Cryptographic Sensing

Yuval Ishai[1], Eyal Kushilevitz[1], Rafail Ostrovsky[2], and Amit Sahai[2]

[1] Technion, Israel
{yuvali,eyalk}@cs.technion.ac.il
[2] University of California, Los Angeles
{rafail,sahai}@cs.ucla.edu

Abstract. Is it possible to measure a physical object in a way that makes the measurement signals unintelligible to an external observer? Alternatively, can one learn a natural concept by using a contrived training set that makes the labeled examples useless without the line of thought that has led to their choice? We initiate a study of "cryptographic sensing" problems of this type, presenting definitions, positive and negative results, and directions for further research.

1 Introduction

The traditional goal of cryptography is to *design* cryptographic algorithms for well-defined tasks, such as public-key encryption. In this work we study the following question: when can we *embed* a cryptographic function in a function that was not designed for this purpose, say a function created by nature?

To make the question more concrete and illustrate a potential application scenario, consider the goal of observing a physical object in total darkness. Is it possible to design a flashlight, and a matching pair of glasses, such that the flashlight will only make the object visible to the owner of the glasses? Note that we are not attempting to hide the existence of the object and the flashlight. Our goal is to embed in the physical implementation of the flashlight a hidden secret (which is only explicitly found in the glasses), such that without knowing this secret it is *computationally infeasible* to make sense of the signals directed at and reflected from the object. The latter requirement should hold *even if the flashlight can be captured and analyzed completely* by an adversary, and even if the adversary can design its own pair of glasses based on this analysis.

A bit more rigorously and abstractly, we model the object being observed as a vector $x \in \mathcal{X}$, where $\mathcal{X} = \{0,1\}^n$ by default. The flashlight is modeled as a randomized measurement algorithm Sen that can carefully choose a sequence of measurement functions $f_1, f_2, \ldots$, where each f_i is taken from some fixed and publicly known class $\mathcal{F}$. For each function f_i, the algorithm Sen learns the value $a_i = f_i(x)$. The choice of the measurement sequence f_i may either be adaptive, in the sense that each f_i can depend on $a_1, \ldots, a_{i-1}$, or non-adaptive, in the sense that all f_i are chosen together. We would like the following two requirements to hold. First, given the randomness that was used to choose the measurement functions f_i, it is possible to efficiently decode the object x from the sequence (f_i, a_i). In the flashlight example, this decoding algorithm is performed by the glasses. Second, we would like the object x to remain "hidden" (in a sense that should be defined) from any polynomial-time (passive) external observer Adv

who can only view the measurements (f_i, a_i) without the randomness that was used to generate them. We refer to such an algorithm Sen as a *cryptographic sensing algorithm* for the measurement class $\mathcal{F}$.

For the purpose of obtaining better efficiency or stronger security, it will sometimes be useful to relax the above goal by settling for Sen learning only partial information $g(x)$ about x (e.g., a lower resolution version of an image x or some targeted portion of x), or by allowing Adv to learn partial information $\ell(x)$ about x (e.g., the brightness level of an image). As long as Sen has a meaningful advantage over Adv, we realize a non-trivial notion of cryptographic sensing. A different type of relaxation is to consider a *distributed* setting, where two or more instances of Sen can be executed. Here Adv only has access to a bounded number of these instances (say, one out of two) but the decoder has access to all of them. This is analogous to the type of security provided by secret sharing schemes or protocols for secure multiparty computation.

One can also consider a dual formulation of the problem in the language of computational learning theory. Here the object is a secret *concept*, namely a function $f \in \mathcal{F}$, and the goal of the (active, randomized) learning algorithm Sen is to come up with a training set $x_1, x_2, \ldots$ such that given the labeled examples $(x_i, f(x_i))$ it can efficiently learn some representation of f. The unusual requirement we make here is that without the “line of thought” that has led to the choice of the training set, it should be impossible for an efficient, passive Adv to learn f from the labeled examples. There are instances of the cryptographic sensing problem that are better motivated or are more naturally cast in this dual form; however, most instances considered in this work are more naturally cast in the original “sensing” framework.

If we could choose the class $\mathcal{F}$ at will, we could simply make it rich enough to directly implement an encryption of x via a standard public-key encryption scheme, say the RSA scheme. This is akin to allowing the flashlight to shoot a miniature robot at the object, where the robot physically senses the object and sends back an encrypted image using the flashlight's public encryption key. However, our goal here is to study the possibility of coping with *natural* classes $\mathcal{F}$, such as ones that can potentially be realized by a simple physical measurement process (in the sensing formulation) or that capture simple and/or realistic classes of concepts (in the learning formulation).

Other than the type of “sensing” applications illustrated by the cryptographic flashlight metaphor, it is not hard to imagine additional potential application scenarios. For instance, consider a drug company A that must outsource expensive experiments to a company B for the purpose of determining the chemical structure of some virus. Company A would like to deter employees of B who conduct the actual experiments from selling the results to a competing drug company C. Here too, our goal is not to hide the fact that a virus is being analyzed, but rather to render the “questions and answers” that must inevitably be obtained by B in the process of analysis useless to anyone but the company A that designed (and paid for) the experiments. A similar goal can apply to other measurement and learning scenarios such as conducting polls, training deep neural nets, and many more.

1.1 Our Contribution

In this work we initiate a study of cryptographic sensing by presenting definitions, some positive and negative results, and directions for further research.

Formalizing Security. We start by putting forward different notions of security for cryptographic sensing. The weakest notion is that of *one-way security*, which ensures that Adv has a negligible success of guessing x *exactly*, when x is picked at random. Using standard cryptographic terminology, a *non-adaptive* cryptographic sensing algorithm for $\mathcal{F}$ with one-way security is equivalent to an $\mathcal{F}$-computable injective trapdoor function, namely one that can be computed by concatenating functions from $\mathcal{F}$. One-way security is typically not very useful, since it only applies to a specific object distribution and even in this case it does not rule out revealing a big amount of partial information about the object.

A stronger and more useful notion is that of *entropic security*, requiring that any two object distributions that have high min-entropy cannot be distinguished by Adv. This intuitively means that the interaction does not help Adv distinguish between objects that were sufficiently unpredictable to start with. Using standard cryptographic terminology, a non-adaptive cryptographic sensing algorithm with entropic security for $\mathcal{F}$ can be viewed as an $\mathcal{F}$-computable one-time-secure *deterministic* public-key encryption scheme [46, 24, 13].

As in the case of non-cryptographic sensing (e.g., compressed sensing), it is often useful to settle for *lossy decoding*, where Sen outputs some useful partial information about x such as a projection of x to a subset of the coordinates or a compressive linear sketch of x from which an approximate version (e.g., a lower resolution image) can be recovered. Note that with lossy decoding, one-way security may become meaningless. However, entropic security is still as meaningful. Lossy decoding is motivated by the possibility of obtaining better efficiency (e.g., fewer measurements) and better security (e.g., entropic security with a lower entropy bound).

Finally, we consider a useful combination of entropic security and lossy decoding we refer to as *security with background noise*. Here we aim to completely hide the object x by masking it with "background noise" r, where measurements apply jointly to (x, r). (In the case of physical measurements, r can be taken from parts of the object that are considered irrelevant, or from nearby objects.) This is analogous to the role of randomness in semantically secure probabilistic encryption [27]. We distinguish between different types of security with background noise, depending on whether r is assumed to be random and whether it is independent of x.

Constructions and Negative Results. At a first glance, obtaining cryptographic sensing algorithms for *natural* classes $\mathcal{F}$ may seem hopeless. Indeed, such classes are expected to be either hard to learn (even without any security requirements) or alternatively admit *simple* learning algorithms in which case there is no hope to embed any cryptographic hardness, let alone the intricate structure of public-key cryptography.

However, a second thought reveals that this view may be too pessimistic. First, there is a rich line of work on *low complexity cryptography* (see Section 1.2), showing that sophisticated cryptographic primitives can be implemented in low complexity classes under well studied intractability assumptions. Second, an even richer line of work shows how to construct code-based [39, 2] or lattice-based [1, 44, 25] public-key encryption schemes in which encryption can be implemented by computing a *linear function* of the message and secret randomness over some finite ring $\mathbb{Z}_q$. In our language, these cryptosystems imply cryptographic sensing algorithms with $\mathbb{Z}_q$-linear measurements that achieve semantic security using random background noise. In fact, lattice-based *deterministic* public-key encryption schemes imply a similar result with entropic security [42, 19, 15] and without the need for background noise.

Let us pause to explain how linear functions, that are trivially "learnable" by using Gaussian elimination, can be a source for cryptographic hardness. The key feature that makes this possible is that the object x is restricted to be in $\{0,1\}^n$ (or, more generally, a vector of a small norm) whereas the linear combinations are taken over the larger domain $\mathbb{Z}_q^n$. This means that even though Adv can obtain an explicit description of affine space of objects in $\mathbb{Z}_q^n$ that are consistent with the labeled examples, it has no obvious way of making sense of this information. Indeed, the affine space is of exponential size, which makes it possible to hide inside it a low-norm object x that has sufficient entropy.

The simplicity of linear functions makes lattice-based cryptography an attractive venue for cryptographic sensing algorithms. However, even when restricting attention to linear measurements, there are several reasons why these off-the-shelf solutions from the literature are not completely satisfying.

First, they all inherently require *modular* linear measurements, modulo some finite integer $q \geq 2$. While the class of such measurements is very natural from a theory perspective, we are not aware of any realistic way of obtaining a direct physical implementation of such measurements. Note that it is crucial that no additional information except the output of the measurement is leaked. In contrast, linear combinations with small integer coefficients (or alternatively bounded-precision reals) can conceivably be realized without significant additional leakage. As a toy example, consider an implementation of a "flashlight" that shoots small balls or a spray of water at a metal board with holes. The amount of noise made by the impact (alternatively the amount of substance that bounces back) reveals a linear combination *over the integers* of the characteristic vector of the board (0 for hole, 1 for no hole) and the density vector of objects shot at it. One can easily imagine more sophisticated and scalable physical measurement processes of this type. Other disadvantages of off-the-shelf solutions is that their entropically secure variants have a poor quality of entropic security and require a large number of measurements.

We start by addressing the latter disadvantages. We show a simple cryptographic sensing algorithm that achieves a good quality of entropic security (i.e., with weak entropy requirements) and only uses a small number of linear measurements over $\mathbb{Z}_q$. The price we pay is that we settle for lossy decoding, revealing a sublinear number of linear combinations of the object x. In the context of natural objects (such as images), such compressive linear mappings are often almost as good as full recovery (see, e.g., [29] for a survey). Our algorithm is simple and intuitive, and builds on the same technique that underlies lattice-based encryption schemes such as Regev's cryptosystem [44]. The high level idea is to hide the "useful" linear combinations in a low-dimensional linear space by adding noise. Using a combination of the Learning With Errors assumption and the Leftover Hash Lemma, it is ensured that the linear space spanned by the measurements together with the measurement outcomes looks completely random to a computationally bounded Adv, assuming that the object has sufficient entropy.

Then, we use a simple generic transformation from cryptographic sensing algorithms that use linear measurements over $\mathbb{Z}_q$ to ones that use linear measurements over the integers. This transformation relies on background noise, effectively implementing mod-q reduction by adding secret random multiples of q that are harvested from the background noise. This transformation has two disadvantages: first, if the coefficients of the linear combinations are polynomially bounded, Adv gets an inverse polynomial distinguishing advantage (compared to the negligible advantage in the $\mathbb{Z}_q$ solutions). Second, the resulting algorithm relies on background noise and does not achieve entropic security. While we show that both disadvantages are in some sense inherent, there is still a lot of room for improving both the qualitative security guarantee and the quantitative parameters.

We conclude by presenting positive and negative results for other classes $\mathcal{F}$, beyond linear measurements. These results build heavily on previous results in the literature on computational learning theory or low-complexity cryptography. See Section 5 for details.

Can public-key encryption *really* be implemented by nature? As discussed above, cryptographic sensing for naturally occurring functions essentially requires public-key encryption (PKE) to be implemented by nature. This may seem inconceivable in light of the complexity and relative scarcity of known PKE candidates. However, as our results suggest, this view may be overly pessimistic for several reasons.

First, it ignores the extra degree of freedom one has by *encoding* the output of a standard PKE scheme. Indeed, even complex functions can be encoded by randomized functions (such as NC^0 functions [9]) for which *every individual output* is a very simple function of the input. The space of possible constructions of such randomized encoding schemes for functions is far from being well understood or even systematically explored.

A second degree of freedom that our constructions exploit is the ability of the sensing algorithm to pick an arbitrary, possibly contrived, distribution over the class of measurement functions. We heavily exploit this in our sensing algorithm that uses linear measurement functions with small integer coefficients. (As argued above, this class of measurements admits simple physical realizations.) In the dual learning formulation, this amounts to using a contrived training set or input distribution. The combination of a "contrived" input distribution with a "natural" function class might be just as powerful as the usual combination of "contrived" function class with a "natural" input distribution, which is commonly used in cryptography. For instance, for all we know, even DNF formulas can compute weak pseudo-random functions or public-key encryption schemes for contrived input distributions.

Finally, as we demonstrate in Section 5, relaxing the basic model to a distributed setting, which allows two or more separate interactions, breaks the PKE implication and opens the space to a much larger class of cryptographic sensing and learning schemes.

1.2 Related Work

A central theme of this work is that of using *simple* forms of cryptography, that one can actually implement via physical measurements. The study of simple forms of cryptography is not new and has already lead to rich and often surprising results. This study of low-complexity cryptography includes works on local one-way functions and other cryptographic primitives [26, 9] that led to the notion of randomized encoding (RE), works on low-degree polynomials [30, 8], linear-time functions [31], as well as similar results for arithmetic functions [32, 6].

By necessity, we will generally focus on less traditional notions of cryptographic security and correctness. For instance, we will largely eschew the traditional notion of probabilistic encryption with semantic security [27] in favor of entropic security [46, 24, 13] notions that have arisen primarily in the context of deterministic encryption. Moreover, we will consider relaxations of correctness, as well, inspired by compressed sensing (see, e.g., [33, 29]) where not the entire message is recovered.

Finally, the interaction between cryptography and learning has rich history. Valiant [48] already pointed out that if $\mathcal{F}$ contains a cryptographic pseudorandom function (PRF) then this makes the class $\mathcal{F}$ hard to learn, even given membership queries. Other cryptographic primitives like PKE, were used to base some more advanced hardness results in learning theory, e.g., [35, 4, 36] (beyond hardness results for so-called "proper" learning [43]). See [23] for a more recent work in this direction. Concretely, the work of [4] may seem as an obstacle for cryptographic-sensing of sufficiently rich classes $\mathcal{F}$ (such as CNF formulae) that allow embedding of signature verification. For such classes, they prove that if $\mathcal{F}$ can be leaned from random examples and membership queries (MQs) then $\mathcal{F}$ can also be leaned from random examples and *without* membership queries, which seems to indicate that whatever the sensing algorithm can learn the adversary will be able to learn as well. This however is not the case as, in the construction of [4], for learning $f \in \mathcal{F}$ wrt distribution D and without MQs, the learner invokes a learning algorithm with MQs that learns a *different* (but related) function $f' \in \mathcal{F}$ wrt a related distribution D'. This is not possible in the physical setting of cryptographic sensing where f, D are chosen by nature and the sensing algorithm and the adversary both have access to them.

In the reverse direction, hard learning problems were proposed as a source for cryptographic assumptions [16] where the LPN and LWE assumptions (proposed by [16, 44] respectively) serve, due to their convenient structure and simplicity, as some of the most useful assumptions for cryptographic constructions. There is currently a large body of work on adversarial machine learning (e.g, [22, 38] and references therein); these works are mainly concerned with the correctness of the learning process in the presence of adversaries that harm the samples (e.g., by changing examples and/or their labels). In a very different direction, [34] initiated the study of private learning, whose goal is to protect the privacy of individuals whose sensitive inputs are used for learning.

1.3 Future Directions

Our work leaves many directions for future research. Which natural classes admit cryptographic sensing algorithms? We note that not much attention has been spent on finding explicit “hard distributions” for classes that are learnable with membership queries, but for which PAC learning algorithms are unknown. This includes even very simple classes such as DNF formulas. Hard input distributions for these classes can serve as a starting point for designing a cryptographic sensing algorithm. Another direction that we haven’t explored at all is potential applications in the context of practical machine learning algorithms.

On the physical side: which simple functions of an object can be measured (say, using radar technologies or particle physics) without significant additional leakage? We described specific toy experiments for realizing linear measurements with small non-negative integer coefficients. Is there a direct way to measure mod-q linear combinations using quantum measurements or classical wave interference? Is there a good algorithmic way to cope with the type of additional leakage that can be expected from physical measurements?

To conclude, while the feasibility of implementing a cryptographic flashlight “in the wild” is left open, we do not see any fundamental barriers to making this idea applicable for real-world sensing and learning problems. Our results leave much room for further quantitative and qualitative improvements that can help make this happen. Alternatively, the question of cryptographic sensing can help motivate a rich line of theoretical questions that explore new kinds of interaction between cryptography and computational learning theory.

2 Preliminaries

In this section we recall some standard definitions and facts that will be useful for formalizing our security notions and analyzing the constructions.

Definition 1 (Statistical distance) *The* statistical distance *between distributions X and Y, denoted* $\mathrm{SD}(X, Y)$*, is defined as the maximum, over all functions A, of the* distinguishing advantage $|\Pr[A(X) = 1] - \Pr[A(Y) = 1]|$.

Definition 2 (Min-Entropy) *We say that a random variable X over a set S has* min-entropy k*, and denote $H_\infty(X) = k$, if* $\max_{s \in S} \Pr[X = s] = 2^{-k}$.

We will use the following standard Leftover Hash Lemma (LHL).

Lemma 3 (Leftover Hash Lemma [28]). Let $H = \{h\}$ be a family of pairwise-independent hash functions $h : \mathcal{D} \to \mathcal{R}$. Let X be a distribution over $\mathcal{D}$ with $H_\infty(X) \geq k$, where $k \geq \log|\mathcal{R}| + 2\log(1/\epsilon)$. Then, the distribution $(h, h(x))$ with $h \in_R H$ and x selected according to X is ϵ-close to (h, r), with $h \in_R H$ and $r \in_R \mathcal{R}$.

2.1 The LWE Assumption

We rely on well studied decisional variants of the Learning with Errors (LWE) assumption [44]. This assumption says that a random noisy codeword in a publicly known random linear code is pseudo-random. More precisely, the distribution $(M, Ms + e)$, for a random matrix $M \in_R \mathbb{Z}_q^{n \times m}$, secret vector $s \in_R \mathbb{Z}_q^m$ and appropriately chosen noise distribution $e \in \mathbb{Z}_q^n$ is computationally indistinguishable from the distribution (M, u), where u is uniformly distributed over $\mathbb{Z}_q^n$ independently of M. The noise distribution for LWE is χ^n for some distribution χ over $\mathbb{Z}_q$, which is typically a discrete Gaussian. For simplicity, it is convenient to replace the Gaussian distribution with a uniform distribution over an interval $[0, b]$, where $b = q^{\Omega(1)}$. When q is super-polynomial in λ, security with such an "interval noise" reduces to security with Gaussian noise of standard deviation $\approx b$. While no such reduction is known in the regime of polynomially large q (which is more relevant to our work), this alternative form of the LWE assumption resists known attacks.

Definition 4 (Learning With Errors) *Let λ be a security parameter. For $m = m(\lambda)$, $n = n(\lambda)$, $q = q(\lambda)$, $b = b(\lambda)$, $t = t(\lambda)$, and $\epsilon = \epsilon(\lambda)$, we say that the* decisional learning with errors *problem (with interval noise)* $\mathsf{LWE}_{m,n,q,b}$ *is* (t, ϵ)-hard *if for all sufficiently large λ, every circuit of size $t = t(\lambda)$ has at most an $\epsilon = \epsilon(\lambda)$ advantage in distinguishing between the distributions $(M, Ms + e)$ and (M, u), where $M \xleftarrow{R} \mathbb{Z}_q^{n \times m}$, $s \xleftarrow{R} \mathbb{Z}_q^m$, e is uniformly distributed in $[0, b]^n$, and $u \xleftarrow{R} \mathbb{Z}_q^n$. We say that LWE holds with parameters m, n, q, b, if* $\mathsf{LWE}_{m,n,q,b}$ *is $(t = p(\lambda), \epsilon = 1/p(\lambda))$-hard for every polynomial $p(\lambda)$.*

Our typical choice of parameters is $m = \lambda$, $n \leq \mathrm{poly}(m)$, $q \approx n^2$, and $b \approx \sqrt{m}$; however, smaller values of q and b can be used for better efficiency. See [40] and references therein for choices of LWE parameters that resist known attacks, including ones that are provably secure under worst-case hardness assumptions for integer lattices. We note that one could alternatively settle for sampling the secret s from the the same distribution as the noise instead of the uniform distribution [7]. This optimization can improve the concrete efficiency of our LWE-based constructions.

3 Defining Cryptographic Sensing

In this section we formalize our notion of cryptographic sensing. We start with the default "sensing" formulation and then describe how to modify it to get the dual "learning" formulation.

Function classes. A *function class* $\mathcal{F}$ is defined by a polynomial-time algorithm that given a description $\hat{f}$ of a finite function f and an input x for f outputs $f(x)$. (We will often abuse notation and identify functions and other objects with their representations.) We assume that $\hat{f}$ includes a description of the input domain $\mathcal{X}_f$ and that $\mathcal{F}$ outputs $\perp$ when the input $(\hat{f}, x)$ is not of the expected form, namely when $\hat{f}$ is not a valid function description or when $x \notin \mathcal{X}_f$. We let $\mathcal{X}(\mathcal{F})$ denote the set of (descriptions of) valid input domains for $\mathcal{F}$ and $\mathcal{F}_{\mathcal{X}}$ denote the set of function descriptions $\hat{f}$ with input domain $\mathcal{X}$. The set $\mathcal{F}_{\mathcal{X}}$ defines the allowable *measurement functions* that can apply to a hidden object $x \in \mathcal{X}$. The function class $\mathcal{F}$ also assigns a *cost* measure to every function description $\hat{f}$. For instance, if $\mathcal{F}$ is the class of DNF formulas then a natural cost of $\hat{f}$ is the number of clauses and if $\mathcal{F}$ is the class of linear functions with non-negative integer coefficients then a natural cost is the sum of all coefficients.

Cryptographic sensing: syntax. A *cryptographic sensing algorithm for* $\mathcal{F}$ is a PPT algorithm Sen with oracle access to $\mathcal{F}$ that, given a security parameter 1^λ and description of an input domain $\mathcal{X} \in \mathcal{X}(\mathcal{F})$, proceeds as follows. It starts by randomly generating a secret decoding key sk. (If concrete efficiency is not a concern, one can let sk include all random coins of Sen.) It then interacts with $\mathcal{F}$, feeding it with measurement functions $\hat{f}_i \in \mathcal{F}_{\mathcal{X}}$, and receiving the outcomes $a_i = f_i(x)$ on some fixed object $x \in \mathcal{X}$ unknown to Sen. We will mostly consider *non-adaptive* sensing algorithms Sen in which all measurements $\hat{f}_i$ are chosen simultaneously before querying $\mathcal{F}$. In this case, we will sometimes consider the concatenation of all f_i as a single function f taken from the multi-output extension of $\mathcal{F}$. Once it is done querying, Sen uses sk and the interaction transcript $\mathcal{I} = ((\hat{f}_1, a_1), \ldots, (\hat{f}_m, a_m))$ to produce a guess for x.

Correctness. The default correctness requirement, which will later be relaxed, is that for every efficient non-uniform adversary that on input 1^λ picks an input domain $\mathcal{X} \in \mathcal{X}(\mathcal{F})$ and an object $x \in \mathcal{X}$, the interaction of $\mathsf{Sen}(1^\lambda, \mathcal{X})$ with $\mathcal{F}_{\mathcal{X}}$ on object x results in Sen outputting the correct value of x except with $\mathrm{neg}(\lambda)$ probability.

One-way security. The minimal security requirement we consider is *one-wayness*. Since it is not always natural to consider a uniform distribution over objects (let alone over *functions* in the learning formulation), we allow an arbitrary efficiently samplable distribution. Concretely, we say that Sen is one-way secure with respect to $\mathcal{F}$ if there is a PPT object sampling algorithm S such that every efficient non-uniform adversary Adv succeeds in the following game with $\mathrm{neg}(\lambda)$ probability. First, $S(1^\lambda)$ outputs a challenge input domain $\mathcal{X} \in \mathcal{X}(\mathcal{F})$ and an object $x \in \mathcal{X}$. Then, $\mathsf{Sen}(1^\lambda, \mathcal{X})$ interacts with $\mathcal{F}_\mathcal{X}$ on object x, resulting in an interaction transcript $\mathcal{I} = ((\hat{f}_1, a_1), \ldots, (\hat{f}_m, a_m))$. Finally, $\mathsf{Adv}(\mathcal{X}, \mathcal{I})$ outputs a guess for x. We say that Adv succeeds if its guess is correct.

Using standard cryptographic terminology, a *non-adaptive* cryptographic sensing algorithm with one-way security for $\mathcal{F}$ is equivalent (up to the choice of input distribution) to an $\mathcal{F}$-computable injective trapdoor function, namely one that can be computed by concatenating functions from $\mathcal{F}$. One-way security is typically not very useful, since it only applies to a specific object distribution and even in this case it does not rule out revealing a big amount of partial information about the object. Below we define several stronger notions, analogously to different notions of security for (one-time) encryption in the cryptographic literature.

Entropic security. Entropic security is in a sense the best possible notion of security for deterministic encryption. It requires that any two object distributions that have high min-entropy cannot be distinguished by Adv. This intuitively means that the interaction does not help Adv distinguish between objects that were sufficiently unpredictable to start with. Formally, let $k : \mathbb{N} \times \mathcal{X}(\mathcal{F}) \to \mathbb{R}$ be an entropy bound function, specifying a lower bound on object entropy as a function of the security parameter λ and object domain $\mathcal{X}$. For $\epsilon = \epsilon(\lambda)$ we say that Sen is (k, ϵ)-*entropically secure* if every efficient non-uniform adversary Adv succeeds in the following game with at most $1/2 + \epsilon(\lambda)$ probability for all sufficiently large λ. First, $\mathsf{Adv}(1^\lambda)$ outputs an input domain $\mathcal{X} \in \mathcal{X}(\mathcal{F})$ and a pair of circuits describing input distributions X_0, X_1 over $\mathcal{X}$ with $H_\infty(X_\sigma) \geq k$ for $\sigma = 0, 1$. Then, a challenger picks a random bit $\sigma \in \{0, 1\}$ and lets $\mathsf{Sen}(1^\lambda, \mathcal{X})$ interact with $\mathcal{F}_\mathcal{X}$ on an object x sampled from X_σ. This results in an interaction transcript $\mathcal{I} = ((\hat{f}_1, a_1), \ldots, (\hat{f}_m, a_m))$. Finally, $\mathsf{Adv}(\mathcal{I})$ outputs a guess for σ. We say that Adv succeeds if its guess is correct. When ϵ is omitted we assume it is negligible; however, some of our results inherently require ϵ to be non-negligible.

To gain more flexibility, it can be convenient to give an entropy bound k as an additional input for Sen and modify the above definition accordingly (allowing Sen to declare failure in case k is too low; for instance, when $k = O(\log(\lambda))$ a brute-force search attack is possible). Using standard cryptographic terminology, a non-adaptive cryptographic sensing algorithm with entropic security for $\mathcal{F}$ can be viewed as an $\mathcal{F}$-computable one-time-secure *deterministic* public-key encryption scheme [46, 24, 13].

Lossy decoding. A useful relaxation of the above correctness requirement settles for *lossy decoding*, where Sen outputs some useful partial information about x such as a projection of x to a subset of the coordinates or a compressive linear sketch of x from which an approximate version (e.g., a lower resolution image) can be recovered. We formalize this by introducing a *target function class* $\mathcal{G}$ with the same input domains as $\mathcal{F}$ (i.e., $\mathcal{X}(\mathcal{G}) = \mathcal{X}(\mathcal{F})$) and adding to the inputs of Sen a description $\hat{g}$ of a target function $g : \mathcal{X} \to \mathcal{Z}$. The correctness requirement is changed in a natural way, requiring that $\mathsf{Sen}(1^\lambda, \mathcal{X}, \hat{g})$ correctly output $g(x)$. In the definition of entropic security, the entropy bound k is allowed to also depend on g (where typically k needs to grow with the output size of g). Note that with lossy decoding, one-way security may become meaningless. However, entropic security is still as meaningful.

Allowing background noise. A useful special case of lossy decoding is a projection to a *fixed* set of coordinates, where the other coordinates are viewed as background noise whose entropy can be exploited to protect the target output. In this case we will view the measurements of Sen as applying to (x, r), where x is the target object and r is the background noise, and require Sen to only output x. One can consider three notions of security with background noise. The strongest, referred to as security with *correlated background noise* does not assume independence between x and r and only requires entropic security when the *joint entropy* is at least k. The second, referred to as security with *independent background noise*, requires that x remain completely hidden if the background noise is independent and has high min-entropy. This is formalized as in the definition of entropic security, except that the distributions X_0 and X_1 are of the form (x_0, R) and (x_1, R) for $x_0, x_1 \in \mathcal{X}$ and an adversarially chosen R such that $H_\infty(R) \geq k$. Finally, the third and weakest notion, referred to as security with *random background noise*, is similar to the above except that the noise is picked from some specified noise distribution (uniform by default).

The weakest variant of security with background noise corresponds to the usual notion of semantically secure probabilistic encryption [27]. The strongest is equivalent to (one-time) indistinguishability under a *chosen distribution attack*, as defined in [14].

A dual learning formulation. In the above, we assumed that Sen tries to recover a secret object $x \in \mathcal{X}$ using a sequence of measurement functions f_i. In the setting of computational learning theory [48], one considers the dual goal of learning a secret *concept* $\hat{f} \in \mathcal{F}$ by evaluating it on a sequence of inputs x_i. The above definitions can be adapted in a natural way to this dual formulation. However, some changes should be made. First, the role of the object domain $\mathcal{X}$, which is given as input to Sen, is replaced by a sub-class of concepts in $\mathcal{F}$ from which the target concept is picked. For instance, if $\mathcal{F}$ is the class of DNF formulas, this sub-class can include all formulas with a fixed number of inputs n, or alternatively formulas with n inputs and ℓ clauses. Second, since $\mathcal{F}$ may define many equivalent representations $\hat{f}$ for the same concept f, we define the entropic security requirements semantically, namely with respect to the functions rather than their representations. Finally, our correctness requirements can accommodate relaxed notions of correctness from the machine learning literature. For instance, we can allow approximate correctness as in the PAC model (except that we need to additionally allow membership queries), and we can consider improper learning, namely allow Sen to output a general circuit representation of the target concept or its approximation.

The distributed setting. Finally, we will consider a distributed relaxation of the above notions, where Sen may be involved in $d \geq 2$ separate interactions, producing transcripts $\mathcal{I}_1, \ldots, \mathcal{I}_d$. The output can be decoded by Sen given all transcripts, but only a bounded number t of these transcripts is available to Adv. In the context of the drug company example from the Introduction, this corresponds to distributing the experiments among d companies $B_1, \ldots, B_d$, where security of A is only guaranteed as long as no more than t companies B_i reveal their information to C.

4 Cryptographic Sensing with Linear Measurements

In this section we describe simple cryptographic sensing algorithms that use different types of linear measurement functions: linear functions over $\mathbb{Z}_q$ and linear functions over the integers.

4.1 Linear Measurements over $\mathbb{Z}_q$

Code-based public-key encryption schemes presented by McEliece [39] and by Alekhnovich [2] imply cryptographic sensing algorithms with linear measurements over $\mathbb{Z}_2$ that require uniformly random background noise and a large number of measurements. These constructions can in fact be generalized to apply over any finite field. Lattice-based encryption schemes such as Ajtai-Dwork [1], Regev [44] and GPV [25] imply similar algorithms with linear measurements over $\mathbb{Z}_q$, where q grows with the object length n. These lattice-based constructions have the additional benefit of provable security under well-studied worst-case hardness assumptions; however they still require a random background noise and a large number of measurements.

Targeting the stronger security notion of entropic security, one could obtain lattice-based cryptographic sensing algorithms with $\mathbb{Z}_q$-linear measurements by using known lattice-based constructions of deterministic encryption schemes [19], which in turn are based on constructions of lossy injective trapdoor functions [42, 15]. However, these constructions require a large number of measurements, and only tolerate a constant entropy rate.

In the following we present an LWE-based *lossy* cryptographic sensing algorithm that uses $\mathbb{Z}_q$-linear measurements and achieves entropic security, where both the entropy bound and the number of measurements are comparable to the length of the lossy output $g(x)$, independently of the length of the measured object x.

We first assume for simplicity that the object is $x \in \mathcal{X}_n = \{0,1\}^n$ and the target function class $\mathcal{G}$ is the class of mod-2 linear mappings with output length $t < n$. This is already useful for obtaining many natural approximations of x [33]. We then generalize the algorithm to the case where x is a vector of bounded-size integers and $\mathcal{G}$ includes compressive linear mappings with bounded integer coefficients. This generalization allows for a wider range of useful approximations, via compressed sensing and other linear sketching techniques (cf. [29]).

The algorithm uses the standard approach of lattice-based cryptosystems, in particular Regev's cryptosystem [44]. The high level idea is as follows. If the object x has min-entropy k, then (by Lemma 3) revealing up to $\approx k/\log q$ publicly known *random* $\mathbb{Z}_q$-linear combinations of the entries of x gives essentially no information about x. However, if we could choose special linear combinations, say ones in which each coefficient is either close to 0 or to $q/2$ (where $q \gg n$), then we could learn parities of subsets of the bits of x. Assuming LWE, we can hide such special linear combinations in the span of a small number of random-looking linear combinations. We formalize this idea below. Note that below we assume LWE for uniform noise. However, this is strictly for simplicity of exposition; all our results also hold assuming LWE with discrete Gaussian noise.

Decoding a single parity. We start by considering the case $t = 1$, namely the target function g computes $\langle y, x\rangle \bmod 2$ for $y \in \{0,1\}^n$, and then generalize to $t > 1$. The class of measurement functions $\mathcal{F}$ includes all linear functions mod q. That is, each measurement f_i is represented by $\ell \in \mathbb{Z}_q^n$ and returns $\langle \ell, x\rangle \bmod q$. We will also have a dimension parameter m and noise parameter b where the choice of n, m, q, b (all as functions of a security parameter λ) satisfies the LWE assumption with interval noise. Furthermore, we require that q be at most polynomial in n and that $q > 4nb$. See Section 2.1 for possible choices of parameters.

Algorithm $\mathsf{Sen}(1^\lambda, 1^n, y \in \{0,1\}^n)$*:*

1. Pick $A \in_R \mathbb{Z}_q^{m\times n}$ and set $z = s^T A + e + \lfloor q/2 \rfloor \cdot y$ for "LWE secret" $s \in_R \mathbb{Z}_q^m$ (which serves as the secret key sk) and "noise" vector $e \in_R [0,b]^n$, as in the LWE assumption. (Note that since $s^T A + e$ is pseudorandom given A, then so is z.)
2. Make the $m+1$ measurements corresponding to the m rows of A, as well as z. Get in response $m+1$ values that are viewed as $v_1 = Ax \in \mathbb{Z}_q^m$ and $v_2 = z \cdot x$ (where all arithmetic is over $\mathbb{Z}_q$).
3. Use the secret key s to compute $w = v_2 - s^T v_1 = z \cdot x - s^T Ax = (s^T A + e + \lfloor q/2 \rfloor \cdot y) \cdot x - s^T Ax = e \cdot x + \lfloor q/2 \rfloor \cdot y \cdot x$ from which $g(x) = \langle y, x\rangle \bmod 2$ is decoded: 0 if w is closer to 0 than to $\lfloor q/2 \rfloor$, and 1 otherwise.

The correctness of Sen follows from its description and from the choice of parameters: we have $0 \le e \cdot x \le nb$, and the parity of $y \cdot x$ determines whether we add $\lfloor q/2 \rfloor$ an even number of times, implying that $w \approx 0 \bmod q$ or an odd number of times, implying that $w \approx \lfloor q/2 \rfloor \bmod q$. Hence, $\langle y, x \rangle \bmod 2$ is correctly decoded with probability 1.

Let us argue that Sen is entropically secure for entropy bound $k \approx m \log q$. Note that the view of the adversary Adv consists of the queries of Algorithm Sen, as well as the answers, namely A, z, v_1, v_2. Our goal is to argue that if the entropy condition is met, this view is indistinguishable from a random tuple in $\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^n \times \mathbb{Z}_q^m \times \mathbb{Z}_q$. The proof will be based on the LWE assumption and the leftover hash lemma (see Lemma 3). Note that if we take the collection of matrices $A \in \mathbb{Z}_q^{m \times n}$ and consider the functions $h_A(x) = Ax$ (from $\mathbb{Z}_q^n$ to $\mathbb{Z}_q^m$), then the family $H = \{h_A\}$ is indeed known to be pairwise-independent hash family.

Security analysis: Let X be a distribution of objects with $H_\infty(X) = k$, for $k \geq (m+1)\log q + 2\log(1/\epsilon)$. Let $\mathcal{I} = (A, z, v_1, v_2)$ be the distribution of interactions, as generated by Sen, when interacting with an object x drawn from X. Let $\mathcal{I}' = (A, z, v_1, v_2)$ be the same distribution, except that now z is selected by $z \in_R \mathbb{Z}_q^n$. By the LWE assumption, $\mathcal{I} \approx_c \mathcal{I}'$. Let B be the $(m+1) \times n$ matrix obtained by placing A on top of z and let v be the $(m+1)$-vector obtained by concatenating v_2 to v_1. Observe that $v = Bx$. By the leftover hash lemma (with $\log |\mathcal{R}| = (m+1)\log q$), since X has sufficient min-entropy then the pair $(B, v = Bx)$, for a random B, is ϵ-close to a random pair (B, v) (of the corresponding lengths). This means that $\mathcal{I}'$ is indistinguishable from a distribution $\mathcal{I}'' = (A, z, v_1, v_2)$ consisting of randomly selected elements from $\mathbb{Z}_q$ of the corresponding length. This analysis yields the following theorem.

Theorem 5. *Suppose n, m, q, b are chosen such that $\mathsf{LWE}_{m,n,q,b}$ holds, $q > 4nb$, and $k \geq m \log q + \lambda$. Then Sen is a k-entropically secure cryptographic sensing algorithm for decoding a single parity of $x \in \{0,1\}^n$ using $m+1$ linear measurements over $\mathbb{Z}_q$.*

Extensions. We now extend the previous algorithm in a few simple ways. First, we observe that, for any "small" c, we can modify Sen to learn $\langle y, x \rangle \bmod c$ (rather than only for $c = 2$, as above). This is simply done by computing $z = s^T A + e + \lfloor q/c \rfloor \cdot y$. Again, each coordinate j where $x_j y_j = 1$ will contribute $\approx \lfloor q/c \rfloor$ to the value $v_2 - s^T v_1$. As long as q is sufficiently large (say, $q > 4cnb$) the noise does not prevent the algorithm from recovering $\langle y, x \rangle \bmod c$. In fact, in this case we can let y be any vector in $[0, c-1]^n$ and the algorithm still works, as is. Moreover, it also works when x is not a binary vector but is rather an integer-valued vector from $[0, d]^n$, provided that $q > 4cdnb$.

Next, we consider the case where the sensing algorithm wishes to learn not only a single linear combination $\langle y, x \rangle \bmod 2$, but rather a few of those; namely, for $y^1, \ldots, y^t$, where each y^j is in $\{0,1\}^n$, the algorithm needs to learn all of $\langle y^1, x \rangle \bmod 2, \ldots, \langle y^t, x \rangle \bmod 2$ (this can also naturally be combined with the previous extensions, to allow learning linear combinations mod c of $x \in [0, d]^n$). The first approach that comes to mind is to independently pick queries

$(A^1, z^1), \ldots, (A^t, z^t)$, as in the basic algorithm Sen. While this in principle works, it rapidly "consumes" the entropy of x (as Adv gets to see $m+1$ linear combinations per each y^j). Instead, we will pick a single matrix $A \in_R \mathbb{Z}_q^{m \times n}$ and t vectors $z^1, \ldots, z^t$, as above (namely, for each $j \in [t]$, we set $z^j = s^j A + e^j + \lfloor q/2 \rfloor \cdot y^j$, for "secret" $s^j \in_R [b]^m$). The correctness remains unchanged. As for security, as long as X has entropy $k \geq (m+t)\log q + 2\log(1/\epsilon)$, then a similar argument holds. Namely, $A, z^1, \ldots, z^t$ is still pseudorandom, by repeated application of the LWE assumption, and then the leftover hash lemma is applied where our hash function has output of length $m+t$.

Applying the above extensions to Sen, we get a general algorithm Sen' for decoding compressive linear mappings of x over the integers using $\mathbb{Z}_q$-linear measurements.

Theorem 6. *Suppose n, m, q, b, t, c are chosen such that $\mathsf{LWE}_{m,n,q,b}$ holds, $q > 4c^3nb$, and $k \geq (m+t)\log q + \lambda$. Then Sen' is a k-entropically secure cryptographic sensing algorithm for decoding Gx, where $x \in [0,c]^n$ and G is a $t \times n$ integer matrix with entries in $[0,c]$, using $m+t$ linear measurements over $\mathbb{Z}_q$.*

Note that unlike solutions based on deterministic encryption, Theorem 6 cannot be used to obtain full decoding of x even when x is uniformly random. Indeed, this would require choosing t so that the entropy requirement becomes impossible to meet. However, for the case of lossy decoding Theorem 6 gives near-optimal complexity.

4.2 Linear Measurements over the Integers

The LWE-based solution inherently makes use of linear measurements over $\mathbb{Z}_q$. From a physical realization perspective, it is much more desirable to use linear measurements over the *integers* (or reals), since it is not clear how to design a simple physical measurement process that reveals *only* a mod-q linear combination. However, applying the previous construction directly over the integers would render it insecure, since modular reduction is crucial for ruling out efficient real-valued approximation and decoding techniques [20].

We start by proving some inherent limitations on the type of security that can be achieved using linear measurements over the integers, and then present a positive result.

Can't make ϵ negligible. Our first negative result says that even if x is a single bit and we settle for semantic security with random background noise, we cannot obtain a negligible distinguishing advantage with polynomial-size linear measurement coefficients. This negative result is based on the following lemma, which says that for a random variable Z over integers in a small range, the statistical distance between Z and $Z+1$ is noticeable.

Lemma 7. Let Z be distributed over $[0, c-1]$. Then $\mathrm{SD}(Z, Z+1) \geq 1/c$.

Proof: Let $Z_i = Z + i$. Since Z and $Z + c$ have disjoint supports, we have $\mathrm{SD}(Z_0, Z_c) = \mathrm{SD}(Z, Z + c) = 1$. By the triangle inequality, there must exist $0 \leq j < c$ such $\mathrm{SD}(Z_j, Z_{j+1}) \geq 1/c$. The lemma follows by observing that $\mathrm{SD}(Z, Z + 1) = \mathrm{SD}(Z_j, Z_{j+1})$. □

It follows from the above lemma that for any vector of linear measurement coefficients $\ell \in \mathbb{N}^{m+1}$ with $\ell_1 > 0$ and $\ell_i \in [0, c-1]$, and for a random background noise $r \in_R \{0, 1\}^n$ we have $\mathrm{SD}(\langle \ell, (0, r)\rangle, \langle \ell, (1, r)\rangle) \geq 1/(cm)$. Moreover, since the distributions have polynomial-size support, a statistical distinguisher implies (non-uniform) efficient distinguisher. This rules out security with negligible distinguishing advantage $\epsilon = \mathrm{neg}(\lambda)$, polynomial background noise $m = \mathrm{poly}(\lambda)$, and $\mathrm{poly}(\lambda)$-bounded coefficients.

We note that this negative result does *not* apply to our minimal notion of one-way security. Indeed, one-way security can be achieved with polynomial-size coefficients by dividing x into λ disjoint blocks and applying the positive result presented below for each block. This exploits the fact that one-way security can be amplified via independent repetition.

Can't get entropic 1/poly-security. Next, we show that our positive result for entropic security with mod-q linear measurements (cf. Theorem 6) cannot be achieved over the integers, even if one settles for 1/poly distinguishing advantage, and even if there is no bound on the size of the integers. The intuition is that when there is no background noise to mask the object x, there is a noticeable difference between a "bright" object and a "dark" object. Concretely, consider the case of entropy bound $k = n/3$, and let X_0 be a distribution over $\{0, 1\}^n$ in which a random set of $n/3$ bits are picked at random and the rest are set to 0 (a bright object), and X_1 is a similar except that the other bits are set to 1 (a dark object). Then, for any nonzero $\ell \in \mathbb{N}^n$, a distinguisher that tests whether its input is bigger than $\sum_{i=1}^{n} \ell_i/2$ distinguishes between $\langle \ell, X_0\rangle$ and $\langle \ell, X_1\rangle$ with constant advantage. We leave open the question of obtaining security up to leakage of brightness, namely obtaining an entropic secure solution (say, with poly-bounded integer coefficients and constant fractional entropy bound) in which the view of Adv can be simulated given $\sum x_i$ with simulation error that vanishes with n.

A positive result. Our positive result complements the first negative result above by showing that semantic security with random background noise is indeed achievable with $\epsilon \geq 1/\mathrm{poly}(\lambda)$ by only using polynomial-size non-negative integer coefficients. More generally, we give a simple method for compiling any solution that uses linear measurements over $\mathbb{Z}_q$ into one that uses linear measurements over the integers, at the price of relying on random background noise and settling for an inverse polynomial error. More precisely, the cost of the linear measurements (namely, the magnitude of the coefficients) grows polynomially with n/ϵ.

The high level idea is to perform the $\mathbb{Z}_q$ measurements over the integers, and effectively achieve modular reduction by adding (over the integers) a large random multiple of q. (The previous negative result suggests that this is in some sense inherent.) The randomness used by this reduction is taken from the background noise. Concretely, given a bound $\mu = 2^c$ on the coefficients, we add to each original measurement a weighted sum of the form $\sum_{i=1}^{c}(2^i q) \cdot r_i$ where each measurement uses a disjoint set of c background points. Note that this effectively means that we add a random multiple $\beta \cdot q$ for $\beta \in_R [0, \mu - 1]$.

We turn to analyze the correctness and security of the above transformation. Decoding in the integer case can proceed as in the mod-q case, reducing the integer measurement values modulo q. This is not affected by adding multiples of q, hence correctness is maintained. The security analysis relies on the following standard lemma (cf. [12]), showing that if we add βq to a value from a bounded range $[0, B]$ and β is uniform in $[0, \mu - 1]$ (for μ sufficiently large, depending on B and q), then little is revealed beyond mod q.

Lemma 8. Let $\alpha_1, \alpha_2 \in [0, B]$ be two integers such that $\alpha_1 \equiv \alpha_2 \bmod q$. Consider the two distributions Y_1, Y_2 where Y_i is obtained by $\alpha_i + \beta q$, for $\beta \in_R [0, \mu - 1]$. Then, $\mathrm{SD}(Y_1, Y_2) \leq \frac{B}{q\mu}$.

Note that when allowing a random background noise, the entropic security with lossy decoding of Theorem 6 implies semantic security with full decoding by applying the algorithm of Theorem 6 to the concatenation (x, r) and decoding only the x portion. Applying the above transformation, we get the following integer analogue.

Theorem 9. *Suppose* $\mathsf{LWE}_{m,n,q,b}$ *holds for* $m = \lambda$, $n = m^d$ *(for some constant* $d > 2$*),* $b = \sqrt{q}$ *and* $q = \Theta(c^3 n^2)$ *(for some positive integer* $c = c(\lambda)$*). Then there is a semantically* ϵ*-secure cryptographic sensing algorithm with random background noise for decoding* Gx*, where* $x \in [0, c]^n$ *and* G *is a* $t \times n$ *integer matrix with entries in* $[0, c]$*, using* $m + t$ *linear measurements with non-negative integer coefficients, where the cost of each measurement is* $\mathrm{poly}(n, c, 1/\epsilon)$*.*

Studying the extent to which the random background noise assumption can be relaxed, as well as a more refined study of the achievable tradeoffs between the parameters, are left for future work.

5 Beyond Linear Measurements

In this section, we briefly discuss other classes of measurements, beyond linear functions. Here we will typically use the learning formulation of cryptographic sensing (see Introduction and Section 3). We give examples for positive and negative results that follow quite easily from the literature, as well as some directions for further research.

5.1 Negative Results for Simple Classes via Occam's Razor

Our first observation in this section is that classes of functions $\mathcal{F}$ that are "learnable" in a strong sense (to be made precise below) cannot be used for cryptographic sensing. The intuition being that the adversary Adv who observes the interaction transcript $\mathcal{I}$ can simply apply the learning algorithm to the examples it sees throughout the observed interaction and learn the concept by itself. To make this a bit more formal, we first recall the notion of *OCCAM learning*.

An OCCAM Learning algorithm for a class of functions $\mathcal{F}$, using a class of hypotheses $\mathcal{H}$ and constants $a \geq 0$ and $0 \leq b < 1$, is an algorithm $\mathcal{A}$ that, given any set (sample) S of m examples in $\{0,1\}^n$, labeled by any $f \in \mathcal{F}$, outputs an hypothesis $h \in \mathcal{H}$ such that: (1) h is consistent with S (i.e., it agrees with the hidden f on the labels of all examples); and (2) h is "succinct", i.e. size(h) is bounded by[3] $(n \cdot \text{size}(f))^a \cdot m^b$. Algorithm $\mathcal{A}$ is efficient if it runs in time polynomial in n, m and size(f).

Occam's Razor is a well-known philosophical principle. Its connection to machine learning was made by [17], who showed that it is essentially equivalent to Valiant's notion of PAC-learnability [48]. Concretely, they showed that an OCCAM learner $\mathcal{A}$ can be turned into a PAC learner $\mathcal{A}'$ (essentially showing that if $\mathcal{A}'$ feeds $\mathcal{A}$ with enough random examples, as a function of the parameters a, b of $\mathcal{A}$, the hypotheses h that $\mathcal{A}$ outputs is good enough) thus providing a natural approach for designing PAC-learning algorithms. The converse direction, namely that PAC learnability implies OCCAM learnability also holds [47].[4]

We now conclude that, for classes $\mathcal{F}$ that admit OCCAM learnability, the adversary Adv can apply the corresponding OCCAM algorithm $\mathcal{A}$ to get a hypothesis h that is consistent with f on all examples. If there is a sensing algorithm Sen that is able to identify the concept f based on these examples, then so can Adv. This rules out even our weakest notion of one-way security.

Such efficient OCCAM (alternatively PAC) learning algorithms are known for classes such as disjunctions, conjunctions and k-DNFs for constant k [48], decision lists [45], and more. Thus, all these classes are not candidates for cryptographic sensing. For a richer class such as (poly-size) DNFs, the question of its efficient learnability is wide open. Designing (even a one-way secure) cryptographic sensing algorithm for such a class would therefore imply that it cannot be PAC-learned efficiently without membership queries. While proving hardness under standard intractability assumptions may be a difficult challenge, coming up with explicit plausible candidates for hard distributions is a problem that apparently did not receive much attention.

On the optimistic side, PAC-learning algorithms are known only for limited classes of functions (hence, the above negative result is limited as well). For other classes, sensing may or may not be possible. Note that, intuitively, cryptographic sensing is closer in spirit to the stronger setting of PAC with membership queries (MQ). In such a model, one can learn more expressive classes such as Decision Trees [21] and DFAs [3].

[3] The requirement that $b < 1$ is what rules out the trivial solution where h is just the list of labels for the m points in S and forces actual "learning".

[4] In the case of *proper* PAC-learning (i.e., when $\mathcal{H} = \mathcal{F}$), [18] present a condition (called "closure under exception lists") on $\mathcal{F}$ under which PAC still implies OCCAM learning.

There are several non-trivial sub-exponential algorithms for DNF. The best such algorithm is by Klivans and Servedio [37] and has complexity of roughly $2^{O(n^{1/3}\log n \log s)}$ for learning s-term DNF with n variables. Transforming this algorithm to an OCCAM learning algorithm, as above, gives a limit on the security of cryptographic sensing algorithms for DNF that one may hope to achieve. We also remark that known results on PAC-learnability of DNF under uniform distribution (this is known to be possible in quasi-polynomial time [49]) do not imply a negative result for cryptographic sensing.

5.2 Local Measurements

We now go back temporarily to the sensing formulation, focusing on a simple class of measurements that corresponds to work on low-complexity cryptography. Consider d-local measurement functions, namely the class $\mathcal{F}$ of "finite" functions f that depend on at most d bits of x. We note that, despite the simplicity of such functions, we are not aware of any natural physical realization that does not involve additional leakage. Still, it is natural to study the power of this class.

Entropic security cannot be realized in NC^0, as it is easy to construct, for any d-local function f, a pair of high-entropy distributions X_0, X_1 where for every $x \in X_0$ we have $f(x) = 0$ and for every $x \in X_1$ we have $f(x) = 1$ (the entropy can be as large as, say, $n - d$). The same impossibility holds for security with independent background noise (as defined in Section 3). However, in the setting of *random* background noise, where noise is a uniformly random bit-string, we can get positive results for $d = 4$. Indeed, under standard cryptographic assumptions, there is 4-local PKE [9], which implies a cryptographic sensing algorithm with random background noise. The above is still not satisfactory because it does not respect physical locality. Under a less standard but still plausible security assumption, namely the security of a variant of the McEliece cryptosystem, it is possible to get an analogous result with constant physical locality [11].

The amount of background noise in the above solutions is very large, $|x| \cdot \mathrm{poly}(\lambda)$. If we do not insist on physical locality, we can trade background noise for locality by using polynomial-stretch local PRGs [26, 10, 31, 5]. This can reduce the amount of background noise to $|x|^\epsilon \cdot \mathrm{poly}(\lambda)$, for any constant ϵ, while still maintaining constant locality d.

5.3 Distributed Solution for Learning Juntas

Finally, we demonstrate the potential usefulness of the distributed variant of cryptographic sensing by showing a positive result for the class of juntas. Learning juntas on $k = O(\log n)$ inputs from random examples is conjectured to be a hard learning problem (this conjecture is attributed to Avrim Blum). However, we argue that such f can be learned in the distributed setting (see Section 3) via two sets of labeled examples: S_1 that contains $\mathrm{poly}(n)$ random examples (the exact polynomial depends on k), and S_2 that contains a random Hamming-neighbor for each example in S_1, namely each example in S_2 is obtained by flipping a random bit in the corresponding example in S_1. Note that each of the two interactions $\mathcal{I}_1, \mathcal{I}_2$ separately is a collection of labeled random examples from which learning f, as mentioned, is conjectured to be hard. On the other hand, putting together the two interactions allow Sen to identify each of the k sensitive variables x_i, with high probability (by selecting, with probability $\geq 2^{-k}$, an assignment to S_1 that is sensitive at x_i and selecting to S_2 its i-th neighbor). Then the function itself can recovered in polynomial time from the answers to questions from, say, S_1 that cover all 2^k assignments to the k sensitive variables. Note that here quasi-polynomial security is the best that one can hope for, since the original problem can be solved in, roughly, $n^{k+O(1)}$ time (via a naive algorithm that checks all subsets of k variables) or even slightly better via a sophisticated algorithm of [41] that runs in time $n^{ck+O(1)}$, for some $c < 1$.

Acknowledgements. We thank Brent Waters for helpful discussions.

Research supported by NSF-BSF grant 2015782. Y. Ishai and E. Kushilevitz were additionally supported by ISF grant 1709/14 and a grant from the Ministry of Science and Technology, Israel and Department of Science and Technology, Government of India. Y. Ishai was additionally supported by ERC Project NTSC (742754). R. Ostrovsky was additionally supported by NSF grant 1619348, DARPA SafeWare subcontract to Galois Inc., DARPA SPAWAR contract N66001-15-C-4065, JP Morgan Faculty Research Award, OKAWA Foundation Research Award, IBM Faculty Research Award, Xerox Faculty Research Award, B. John Garrick Foundation Award, Teradata Research Award, and Lockheed-Martin Corporation Research Award. A. Sahai was additionally supported by a DARPA/ARL SAFEWARE award, NSF Frontier Award 1413955, and NSF grant 1619348, a Xerox Faculty Research Award, a Google Faculty Research Award, an equipment grant from Intel, and an Okawa Foundation Research Grant. This material is based upon work supported by the Defense Advanced Research Projects Agency through the ARL under Contract W911NF-15-C-0205. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense, the National Science Foundation or the U.S. Government.

References

1. Ajtai, M., Dwork, C.: A public-key cryptosystem with worst-case/average-case equivalence. In: Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997 (1997)
2. Alekhnovich, M.: More on average case vs approximation complexity. In: 44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings (2003)
3. Angluin, D.: Learning regular sets from queries and counterexamples. Inf. Comput. 75(2), 87–106 (1987)
4. Angluin, D., Kharitonov, M.: When won't membership queries help? (extended abstract). In: STOC (1991)
5. Applebaum, B.: Exponentially-hard gap-csp and local PRG via local hardcore functions. In: FOCS (2017)
6. Applebaum, B., Avron, J., Brzuska, C.: Arithmetic cryptography. J. ACM 64(2), 10:1–10:74 (2017)

7. Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In: Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings (2009)
8. Applebaum, B., Haramaty, N., Ishai, Y., Kushilevitz, E., Vaikuntanathan, V.: Low-complexity cryptographic hash functions. In: ITCS (2017)
9. Applebaum, B., Ishai, Y., Kushilevitz, E.: Cryptography in NC^0. In: FOCS (2004)
10. Applebaum, B., Ishai, Y., Kushilevitz, E.: On pseudorandom generators with linear stretch in nc^0. In: APPROX-RANDOM (2006)
11. Applebaum, B., Ishai, Y., Kushilevitz, E.: Cryptography by cellular automata or how fast can complexity emerge in nature? In: ICS (2010)
12. Applebaum, B., Ishai, Y., Kushilevitz, E.: How to garble arithmetic circuits. In: IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011 (2011)
13. Bellare, M., Boldyreva, A., O'Neill, A.: Deterministic and efficiently searchable encryption. IACR Cryptology ePrint Archive 2006, 186 (2006)
14. Bellare, M., Brakerski, Z., Naor, M., Ristenpart, T., Segev, G., Shacham, H., Yilek, S.: Hedged public-key encryption: How to protect against bad randomness. In: Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings (2009)
15. Bellare, M., Kiltz, E., Peikert, C., Waters, B.: Identity-based (lossy) trapdoor functions and applications. In: Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings (2012)
16. Blum, A., Furst, M.L., Kearns, M.J., Lipton, R.J.: Cryptographic primitives based on hard learning problems. In: CRYPTO (1993)
17. Blumer, A., Ehrenfeucht, A., Haussler, D., Warmuth, M.K.: Occam's razor. Inf. Process. Lett. 24(6), 377–380 (1987)
18. Board, R.A., Pitt, L.: On the necessity of occam algorithms. In: STOC (1990)
19. Boldyreva, A., Fehr, S., O'Neill, A.: On notions of security for deterministic encryption, and efficient constructions without random oracles. In: Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings (2008)
20. Bootle, J., Delaplace, C., Espitau, T., Fouque, P., Tibouchi, M.: LWE without modular reduction and improved side-channel attacks against BLISS. IACR Cryptology ePrint Archive 2018, 822 (2018), to appear in Asiacrypt 2018
21. Bshouty, N.H.: Exact learning via the monotone theory (extended abstract). In: FOCS (1993)
22. Bshouty, N.H., Eiron, N., Kushilevitz, E.: PAC learning with nasty noise. In: ALT (1999)
23. Cohen, A., Goldwasser, S., Vaikuntanathan, V.: Aggregate pseudorandom functions and connections to learning. In: TCC (2015)
24. Dodis, Y., Smith, A.D.: Entropic security and the encryption of high entropy messages. In: Theory of Cryptography, Second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA, February 10-12, 2005, Proceedings (2005)
25. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008 (2008)
26. Goldreich, O.: Candidate one-way functions based on expander graphs. In: Studies in Complexity and Cryptography (2011)
27. Goldwasser, S., Micali, S.: Probabilistic encryption. J. Comput. Syst. Sci. 28(2), 270–299 (1984)
28. Impagliazzo, R., Levin, L.A., Luby, M.: Pseudo-random generation from one-way functions. In: Proceedings of the Twenty-first Annual ACM Symposium on Theory of Computing. STOC '89 (1989)
29. Indyk, P.: Sketching via hashing: from heavy hitters to compressed sensing to sparse fourier transform. In: Proceedings of the 32nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2013, New York, NY, USA - June 22 - 27, 2013 (2013)
30. Ishai, Y., Kushilevitz, E.: Randomizing polynomials: A new representation with applications to round-efficient secure computation. In: FOCS (2000)
31. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Cryptography with constant computational overhead. In: STOC (2008)
32. Ishai, Y., Prabhakaran, M., Sahai, A.: Secure arithmetic computation with no honest majority. In: TCC (2009)
33. Kannan, S., Mossel, E., Sanyal, S., Yaroslavtsev, G.: Linear sketching over f_2. In: 33rd Computational Complexity Conference, CCC 2018, June 22-24, 2018, San Diego, CA, USA (2018)
34. Kasiviswanathan, S.P., Lee, H.K., Nissim, K., Raskhodnikova, S., Smith, A.D.: What can we learn privately? In: FOCS (2008)
35. Kearns, M.J., Valiant, L.G.: Cryptographic limitations on learning boolean formulae and finite automata. In: STOC (1989)
36. Kharitonov, M.: Cryptographic hardness of distribution-specific learning. In: STOC (1993)

37. Klivans, A.R., Servedio, R.A.: Learning DNF in time $2^{\tilde{o}(n^{1/3})}$. In: STOC (2001)
38. Mahloujifar, S., Diochnos, D.I., Mahmoody, M.: Learning under p-tampering attacks. In: ALT (2018)
39. McEliece, R.J.: A Public-Key Cryptosystem Based On Algebraic Coding Theory. Deep Space Network Progress Report 44, 114–116 (jan 1978)
40. Micciancio, D., Peikert, C.: Hardness of SIS and LWE with small parameters. In: Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I (2013)
41. Mossel, E., O'Donnell, R., Servedio, R.A.: Learning juntas. In: STOC (2003)
42. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008 (2008)
43. Pitt, L., Valiant, L.G.: Computational limitations on learning from examples. J. ACM 35(4), 965–984 (1988)
44. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC (2005)
45. Rivest, R.L.: Learning decision lists. Machine Learning 2(3), 229–246 (1987)
46. Russell, A., Wang, H.: How to fool an unbounded adversary with a short key. In: Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings (2002)
47. Schapire, R.E.: The strength of weak learnability (extended abstract). In: FOCS (1989)
48. Valiant, L.G.: A theory of the learnable. In: STOC (1984)
49. Verbeurgt, K.A.: Learning DNF under the uniform distribution in quasi-polynomial time. In: COLT (1990)