10.1007/978-3-030-17659-4_21)
11. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. SIAM J. Comput. **43**(2), 831–871 (2014). <https://doi.org/10.1137/120868669>
12. Cachin, C., Camenisch, J., Kilian, J., Müller, J.: One-round secure computation and secure autonomous mobile agents. In: Montanari, U., Rolim, J.D.P., Welzl, E. (eds.) ICALP 2000. LNCS, vol. 1853, pp. 512–523. Springer, Heidelberg (2000). [https://doi.org/10.1007/3-540-45022-X\\_43](https://doi.org/10.1007/3-540-45022-X_43)
13. Cohen, S.P., Naor, M.: Low communication complexity protocols, collision resistant hash functions and secret key-agreement protocols. In: Dodis, Y., Shrimpton, T. (eds.) Advances in Cryptology, CRYPTO 2022. LNCS, vol. 13509, pp. 252–281. Springer, Cham (2022). [https://doi.org/10.1007/978-3-031-15982-4\\_9](https://doi.org/10.1007/978-3-031-15982-4_9)
14. Di Crescenzo, G., Malkin, T., Ostrovsky, R.: Single database private information retrieval implies oblivious transfer. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 122–138. Springer, Heidelberg (2000). [https://doi.org/10.1007/3-540-45539-6\\_10](https://doi.org/10.1007/3-540-45539-6_10)
15. Damgård, I., Jurik, M.: A generalisation, a simplification and some applications of Paillier’s probabilistic public-key system. In: Kim, K. (ed.) PKC 2001. LNCS, vol. 1992, pp. 119–136. Springer, Heidelberg (2001). [https://doi.org/10.1007/3-540-44586-2\\_9](https://doi.org/10.1007/3-540-44586-2_9)
16. Dwork, C., Naor, M., Reingold, O.: Immunizing encryption schemes from decryption errors. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 342–360. Springer, Heidelberg (2004). [https://doi.org/10.1007/978-3-540-24676-3\\_21](https://doi.org/10.1007/978-3-540-24676-3_21)
17. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 10–18. Springer, Heidelberg (1985). [https://doi.org/10.1007/3-540-39568-7\\_2](https://doi.org/10.1007/3-540-39568-7_2)
18. Gentry, C.: A fully homomorphic encryption scheme. Ph.D. thesis, Stanford University, USA (2009). <https://searchworks.stanford.edu/view/8493082>
19. Gentry, C., Halevi, S., Vaikuntanathan, V.:  $i$ -hop homomorphic encryption and rerandomizable Yao circuits. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 155–172. Springer, Heidelberg (2010). [https://doi.org/10.1007/978-3-642-14623-7\\_9](https://doi.org/10.1007/978-3-642-14623-7_9)

20. Gentry, C., Halevi, S., Vaikuntanathan, V.: A simple BGN-type cryptosystem from LWE. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 506–522. Springer, Heidelberg (2010). [https://doi.org/10.1007/978-3-642-13190-5\\_26](https://doi.org/10.1007/978-3-642-13190-5_26)
21. Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8042, pp. 75–92. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-40041-4\\_5](https://doi.org/10.1007/978-3-642-40041-4_5)
22. Goldwasser, S., Micali, S.: Probabilistic encryption. *J. Comput. Syst. Sci.* **28**(2), 270–299 (1984). [https://doi.org/10.1016/0022-0000\(84\)90070-9](https://doi.org/10.1016/0022-0000(84)90070-9)
23. Hemenway, B., Libert, B., Ostrovsky, R., Vergnaud, D.: Lossy encryption: constructions from general assumptions and efficient selective opening chosen ciphertext security. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 70–88. Springer, Heidelberg (2011). [https://doi.org/10.1007/978-3-642-25385-0\\_4](https://doi.org/10.1007/978-3-642-25385-0_4)
24. Holenstein, T., Renner, R.: One-way secret-key agreement and applications to circuit polarization and immunization of public-key encryption. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 478–493. Springer, Heidelberg (2005). [https://doi.org/10.1007/11535218\\_29](https://doi.org/10.1007/11535218_29)
25. Horvitz, O., Katz, J.: Universally-composable two-party computation in two rounds. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 111–129. Springer, Heidelberg (2007). [https://doi.org/10.1007/978-3-540-74143-5\\_7](https://doi.org/10.1007/978-3-540-74143-5_7)
26. Ishai, Y., Kushilevitz, E., Ostrovsky, R.: Sufficient conditions for collision-resistant hashing. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 445–456. Springer, Heidelberg (2005). [https://doi.org/10.1007/978-3-540-30576-7\\_24](https://doi.org/10.1007/978-3-540-30576-7_24)
27. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Prabhakaran, M., Sahai, A.: Efficient non-interactive secure computation. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 406–425. Springer, Heidelberg (2011). [https://doi.org/10.1007/978-3-642-20465-4\\_23](https://doi.org/10.1007/978-3-642-20465-4_23)
28. Ishai, Y., Paskin, A.: Evaluating branching programs on encrypted data. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 575–594. Springer, Heidelberg (2007). [https://doi.org/10.1007/978-3-540-70936-7\\_31](https://doi.org/10.1007/978-3-540-70936-7_31)
29. Kushilevitz, E., Nisan, N.: *Communication Complexity*. Cambridge University Press (1997)
30. Kushilevitz, E., Ostrovsky, R.: Replication is NOT needed: SINGLE database, computationally-private information retrieval. In: 38th Annual Symposium on Foundations of Computer Science, FOCS 1997, Miami Beach, Florida, USA, 19–22 October 1997, pp. 364–373. IEEE Computer Society (1997). <https://doi.org/10.1109/SFCS.1997.646125>
31. Lipmaa, H.: An oblivious transfer protocol with log-squared communication. In: Zhou, J., Lopez, J., Deng, R.H., Bao, F. (eds.) ISC 2005. LNCS, vol. 3650, pp. 314–328. Springer, Heidelberg (2005). [https://doi.org/10.1007/11556992\\_23](https://doi.org/10.1007/11556992_23)
32. Liu, T., Vaikuntanathan, V.: On basing private information retrieval on NP-hardness. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016. LNCS, vol. 9562, pp. 372–386. Springer, Heidelberg (2016). [https://doi.org/10.1007/978-3-662-49096-9\\_16](https://doi.org/10.1007/978-3-662-49096-9_16)
33. Matsui, M.: Linear cryptanalysis method for DES cipher. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 386–397. Springer, Heidelberg (1994). [https://doi.org/10.1007/3-540-48285-7\\_33](https://doi.org/10.1007/3-540-48285-7_33)
34. 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). [https://doi.org/10.1007/3-540-48910-X\\_16](https://doi.org/10.1007/3-540-48910-X_16)

35. Peikert, C., Vaikuntanathan, V., Waters, B.: A framework for efficient and composable oblivious transfer. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 554–571. Springer, Heidelberg (2008). [https://doi.org/10.1007/978-3-540-85174-5\\_31](https://doi.org/10.1007/978-3-540-85174-5_31)
36. Rao, A., Yehudayoff, A.: Communication Complexity: and Applications. Cambridge University Press (2020)
37. Razborov, A.A.: On the distributional complexity of disjointness. In: Paterson, M.S. (ed.) ICALP 1990. LNCS, vol. 443, pp. 249–253. Springer, Heidelberg (1990). <https://doi.org/10.1007/BFb0032036>
38. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, 22–24 May 2005, pp. 84–93. ACM (2005). <https://doi.org/10.1145/1060590.1060603>
39. Rivest, R.L., Adleman, L., Dertouzos, M.L.: On data banks and privacy homomorphisms. Foundations of Secure Computation, pp. 169–179. Academia Press (1978)
40. Rothblum, R.: Homomorphic encryption: from private-key to public-key. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 219–234. Springer, Heidelberg (2011). [https://doi.org/10.1007/978-3-642-19571-6\\_14](https://doi.org/10.1007/978-3-642-19571-6_14)
41. Sander, T., Young, A.L., Yung, M.: Non-interactive cryptocomputing for  $nc^1$ . In: 40th Annual Symposium on Foundations of Computer Science, FOCS 1999, 17–18 October, 1999, New York, NY, USA, pp. 554–567. IEEE Computer Society (1999). <https://doi.org/10.1109/SFFCS.1999.814630>
42. Stern, J.P.: A new and efficient all-or-nothing disclosure of secrets protocol. In: Ohta, K., Pei, D. (eds.) ASIACRYPT 1998. LNCS, vol. 1514, pp. 357–371. Springer, Heidelberg (1998). [https://doi.org/10.1007/3-540-49649-1\\_28](https://doi.org/10.1007/3-540-49649-1_28)
43. Yao, A.C.C.: Some complexity questions related to distributive computing (preliminary report). In: Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing, pp. 209–213 (1979)
44. Yu, Yu., Zhang, J., Weng, J., Guo, C., Li, X.: Collision resistant hashing from sub-exponential learning parity with noise. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019. LNCS, vol. 11922, pp. 3–24. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-34621-8\\_1](https://doi.org/10.1007/978-3-030-34621-8_1)