Robustness of the Learning with Errors Assumption

Shafi Goldwasser[1] Yael Kalai[2] Chris Peikert[3] Vinod Vaikuntanathan[4]
[1]MIT and the Weizmann Institute
[2]Microsoft Research
[3]Georgia Institute of Technology
[4]IBM Research
shafi@csail.mit.edu yael@microsoft.com cpeikert@cc.gatech.edu vinodv@alum.mit.edu

Abstract: Starting with the work of Ishai-Sahai-Wagner and Micali-Reyzin, a new goal has been set within the theory of cryptography community, to design cryptographic primitives that are secure against large classes of side-channel attacks. Recently, many works have focused on designing various cryptographic primitives that are robust (retain security) even when the secret key is "leaky", under various intractability assumptions. In this work we propose to take a step back and ask a more basic question: which of our cryptographic *assumptions* (rather than cryptographic schemes) are robust in presence of leakage of their underlying secrets?

Our main result is that the hardness of the *learning with error* (LWE) problem implies its hardness with *leaky secrets*. More generally, we show that the standard LWE assumption implies that LWE is secure even if the secret is taken from an arbitrary distribution with sufficient entropy, and even in the presence of hard-to-invert auxiliary inputs. We exhibit various applications of this result.

1. Under the *standard LWE assumption*, we construct a symmetric-key encryption scheme that is robust to secret key leakage, and more generally maintains security even if the secret key is taken from an arbitrary distribution with sufficient entropy (and even in the presence of hard-to-invert auxiliary inputs).
2. Under the *standard LWE assumption*, we construct a (weak) obfuscator for the class of point functions with multi-bit output.

We note that in most schemes that are known to be robust to leakage, the parameters of the scheme depend on the *maximum* leakage the system can tolerate, and hence the efficiency degrades with the *maximum anticipated leakage*, even if no leakage occurs at all! In contrast, the fact that we rely on a robust assumption allows us to construct a *single* symmetric-key encryption scheme, with parameters that are *independent* of the anticipated leakage, that is robust to *any* leakage (as long as the secret key has sufficient entropy left over). Namely, for any $k < n$ (where n is the size of the secret key), if the secret key has only entropy k, then the security relies on the LWE assumption with secret size roughly k.

Keywords: Leakage-resilient Cryptography, Learning with Errors, Symmetric-key Encryption, Program Obfuscation.

1 Introduction

The development of the theory of cryptography, starting from the foundational work in the early 80's has led to rigorous definitions of security, mathematical modeling of cryptographic goals, and what it means for a cryptographic algorithm to achieve the stated security goals.

A typical security definition builds on the following framework: a cryptographic algorithm is modeled as a Turing machine (whose description is known to all) initialized with a secret key. Adversarial entities, modeled as arbitrary (probabilistic) polynomial-time machines have *input/output access* to the algorithm. The requirement is that it is infeasible for any such adversary to "break" the system at hand. The (often implicit) assumption in such a definition is that the secret keys used by the algorithm are *perfectly secret* and chosen afresh for

the algorithm. In practice, however, information about keys does get compromised for a variety of reasons, including: various side-channel attacks, the use of the same secret key across several applications, or the use of correlated and imperfect sources of randomness to generate the keys. In short, adversaries in the real world can typically obtain information about the secret keys other than through the prescribed input/output interface.

In recent years, starting with the work of Ishai, Sahai and Wagner [19] and Micali and Reyzin [21], a new goal has been set within the theory of cryptography community to build *general theories of security in the presence of information leakage*. A large body of work has accumulated by now (see [1, 3, 6, 8, 12–14, 14, 16, 19, 21, 22, 25, 26, 28] and the references therein) in which security against different classes of information leakage has been defined, and different cryptographic primitives have been designed to *provably withstand* these attacks. A set of works particularly relevant to our work are those that design cryptographic primitives which are "robust" in the following sense: they remain secure even in the presence of attacks which obtain *arbitrary polynomial-time computable* information about the secret keys, as long as "the secret key is not fully revealed" (either information theoretically or computationally) [1, 11, 12, 20, 22].

In this paper, we turn our attention to the notion of *robust cryptographic assumptions* rather than robust cryptographic constructions.

Robustness of Cryptographic Assumptions

Suppose one could make a non-standard assumption of the form "cryptographic assumption A holds even in the presence of arbitrary leakage of its secrets". An example would be that the factoring problem is hard even given partial information about the prime factors. Once we are at liberty to make such an assumption, the task of coming up with leakage-resilient cryptographic algorithms becomes substantially easier. However, such assumptions are rather unappealing, unsubstantiated and sometimes (as in the case of factoring) even false! (See below.)

In addition, it is often possible to show that a quantitatively stronger hardness assumption translates to some form of leakage-resilience. For example, the assumption that the discrete logarithm problem is 2^k-hard (for some $k > 0$) directly implies its security in the presence of roughly k bits of leakage.[1] However, in practice, what is interesting is a cryptographic assumption that is secure against leakage of a *constant fraction* of its secret. The problem with the above approach is that it fails this goal, since none of the cryptographic assumptions in common use are $2^{O(N)}$-hard to break (where N is the length of the secret key).

Thus, most of the recent work on leakage-resilient cryptography focuses on constructing cryptographic schemes whose leakage-resilience can be reduced to *standard* cryptographic assumptions such as the *polynomial hardness* of problems based on factoring, discrete-log or various lattice problems, or in some cases, general assumptions such as the existence of one-way functions.

However, a question that still remains is: Which of the (standard) cryptographic assumptions are themselves naturally "robust to leakage". Specifically:

- Is it hard to factor an RSA composite $N = pq$ (where p and q are random n-bit primes) given arbitrary, but bounded, information about p and q?
- Is it hard to compute x given $g^x \pmod p$, where p is a large prime and x is uniformly random given arbitrary, but bounded, information about x?

As for factoring with leakage, there are known negative results: a long line of results starting from the early work of Rivest and Shamir [29] show how to factor $N = pq$ given a small fraction of the bits of one of the factors [10, 17]. Similar negative results are known for the RSA problem, and the square-root extraction problem.

As for the discrete-logarithm problem with leakage, it could very well be hard. In fact, an assumption of this nature has been put forward by Canetti and used to construct an obfuscator for point functions [7], and later to construct a symmetric encryption scheme that is robust to leakage [9]. However, we do not have any evidence for the validity of such an assumption, and in particular, we are far from showing that a standard cryptographic assumption implies the discrete-log assumption with leakage.

[1] It is worth noting that such an implication is not entirely obvious for *decisional* assumptions.

Recently, Dodis, Kalai and Lovett [12] consider this question in the context of the learning parity with noise (LPN) problem. They showed that the LPN assumption with leakage follows from a related, but non-standard assumption they introduce, called the learning *subspaces* with noise (LSN) assumption.

In light of this situation, we ask:

Is there a standard *cryptographic assumption that is robust with leakage?*

Our first contribution is to show that the learning with errors (LWE) assumption [27] that has recently gained much popularity (and whose average-case complexity is related to the *worst-case* complexity of various lattice problems) is in fact robust in the presence of leakage (see Section 1.1 and Theorem 1 below for details).

Leakage versus Efficiency

Typically, the way leakage-resilient cryptographic primitives are designed is as follows: first, the designer determines the *maximum* amount of leakage he is willing to tolerate. Then, the scheme is designed, and the parameters of the scheme (and hence its efficiency) depend on the maximum amount of leakage the scheme can tolerate. The more this quantity is, the less efficient the scheme becomes. In other words, the efficiency of the scheme depends on the *maximum anticipated* amount of leakage. In cases where there is no leakage at all in a typical operation of the system, this is grossly inefficient. To our knowledge, all known leakage-resilient cryptographic schemes that rely on standard assumptions follow this design paradigm.

In contrast, what we would like is a scheme whose *security*, and not efficiency, degrades with the actual amount of leakage. In other words, we would like to design a single scheme whose security with different amounts of leakage can be proven under a (standard) cryptographic assumption, with degraded parameters. The larger the leakage, the stronger the security assumption. In other words, the amount of leakage never shows up in the design phase, and comes up only in the proof of security. We call this a "graceful degradation of security". This leads us to ask:

Are there cryptosystems that exhibit a graceful degradation of security?

Our second contribution is a symmetric-key encryption scheme with a graceful degradation of security. The construction is based on the LWE assumption, and uses the fact that the LWE assumption is robust to leakage. (See Section 1.1 and Theorem 2 below for details.)

Our final contribution is a (weak) obfuscator for the class of point functions with multi-bit output with a graceful degradation of security. (See Section 1.1 and Theorem 3 below for details.)

1.1 Our Results and Techniques

The learning with error (LWE) problem, introduced by Regev [27] is a generalization of the well-known "learning parity with noise" problem. For a security parameter n, and a prime number q, the LWE problem with secret $\mathbf{s} \in \mathbb{Z}_q^n$ is defined as follows: given polynomially many random, noisy linear equations in $\mathbf{s}$, find $\mathbf{s}$. More precisely, given $(\mathbf{a}_i, \langle \mathbf{a}_i, \mathbf{s} \rangle + x_i)$, where $\mathbf{a}_i \leftarrow \mathbb{Z}_q^n$ is uniformly random, and x_i is drawn from a "narrow error distribution", the problem is to find $\mathbf{s}$. The decisional version of LWE is to distinguish between polynomially many LWE samples $\{(\mathbf{a}_i, \langle \mathbf{a}_i, \mathbf{s} \rangle + x_i)\}$ and uniformly random samples. Typically, the LWE problem is considered with a Gaussian-like error distribution (see Section 2 for details). For notational convenience, we use a compact matrix notation for LWE: we will denote m samples from the LWE distribution compactly as $(\mathbf{A}, \mathbf{As} + \mathbf{x})$, where $\mathbf{A}$ is an m-by-n matrix over $\mathbb{Z}_q$.

Our confidence in the LWE assumption stems from a worst-case to average-case reduction of Regev [27], who showed that the (average-case, search) LWE problem is (quantumly) as hard as the approximation versions of various standard lattice problems in the worst-case. Furthermore, the decisional and search LWE problems are equivalent (up to polynomial in q factors).

1.1.1 LWE with Weak Secrets

We prove that the LWE assumption is robust to leakage. More generally, we prove that it is robust to weak keys and to auxiliary input functions that are (computationally) hard to invert. Namely, we show that (for some setting of parameters) the standard LWE assumption implies that LWE is hard even if the secret $\mathbf{s}$ is chosen from an arbitrary distribution with sufficient min-entropy, and even given arbitrary hard-to-invert auxiliary input $f(\mathbf{s})$.

For the sake of clarity, in the introduction we focus on the case that the secret $\mathbf{s}$ is distributed according to an arbitrary distribution with sufficient min-entropy, and defer the issue of auxiliary input to the body of the paper (though all our theorems hold also w.r.t. auxiliary inputs).

We show that if the modulus q is any super-polynomial function of n, then the LWE assumption with weak binary secrets (i.e., when the secret $\mathbf{s}$ is distributed according to an arbitrary distribution over $\{0,1\}^n$ with sufficient min-entropy) follows from the standard LWE assumption. More specifically, we show that the LWE assumption where the secret $\mathbf{s}$ is drawn from an *arbitrary weak source* with min-entropy k over $\{0,1\}^n$ is true, assuming the LWE assumption is true with *uniform* (but smaller) secrets from $\mathbb{Z}_q^\ell$, where $\ell = \frac{k-\omega(\log n)}{\log q}$. Namely, we reduce the LWE assumption with weak keys to the standard LWE assumption with "security parameter" ℓ. A caveat is that we need to assume that the (standard) LWE assumption is true with a super-polynomial modulus q, and a super-polynomially small "error-rate". Translated into the worst-case regime using Regev's worst-case to average-case reduction, the assumption is that standard lattice problems are (quantumly) hard to approximate with quasi-polynomial time and within quasi-polynomial factors.[2]

Theorem 1 (Informal). *For any super-polynomial modulus $q = q(n)$, any $k \geq \log q$, and any distribution $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ over $\{0,1\}^n$ with min-entropy k, the (non-standard) LWE assumption, where the secret is drawn from the distribution $\mathcal{D}$, follows from the (standard) LWE assumption with secret size $\ell \triangleq \frac{k-\omega(\log n)}{\log q}$ (where the "error-rate" is super-polynomially small and the adversaries run in time* $\mathsf{poly}(n)$*).*

We sketch the main ideas behind the proof of this theorem. Let us first perform a mental experiment where the matrix $\mathbf{A}$ in the definition of the LWE problem is a *non-full-rank* matrix. i.e, $\mathbf{A} = \mathbf{B}\cdot\mathbf{C}$ where $\mathbf{B} \leftarrow \mathbb{Z}_q^{m\times\ell}$ and $\mathbf{C} \leftarrow \mathbb{Z}_q^{\ell\times n}$ are uniformly random; this is a uniformly random matrix with rank at most ℓ (and exactly ℓ if both $\mathbf{B}$ and $\mathbf{C}$ are full rank). The LWE distribution then becomes

$$\mathbf{A}\mathbf{s} + \mathbf{x} = \mathbf{B}\cdot(\mathbf{C}\mathbf{s}) + \mathbf{x}$$

At this point, we use the leftover hash lemma,[3] which states that matrix multiplication over $\mathbb{Z}_q$ (and in fact, any universal hash function) acts as a (strong) randomness extractor. In other words, $\mathbf{C}\mathbf{s}$ is statistically close to a uniformly random vector $\mathbf{t}$ (even given $\mathbf{C}$). Now, $\mathbf{B}\mathbf{t} + \mathbf{x}$ is exactly the LWE distribution with a *uniformly random secret* $\mathbf{t}$ which, by the LWE assumption with security parameter ℓ, is pseudorandom. It seems that we are done, but that is not quite the case.

The problem is that $\mathbf{B}\cdot\mathbf{C}$ does not look anything like a uniformly random m-by-n matrix; it is in fact easily distinguishable from the uniform distribution. At first, one may be tempted to simply change the LWE assumption, by choosing $\mathbf{A} = \mathbf{B}\cdot\mathbf{C}$ as above, rather than a random m-by-n matrix. The problem with this approach is that if $\mathbf{C}$ is fixed then the min-entropy of the secret $\mathbf{s}$ may depend on $\mathbf{C}$, in which case we cannot use the leftover hash lemma and claim that $\mathbf{C}\cdot\mathbf{s}$ is uniformly distributed. On the other hand, if $\mathbf{C}$ is chosen at random each time, then we essentially reduced the problem of LWE with weak keys, to the problem to LWE with related secrets $\mathbf{C}_1\mathbf{s}, \ldots \mathbf{C}_t\mathbf{s}$, where $\mathbf{C}_1, \ldots, \mathbf{C}_t$ are known.

We take a different approach: we use the LWE assumption to "hide" the fact that $\mathbf{B}\cdot\mathbf{C}$ is not a full-rank matrix. More precisely, we claim that the matrix $\mathbf{B}\cdot\mathbf{C} + \mathbf{Z}$, where $\mathbf{Z}$ is chosen from the LWE error distribution, is computationally indistinguishable from a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$. This is simply because each column $\mathbf{B}\cdot\mathbf{c}_i + \mathbf{z}_i$ can be thought of as a bunch of LWE samples with "secret" $\mathbf{c}_i \in \mathbb{Z}_q^\ell$. By the LWE assumption with security parameter ℓ, this looks random to a polynomial-time observer. This technique was used in the work of Alekhnovitch [2] in the context of learning parity with noise, and was later used in a number of works in the context of the LWE assumption [24, 27].

An astute reader might have noticed that introducing this trick in fact undid the "extraction" ar-

[2] By taking q to be smooth, and following an argument from [23], our assumption translates to the assumption that standard lattice problems are (quantumly) hard to approximate in *polynomial* time and within super-polynomial factors.

[3] In the regime of auxiliary inputs, we use the Goldreich-Levin theorem over $\mathbb{Z}_q$ from the recent work of [11].

gument. In particular, now one has to consider

$$(\mathbf{B}\cdot\mathbf{C}+\mathbf{Z})\mathbf{s}+\mathbf{x}=(\mathbf{B}\cdot\mathbf{C}\mathbf{s})+\mathbf{Z}\mathbf{s}+\mathbf{x}$$

Indeed, $\mathbf{B}\cdot\mathbf{C}\mathbf{s}+\mathbf{x}=\mathbf{B}\mathbf{t}+\mathbf{x}$ by itself is pseudorandom (as before), but is not necessarily pseudorandom after the addition of the *correlated* $\mathbf{Z}\mathbf{s}$ component. We mitigate this problem by using the following elementary property of a Gaussian distribution: shifting a Gaussian distributed random variable by any number that is super-polynomially smaller than its standard deviation results in distribution that is statistically close to the original Gaussian distribution. Thus, if we set each entry of $\mathbf{Z}$ to be super-polynomially smaller than the standard deviation of $\mathbf{x}$ and choose $\mathbf{s}\in\{0,1\}^n$, we see that $\mathbf{Z}\mathbf{s}+\mathbf{x}$ is distributed statistically close to a Gaussian. Namely, the noise $\mathbf{x}$ "eats up" the correlated and troublesome term $\mathbf{Z}\mathbf{s}$. This is where the restrictions of the LWE secret $\mathbf{s}$ being binary, as well as q being super-polynomial and the noise-rate being negligibly smaller than q come into play.

1.1.2 Applications

Our main result has several applications. We construct an efficient symmetric encryption scheme secure against chosen plaintext attacks, even if the secret key is chosen from an arbitrary distribution with sufficient min-entropy (and even in the presence of arbitrary hard-to-invert auxiliary input). We also construct an obfuscator for the class of point functions with multi-bit output $\mathcal{I}=\{I_{(K,M)}\}$, that is secure w.r.t. any distribution with sufficient min-entropy. A point function with multi-bit output is a function $I_{(K,M)}$ that always outputs $\perp$, except on input K in which it outputs M. Both of these results rely on the standard LWE assumption.

Theorem 2 (Informal). *For any super-polynomial $q=q(n)$ there is a symmetric-key encryption scheme with the following property: For any $k\geq\log q$, the encryption scheme is CPA-secure when the secret-key is drawn from an arbitrary distribution with min-entropy k. The assumption we rely on is the (standard) LWE assumption with a secret of size $\ell\triangleq\frac{k-\omega(\log n)}{\log q}$ (where the "error-rate" is super-polynomially small and the adversaries run in time $\mathsf{poly}(n)$).*

The encryption scheme is simple, and similar versions of it were used in previous works [4, 12]. The secret-key is a uniformly random vector $\mathbf{s}\leftarrow\{0,1\}^n$. To encrypt a message $\mathbf{w}\in\{0,1\}^m$, the encryption algorithm computes

$$E_{\mathbf{s}}(\mathbf{w})=(\mathbf{A},\mathbf{A}\mathbf{s}+\mathbf{x}+q/2\cdot\mathbf{w})$$

where $\mathbf{A}\in_R\mathbb{Z}_q^{m\times n}$ and each component of $\mathbf{x}$ is drawn from the error distribution. To decrypt a ciphertext $(\mathbf{A},\mathbf{y})$ using the secret-key $\mathbf{s}$, the decryption algorithm computes $\mathbf{y}-\mathbf{A}\mathbf{s}$ and it deciphers each coordinate $i\in[m]$ separately, as follows: If the i'th coordinate of $\mathbf{y}-\mathbf{A}\mathbf{s}$ is close to $q/2$ (i.e., between $\frac{3q}{8}$ and $\frac{5q}{8}$) then it deciphers the i'th coordinate of the message to be 1. If the i'th coordinate of $\mathbf{y}-\mathbf{A}\mathbf{s}$ is close to 0 or q (i.e., is smaller than $\frac{q}{8}$ or larger than $\frac{7q}{8}$) then it deciphers the i'th coordinate of the message to be 0. Otherwise, it outputs $\perp$.

The ciphertext corresponding to a message $\mathbf{w}$ consists of polynomially many samples from the LWE distribution with secret $\mathbf{s}$, added to the message vector. Robustness of LWE means that the ciphertext is pseudorandom even if the secret key is chosen from an arbitrary distribution with min-entropy k (assuming that the standard LWE assumption holds with secrets drawn from $\mathbb{Z}_q^\ell$ where $\ell=(k-\omega(\log n))/\log q$). The same argument also shows that the scheme is secure in the presence of computationally uninvertible auxiliary input functions of $\mathbf{s}$.

Finally, we use a very recent work Canetti *et. al.* [9], that shows a tight connection between secure encryption w.r.t. weak keys and obfuscation of point functions with multi-bit output. The security definition that they (and we) consider is a distributional one: Rather than requiring the obfuscation to be secure w.r.t. *every* function in the class, as defined in the seminal work of [5], security is required only when the functions are distributed according to a distribution with sufficient min-entropy. Using this connection, together with our encryption scheme described above, we get the following theorem.

Theorem 3 (Informal). *There exists an obfuscator for the class of point functions with multi-bit output under the (standard) LWE assumption.*

1.2 Related Work

There is a long line of work that considers various models of leakage [1, 3, 8, 9, 11, 11, 12, 14, 14,

16, 19–22, 25, 26]. Still, most of the schemes that are known to be resilient to leakage suffer from the fact that the parameters of the scheme depend on the *maximum anticipated leakage*, and thus they are inefficient even in the absence of leakage! There are a few exceptions. For example, the work of [7, 9] exhibit a symmetric encryption scheme that is based on a *leakage-resilient DDH assumption*, which says that DDH holds even if one of its secrets is taken from an arbitrary distribution with sufficient min-entropy. We also mention several results in the continual-leakage model which rely on exponential hardness assumptions [14, 26]. We emphasize that all these examples rely on *non-standard* assumptions.

The work of Katz and Vaikuntanathan [20] constructs signature schemes that are resilient to leakage. They, similarly to us, "push" the leakage to the assumption level. To construct their signature schemes, they use the observation that any collision-resistant hash function (and even a universally one-way hash function) is a leakage-resilient one-way function. In other words, they use the fact that if h is collision-resistant, then $h(x)$ is hard to invert even given some partial information about the pre-image x. This can be interpreted as saying that the collision-resistant hash function assumption is a robust assumption.

This differs from our work in several ways. The most significant difference is that we show that pseudorandomness (and not only one-wayness) of the LWE distribution is preserved in the presence of leakage. This enables us to construct various cryptographic objects which we do not know how to construct with the [20] assumption. We mention that, whereas the [20] observation follows by a straightforward counting argument, our proof requires non-trivial computational arguments: roughly speaking, this is because the secret $\mathbf{s}$ is uniquely determined (and thus has no information-theoretic entropy) given the LWE samples $(\mathbf{A}, \mathbf{As} + \mathbf{x})$. Thus, we have to employ non-trivial computational claims to prove our statements.

Finally, we would like to contrast our results with those of Akavia et al. [1], who show that the *public-key* encryption scheme of Regev [27] based on LWE is in fact leakage-resilient. Unfortunately, achieving larger and larger leakage in their scheme entails increasing the modulus q correspondingly. Roughly speaking, to obtain robustness against a leakage of $(1-\epsilon)$ fraction of the secret key, the modulus has to be at least $n^{1/\epsilon}$ (where n is the security parameter). In short, the scheme does not degrade gracefully with leakage (much like the later works of [11, 22]). In fact, constructing a public-key encryption scheme with a graceful degradation of security remains a very interesting open question.

2 Preliminaries

We will let n denote the main security parameter throughout the paper. The notation $X \approx_c Y$ (resp. $X \approx_s Y$) means that the random variables X and Y are computationally indistinguishable (resp. statistically indistinguishable).

2.1 Learning with Errors (LWE)

The LWE problem was introduced by Regev [27] as a generalization of the "learning noisy parities" problem. Let $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ be the reals modulo 1, represented by the interval $[0, 1)$. For positive integers n and $q \geq 2$, a vector $\mathbf{s} \in \mathbb{Z}_q^n$, and a probability distribution ϕ on $\mathbb{R}$, let $A_{\mathbf{s},\phi}$ be the distribution over $\mathbb{Z}_q^n \times \mathbb{T}$ obtained by choosing $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random and an error term $x \leftarrow \phi$, and outputting $(\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s} \rangle / q + x) \in \mathbb{Z}_q^n \times \mathbb{T}$.

We also consider a *discretized* version of the LWE distribution. For a distribution ϕ over $\mathbb{R}$ and an (implicit) modulus q, let $\chi = \bar{\phi}$ denote the distribution over $\mathbb{Z}$ obtained by drawing $x \leftarrow \phi$ and outputting $\lfloor q \cdot x \rceil$. The distribution $A_{\mathbf{s},\chi}$ over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ is obtained by choosing $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random, an error term $x \leftarrow \chi$, and outputting $(\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s} \rangle + x)$, where all operations are performed over $\mathbb{Z}_q$. (Equivalently, draw a sample $(\mathbf{a}, b) \leftarrow A_{\mathbf{s},\phi}$ and output $(\mathbf{a}, \lfloor b \cdot q \rceil) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.)

Definition 1. *For an integer $q = q(n)$ and an error distribution $\phi = \phi(n)$ over $\mathbb{T}$, the (worst-case, search) learning with errors problem $\mathsf{LWE}_{n,q,\phi}$ in n dimensions is: given access to arbitrarily many independent samples from $A_{\mathbf{s},\phi}$, output $\mathbf{s}$ with non-negligible probability. The problem for discretized $\chi = \bar{\phi}$ is defined similarly.*

The (average-case) decision variant of the LWE problem, denoted $\mathsf{DLWE}_{n,q,\phi}$, is to distinguish,

with non-negligible advantage given arbitrarily many independent samples, the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{T}$ from $A_{\mathbf{s},\phi}$ for a uniformly random (and secret) $\mathbf{s} \in \mathbb{Z}_q^n$. The problem for discretized $\chi = \bar{\phi}$ is defined similarly.

In this work, we are also concerned with the average-case decision LWE problem where the secret $\mathbf{s}$ is drawn from a distribution $\mathcal{D}$ over $\mathbb{Z}_q^n$ (which may not necessarily be the uniform distribution). We denote this problem by $\mathsf{DLWE}_{n,q,\phi}(\mathcal{D})$.

Observe that simply by rounding, the $\mathsf{DLWE}_{n,q,\chi}(\mathcal{D})$ problem is no easier than the $\mathsf{DLWE}_{n,q,\phi}(\mathcal{D})$ problem, where $\chi = \bar{\phi}$.

We are primarily interested in the LWE problems where the error distribution ϕ is a Gaussian. For any $\alpha > 0$, the density function of a one-dimensional Gaussian probability distribution over $\mathbb{R}$ is given by $D_\alpha(x) = \exp(-\pi(x/\alpha)^2)/\alpha$. We write $\mathsf{LWE}_{n,q,\alpha}$ as an abbreviation for $\mathsf{LWE}_{n,q,D_\alpha}$.

It is known [4, 23, 27] that for moduli q of a certain form, the (average-case decision) $\mathsf{DLWE}_{n,q,\phi}$ problem is equivalent to the (worst-case search) $\mathsf{LWE}_{n,q,\phi}$ problem, up to a $\mathsf{poly}(n)$ factor in the number of samples used. In particular, this equivalence holds for any q that is a product of sufficiently large $\mathsf{poly}(n)$-bounded primes.

Evidence for the hardness of $\mathsf{LWE}_{n,q,\alpha}$ follows from results of Regev [27] and Peikert [23], who (informally speaking) showed that solving $\mathsf{LWE}_{n,q,\alpha}$ (for appropriate parameters) is no easier than solving approximation problems on n-dimensional lattices in the worst case. More precisely, under the hypothesis that $q \geq \omega(\sqrt{n})/\alpha$, these reductions obtain $\tilde{O}(n/\alpha)$-factor approximations for various worst-case lattice problems. (We refer the reader to [23, 27] for the details.) Note that if $q \ll 1/\alpha$, then the $\mathsf{LWE}_{n,q,\alpha}$ problem is trivially solvable because the exact value of each $\langle \mathbf{a}, \mathbf{s} \rangle$ can be recovered (with high probability) from its noisy version simply by rounding.

2.2 Leftover Hash Lemma

Informally, the leftover hash lemma [18] states that any universal hash function acts as a randomness extractor. We will use a variant of this statement applied to a particular universal hash function, i.e, matrix multiplication over $\mathbb{Z}_q$. In particular:

Lemma 1. *Let $\mathcal{D}$ be a distribution over $\mathbb{Z}_q^n$ with min-entropy k. For any $\epsilon > 0$ and $\ell \leq (k - 2\log(1/\epsilon) - O(1))/\log q$, the joint distribution of $(\mathbf{C}, \mathbf{C} \cdot \mathbf{s})$ where $\mathbf{C} \leftarrow \mathbb{Z}_q^{\ell \times n}$ is uniformly random and $\mathbf{s} \in \mathbb{Z}_q^n$ is drawn from the distribution $\mathcal{D}$ is ϵ-close to the uniform distribution over $\mathbb{Z}_q^{\ell \times n} \times \mathbb{Z}_q^\ell$.*

2.3 Goldreich-Levin Theorem over $\mathbb{Z}_q$

We also need a "computational version" of the leftover hash lemma, which is essentially a Goldreich-Levin type theorem over $\mathbb{Z}_q$. The original Goldreich-Levin theorem proves that for every uninvertible function h, $\langle \mathbf{c}, \mathbf{s} \rangle \pmod 2$ is pseudorandom, given $h(\mathbf{s})$ and $\mathbf{c}$ for a uniformly random $\mathbf{c} \in \mathbb{Z}_2^n$. Dodis et al. [11] (following the work of Goldreich, Rubinfeld and Sudan [15]) show the following variant of the Goldreich-Levin theorem over a large field $\mathbb{Z}_q$.

Lemma 2. *Let q be prime, and let H be any polynomial size subset of $\mathbb{Z}_q$. Let $f : H^n \to \{0,1\}^*$ be any (possibly randomized) function. Let $\mathbf{C} \leftarrow \mathbb{Z}_q^{\ell \times n}$ be uniformly random and $\mathbf{s}$ be drawn from the distribution $\mathcal{D}$ over H^n.*

If there is a PPT algorithm A that distinguishes between $(\mathbf{C}, \mathbf{C}\mathbf{s})$ and the uniform distribution over the range given $h(\mathbf{s})$, then there is a PPT algorithm B that inverts $h(\mathbf{s})$ with probability roughly $1/(q^\ell \cdot \mathsf{poly}(n, 1/\epsilon))$.

In other words, if h is (roughly) $1/q^{2\ell}$-hard to invert (by polynomial-time algorithms), then $(\mathbf{C}, \mathbf{C} \cdot \mathbf{s})$ is pseudorandom given $h(\mathbf{s})$.

3 LWE with Weak Secrets

In this section, we show that if the modulus q is some super-polynomial function of n, then the LWE assumption with weak binary secrets (i.e., when the secret $\mathbf{s}$ is distributed according to an arbitrary distribution over $\{0,1\}^n$ with sufficient min-entropy) follows from the standard LWE assumption. More specifically, let $\mathbf{s} \in \{0,1\}^n$ be any random variable with min-entropy k, and let $q = q(n) \in 2^{\omega(\log n)}$ be any super-polynomial function in n. Then, we show that for every $m = \mathsf{poly}(n)$,

$$(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{x}) \approx_c (\mathbf{A}, \mathbf{u}),$$

where $\mathbf{A} \in_R \mathbb{Z}_q^{m \times n}$, $\mathbf{x} \leftarrow \overline{\Psi}_\beta^m$ and $\mathbf{u} \leftarrow \mathbb{Z}_q^m$. We show this statement under the (standard) LWE assumption with the following parameters: The secret size is $\ell \triangleq \frac{k-\omega(\log n)}{\log q}$, the number of examples remains m, and the standard deviation of the noise is any γ such that $\gamma/\beta = \mathsf{negl}(n)$. An example of a parameter setting that achieves the latter is $\beta = q/\mathsf{poly}(n)$ and $\gamma = \mathsf{poly}(n)$.

Theorem 4. *Let $n, q \geq 1$ be integers, let $\mathcal{D}$ be any distribution over $\{0,1\}^n$ having min-entropy at least k, and let $\alpha, \beta > 0$ be such that $\alpha/\beta = \mathsf{negl}(n)$. Then for any $\ell \leq \frac{k-\omega(\log n)}{\log q}$, there is a PPT reduction from $\mathsf{DLWE}_{\ell,q,\alpha}$ to $\mathsf{DLWE}_{n,q,\beta}(\mathcal{D})$.*

Remark 1**.** When converting the non-standard version of the LWE assumption to the standard LWE assumption we lose in two dimensions. The first is that the secret size becomes $\ell \triangleq \frac{k-\omega(\log n)}{\log q}$ rather than n. This loss seems to be inevitable since in the non-standard assumption, although the secret $\mathbf{s}$ was in $\{0,1\}^n$, it had only k bits of entropy, so when converting this assumption to the standard LWE assumption is seems like we cannot hope to get a secret key with more than k bits of entropy, which results with the key size being at most $k/\log q$. We remark that the dimension ℓ can be increased to $O(k - \omega(\log n))$ (thus making the underlying LWE assumption weaker) by consider secrets $\mathbf{s}$ over $\mathbb{Z}_q^n$ (rather than binary secrets). This modification makes the proof of the theorem a bit more cumbersome, and hence we choose not to present it.

Another dimension in which we lose is in the error distribution. The standard deviation of the error distribution becomes γ which is negligibly small compared to the original standard deviation β. This loss which does not seem to be inherent, and seems to be a byproduct of our proof, induces the restriction on q to be super-polynomial in n.

Remark 2**.** The analogous statement for the search version of LWE follows immediately from Theorem 4 and the search to decision reduction for LWE. However, doing so naively involves relying on the assumption that $\mathsf{LWE}_{\ell,m,q,\gamma}$ is hard (for the parameters ℓ, m, q and γ as above) for algorithms that run in *superpolynomial time*. The superpolynomial factor in the running time can be removed by using the more involved search to decision reduction of Peikert [23] that uses a modulus q of a special form.

3.1 Proof of Theorem 4

In the proof of Theorem 4 we rely on the following lemma, which was proven in [11].

Lemma 3. *Let $\beta > 0$ and $q \in \mathbb{Z}$.*

1. *Let $y \leftarrow \overline{\Psi}_\beta$. Then with overwhelming probability, $|y| \leq \beta q \cdot \sqrt{n}$.*
2. *Let $y \in \mathbb{Z}$ be arbitrary. The statistical distance between the distributions $\overline{\Psi}_\beta$ and $\overline{\Psi}_\beta + y$ is at most $|y|/(\beta q)$.*

Proof. Fix any $q, m, k, \ell, \beta, \gamma$ as in the statement theorem. We define the distribution D_γ, as follows: Let $\mathbf{B} \leftarrow \mathbb{Z}_q^{m \times \ell}$ and $\mathbf{C} \leftarrow \mathbb{Z}_q^{\ell \times n}$ be uniformly random and $\mathbf{Z} \leftarrow \overline{\Psi}_\gamma^{m \times n}$. Output the matrix $\mathbf{A}' = \mathbf{BC} + \mathbf{Z}$. The following claim about the distribution D_γ is then immediate:

Claim 1. *Under the $\mathsf{LWE}_{\ell,m,q,\gamma}$ Assumption, $\mathbf{A}'$ is computationally indistinguishable from uniform in $\mathbb{Z}_q^{m \times n}$.*

Claim 1 implies that it suffices to prove that

$$(\mathbf{A}', \mathbf{A}'\mathbf{s} + \mathbf{x}) \approx_c (\mathbf{A}', \mathbf{u}).$$

Note that

$$\mathbf{A}'\mathbf{s} + \mathbf{x} = (\mathbf{BC} + \mathbf{Z})\mathbf{s} + \mathbf{x} = \mathbf{BCs} + \mathbf{Zs} + \mathbf{x}.$$

Thus, it suffices to prove that

$$(\mathbf{BC} + \mathbf{Z}, \mathbf{BCs} + \mathbf{Zs} + \mathbf{x}) \approx_c (\mathbf{BC} + \mathbf{Z}, \mathbf{u}).$$

We prove the stronger statement that

$$(\mathbf{B}, \mathbf{C}, \mathbf{Z}, \mathbf{BCs} + \mathbf{Zs} + \mathbf{x}) \approx_c (\mathbf{B}, \mathbf{C}, \mathbf{Z}, \mathbf{u}). \quad (1)$$

The first item of Lemma 3 implies that with overwhelming probability each entry of $\mathbf{Z}$ is of size at most $\gamma q \cdot \sqrt{n}$. This implies that, with overwhelming probability (over $\mathbf{Z}$), for every $\mathbf{s} \in \{0,1\}^n$ each coordinate of $\mathbf{Zs}$ is of size at most $\gamma q \cdot n$. Let $\mathbf{x}'$ be a random variable distributed according to $(\overline{\Psi}_\beta)^m$. Then, the second item of Lemma 3 implies that for every $i \in [m]$ the statistical distance between $(\mathbf{Z}, \mathbf{s}, \mathbf{x}'_i)$ and $(\mathbf{Z}, \mathbf{s}, (\mathbf{Zs})_i + \mathbf{x}_i)$ is at most $\frac{\gamma q \cdot n}{\beta q}$. Using the fact that $\gamma/\beta = \mathsf{negl}(n)$, we conclude that $(\mathbf{Z}, \mathbf{s}, \mathbf{x}'_i)$ and $(\mathbf{Z}, \mathbf{s}, (\mathbf{Zs})_i + \mathbf{x}_i)$ are statistically close, and thus that $(\mathbf{Z}, \mathbf{s}, \mathbf{x}')$ and

$(\mathbf{Z}, \mathbf{s}, \mathbf{Zs} + \mathbf{x})$ are statistically close. Therefore, it suffices to prove that

$$(\mathbf{B}, \mathbf{C}, \mathbf{Z}, \mathbf{BCs} + \mathbf{x}') \approx_c (\mathbf{B}, \mathbf{C}, \mathbf{Z}, \mathbf{u}).$$

where $\mathbf{x}'$ is drawn from $\overline{\Psi}_\beta^m$.

The fact that $\mathbf{Z}$ is efficiently sampleable implies that it suffices to prove that

$$(\mathbf{B}, \mathbf{C}, \mathbf{BCs} + \mathbf{x}') \approx_c (\mathbf{B}, \mathbf{C}, \mathbf{u}).$$

A standard application of leftover hash lemma [18] using the fact that the min-entropy of $\mathbf{s}$ is at least $\ell \log q + \omega(\log n)$ implies that

$$(\mathbf{C}, \mathbf{Cs}) \approx_s (\mathbf{C}, \mathbf{u})$$

Thus, the $\mathsf{LWE}_{\ell,m,q,\beta}$ Assumption (which follows from the $\mathsf{LWE}_{\ell,m,q,\gamma}$ Assumption), immediately implies that

$$(\mathbf{B}, \mathbf{C}, \mathbf{BCs} + \mathbf{x}) \approx (\mathbf{B}, \mathbf{C}, \mathbf{u}),$$

as desired. □

The same technique used to prove Theorem 4, with the exception of using the Goldreich Levin theorem over $\mathbb{Z}_q$ (i.e, Lemma 2) instead of the leftover hash lemma shows that the LWE assumption holds even given auxiliary input $h(\mathbf{s})$, where h is any *uninvertible function* (See below). In essence, the proof of the auxiliary input theorem below proceeds by using Lemma 2 to extract from the "computational entropy" in the secret (as opposed to using the leftover hash lemma to extract from the "information-theoretic entropy").

Theorem 5. *Let $k \geq \log q$, and let $\mathcal{H}$ be the class of all functions $h : \{0,1\}^n \to \{0,1\}^*$ that are 2^{-k} hard to invert, i.e, given $h(\mathbf{s})$, no PPT algorithm can find $\mathbf{s}$ with probability better than 2^{-k}.*

For any super-polynomial $q = q(n)$, any $m = \mathsf{poly}(n)$, any $\beta, \gamma \in (0, q)$ such that $\gamma/\beta = \mathsf{negl}(n)$

$$(\mathbf{A}, \mathbf{As} + \mathbf{x}, h(\mathbf{s})) \approx_c (\mathbf{A}, \mathbf{u}, h(\mathbf{s}))$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{m \times n}$, $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ and $\mathbf{u} \leftarrow \mathbb{Z}_q^m$ are uniformly random and $\mathbf{x} \leftarrow \overline{\Psi}_\beta^m$. assuming the (standard) $\mathsf{DLWE}_{\ell,m,q,\gamma}$ assumption, where $\ell \triangleq \frac{k - \omega(\log n)}{\log q}$.

4 Symmetric-Key Encryption Scheme

In this section we present a symmetric-key encryption scheme that is secure even if the secret key is distributed according to an arbitrary distribution with sufficient min-entropy, assuming the standard LWE assumption holds (for some setting of parameters). More precisely, if the secret key $\mathbf{s} \in \{0,1\}^n$ is distributed according to some distribution $\mathcal{D}$ min-entropy k then the assumption we rely on is that there exists a super-polynomial function $q = q(n)$ and a parameter $\beta = q/\mathsf{poly}(n)$ for which $\mathsf{DLWE}_{k,q,\beta}(\mathcal{D})$ holds. According to Theorem 4, this assumption follows from the (standard) assumption $\mathsf{LWE}_{\ell,q,\gamma}$, where $\ell \triangleq \frac{k - \omega(\log n)}{\log q}$ and γ is any function that satisfies $\gamma/q = \mathsf{negl}(n)$.

4.1 The Scheme

We next describe our encryption scheme $\mathbb{E} = (G, E, D)$, which is very similar to schemes that appeared in previous work [4, 12].

- **Parameters.** Let $q = q(n)$ be any super-polynomial function, let $\beta = q/\mathsf{poly}(n)$, and let $m = \mathsf{poly}(n)$.
- **Key generation algorithm G.** On input 1^n, $G(1^n)$ outputs a uniformly random secret key $\mathbf{s} \leftarrow \{0,1\}^n$.
- **Encryption algorithm E.** On input a secret key $\mathbf{s}$ and a message $\mathbf{w} \in \{0,1\}^m$,

  $$E_{\mathbf{s}}(\mathbf{w}) = (\mathbf{A}, \mathbf{As} + \mathbf{x} + \frac{q}{2}\mathbf{w})$$

  where $\mathbf{A} \in_R \mathbb{Z}_q^{m \times n}$ and $\mathbf{x} \leftarrow (\overline{\Psi}_\beta)^m$.
- **Decryption algorithm D.** on input a secret key $\mathbf{s}$ and a ciphertext $(\mathbf{A}, \mathbf{y})$, the decryption algorithm computes $\mathbf{y} - \mathbf{As}$ and it deciphers each coordinate $i \in [m]$ separately, as follows: If the i'th coordinate of $\mathbf{y} - \mathbf{As}$ is close to $q/2$, i.e., is between $\frac{3q}{8}$ and $\frac{5q}{8}$, then it deciphers the i'th coordinate of the message to 1. If the i'th coordinate of $\mathbf{y} - \mathbf{As}$ is far from $q/2$, i.e., is smaller than $\frac{q}{8}$ or larger than $\frac{7q}{8}$, then it deciphers the i'th coordinate of the message to 0. Otherwise, it outputs $\perp$.

Remark. We note that, as opposed to most known leakage resilient schemes, the parameters of this scheme do not depend on any amount of leakage (or min-entropy) that the scheme can tolerate. Nevertheless, using Theorem 4, we are able to prove that this scheme is secure w.r.t. weak keys,

and in particular is leakage resilient. We emphasize the change in the order of quantifiers: Rather than proving that for every leakage parameter there exists a scheme that is secure w.r.t. such leakage, we show that there exists a scheme that is secure w.r.t. any leakage parameter, under a standard assumption, whose parameters depend on the leakage parameter. We note that the only other encryption scheme that is known to have this property is based on a *non-standard* version of the DDH assumption, which says that the DDH assumption holds even if one of its secrets is chosen from an arbitrary distribution with sufficient min-entropy [7, 9].

We first prove the correctness of our scheme.

Claim 2. *There exists a negligible function μ such that for every n and every $\mathbf{w} \in \{0,1\}^m$,*

$$\Pr[D_{\mathbf{s}}(E_{\mathbf{s}}(\mathbf{w})) = \mathbf{w}] = 1 - \mu(n)$$

Proof. Fix any n and any message $\mathbf{w} \in \{0,1\}^m$.

$$\Pr[(D_{\mathbf{s}}(E_{\mathbf{s}}(\mathbf{w}))) = \mathbf{w}] =$$
$$\Pr[\forall i \in [m],\ (D_{\mathbf{s}}(E_{\mathbf{s}}(\mathbf{w})))_i = \mathbf{w}_i] =$$
$$\Pr[\forall i \in [m],\ (D_{\mathbf{s}}(\mathbf{A}, \mathbf{As} + \mathbf{x} + \frac{q}{2}\mathbf{w}))_i = \mathbf{w}_i] =$$
$$\Pr\left[\forall i \in [m],\ \mathbf{x}_i \in \left(-\frac{q}{8}, \frac{q}{8}\right)\right] \geq$$
$$1 - m \Pr\left[\mathbf{x}_i \in \left(\frac{q}{8}, \frac{7q}{8}\right)\right] =$$
$$1 - \mathsf{negl}(n).$$

□

We next argue the security of this scheme.

Definition 2. *We say that a symmetric encryption scheme is CPA* **secure w.r.t. $k(n)$-weak keys***, if for any distribution $\{\mathcal{D}_n\}_{n\in\mathbb{N}}$ with min-entropy $k(n)$, the scheme is CPA secure even if the secret key is chosen according to the distribution $\mathcal{D}_n$.*

Theorem 6. *For any $k = k(n)$, the encryption scheme $\mathbb{E} = (G, E, D)$ is CPA secure with $k(n)$-weak keys, under the (standard) $\mathsf{DLWE}_{\ell,q,\gamma}$ assumption, where $\ell = \frac{k - \omega(\log n)}{\log q}$ and where γ satisfies $\gamma/q = \mathsf{negl}(n)$.*

In order to prove Theorem 6, we use the following notation: For any distribution $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ we denote by $G_{\mathcal{D}}$ the key generation algorithm that samples the secret key $\mathbf{s}$ according to the distribution $\mathcal{D}$. Namely, $G_{\mathcal{D}}(1^n)$ outputs $\mathbf{s} \leftarrow \mathcal{D}_n$.

Theorem 4 implies that in order to prove Theorem 6 it suffices to prove the following lemma, which states that if the secret key is samples according to some distribution $\mathcal{D}$, rather than sampled uniformly, then the scheme is secure under the (non-standard) $\mathsf{DLWE}_{n,q,\beta}(\mathcal{D})$ assumption.

Lemma 4. *Let $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ be an arbitrary distribution with min-entropy $k = k(n)$. Then, the encryption scheme $\mathbb{E}_{\mathcal{D}} = (G_{\mathcal{D}}, E, D)$ is CPA secure under the $\mathsf{DLWE}_{n,q,\beta}(\mathcal{D})$ assumption.*

4.2 Proof of Lemma 4

Let $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ be an arbitrary distribution with min-entropy $k = k(n)$. We will prove that no PPT adversary can distinguish between the case that it is given access to a valid encryption oracle and the case that it is given access to an oracle that simply outputs random strings. Suppose for the sake of contradiction that there exists a PPT adversary $\mathcal{A}$ that succeeds in distinguishing between the two oracles with probability ϵ, where ϵ is not a negligible function. We will show that this implies that there exists a polynomial $t = \mathsf{poly}(n)$ and a PPT algorithm $\mathcal{B}$ such that

$$|\Pr[\mathcal{B}(\mathbf{A}, \mathbf{As} + \mathbf{x}) = 1] - \Pr[\mathcal{B}(\mathbf{A}, \mathbf{u}) = 1]| \geq \epsilon(n) \tag{2}$$

where $\mathbf{A} \in_R \mathbb{Z}_q^{t\times n}$, $\mathbf{s} \leftarrow \mathcal{D}_n$ and $\mathbf{x} \leftarrow (\overline{\Psi}_\beta)^t$. This will contradict the $\mathsf{DLWE}_{n,q,\beta}(\mathcal{D})$ assumption. Algorithm $\mathcal{B}$ simply emulates $\mathcal{A}$'s oracle. Every time that $\mathcal{A}$ asks for an encryption of some message $\mathbf{w}$, the algorithm $\mathcal{B}$ takes m fresh rows of his input, denoted by $(\mathbf{A}, \mathbf{y})$ and feeds $\mathcal{A}$ the ciphertext $(\mathbf{A}, \mathbf{y} + \frac{q}{2}\mathbf{w})$. We choose t so that t/m is larger than the number of oracle queries that $\mathcal{A}$ makes. Note that if the input of $\mathcal{B}$ is an LWE instance then $\mathcal{B}$ perfectly simulates the encryption oracle. On the other hand, if the input of $\mathcal{B}$ is random, then $\mathcal{B}$ perfectly simulates the random oracle. Thus, Equation (2) indeed holds. □

Using Theorem 5 and a proof along similar lines as above, this scheme can also be shown to be secure against uninvertible auxiliary inputs.

5 Point Function Obfuscation with Multibit Output

In this section we consider the task of obfuscating the class of *point functions with multibit output* (or *multi-bit point functions (MBPF)* for short),

$$\mathcal{I} = \{I_{(k,m)} \mid k, m \in \{0,1\}^*\},$$

where each $I_{(k,m)} : \{0,1\}^* \cup \{\bot\} \to \{0,1\}^* \cup \bot$ is defined by

$$I_{(k,m)}(x) = \begin{cases} m & \text{if } x = k \\ \bot & \text{otherwise} \end{cases}$$

Namely, $I_{(k,m)}$ outputs the *message* m given a correct *key* k, and $\bot$ otherwise.

We show that our encryption scheme implies a (weak) obfuscator for the class of multi-bit point functions. To this end, we use a recent result of Canetti *et. al.* [9] that shows a tight connection between encryption schemes with weak keys and a weak form of obfuscation of multi-bit point functions. The obfuscation definition they consider is a distributional one: Rather than requiring the obfuscation to be secure w.r.t. *every* function in the class, as defined in the seminal work of [5], security is required only when the functions are distributed according to a distribution with sufficient min-entropy.

Definition 3 (Obfuscation of Point Functions with Multi-bit Output [9])**.** *A multi-bit point function (MBPF) obfuscator is a PPT algorithm $\mathcal{O}$ which takes as input values (k, m) describing a function $I_{(k,m)} \in \mathcal{I}$ and outputs a circuit C.*
Correctness: *For all $I_{(k,m)} \in \mathcal{I}$ with $|k| = n, |m| = \mathsf{poly}(n)$, all $x \in \{0,1\}^n$,*

$$\Pr[C(x) \neq I_{(k,m)}(x) \mid C \leftarrow \mathcal{O}(I_{(k,m)})] \leq \mathsf{negl}(n)$$

where the probability is taken over the randomness of the obfuscator algorithm.
Polynomial Slowdown: *For any k, m, the size of the circuit $C = \mathcal{O}(I_{(k,m)})$ is polynomial in $|k| + |m|$.*
Entropic Security: *We say that the obfuscator has $\alpha(n)$-**entropic security** if for any PPT adversary $\mathcal{A}$ with 1 bit output and any polynomial $\ell(\cdot)$, there exists a PPT simulator $\mathcal{S}$ such that for every distribution $\{X_n\}_{n\in\mathbb{N}}$, where X_n takes values in $\{0,1\}^n$ and $H_\infty(X_n) \geq \alpha(n)$, and for every message $m \in \{0,1\}^{\ell(n)}$,*

$$\Big|\Pr\left[\mathcal{A}(\mathcal{O}(I_{(k,m)})) = 1\right] - \Pr\left[\mathcal{S}^{I_{(k,m)}(\cdot)}(1^n) = 1\right]\Big| \leq \mathsf{negl}(n)$$

where the probability is taken over the randomness of $k \leftarrow X_n$, the randomness of the obfuscator $\mathcal{O}$ and the randomness of $\mathcal{A}, \mathcal{S}$.

Applying the work of [9] to our encryption scheme, gives us the stronger notion of *self-composable* obfuscation, defined below.

Definition 4 (Composability [9])**.** *A multi-bit point function obfuscator $\mathcal{O}$ with $\alpha(n)$-entropic security is said to be* self-composable *if for any PPT adversary $\mathcal{A}$ with 1 bit output and any polynomial $\ell(\cdot)$, there exists a PPT simulator $\mathcal{S}$ such that for every distribution $\{X_n\}_{n\in\mathbb{N}}$ with X_n taking values in $\{0,1\}^n$ and $H_\infty(X_n) \geq \alpha(n)$, and for every $m_1, \ldots, m_t \in \{0,1\}^{\ell(n)}$,*

$$|\Pr[\mathcal{A}(\mathcal{O}(I_{(k,m_1)}), \ldots, \mathcal{O}(I_{(k,m_t)})) = 1] - \Pr[\mathcal{S}^{I_{(k,m_1)}(\cdot),\ldots,I_{(k,m_t)}(\cdot)}(1^n) = 1]| \leq \mathsf{negl}(n)$$

where the probabilities are over $k \leftarrow X_n$ and over the randomness of $\mathcal{A}, \mathcal{S}, \mathcal{O}$.

Keeping Definitions 3 and 4 in mind, we can finally state our result.

Theorem 7. *There exists an obfuscator for the class of point functions with multibit output which is self-composable $\alpha(n)$-entropic secure under the $\mathsf{DLWE}_{\ell,q,\gamma}$ assumption, where $q = q(n)$ is super-polynomial, $\ell = \frac{\alpha(n)-\omega(\log n)}{\log q}$ and γ/q is negligible in n.*

As was mentioned above, the proof of this theorem follows from the tight connection between encryptions scheme that are secure w.r.t. weak keys, and multibit point function obfuscation. To state this connection formally, we need the following definition.

Definition 5 (Wrong-Key Detection [9, 12])**.** *We say that an encryption scheme $\mathbb{E} = (G, E, D)$ satisfies the* wrong-key detection *property if for all $k \neq k' \in \{0,1\}^n$, and every message $m \in \{0,1\}^{\mathsf{poly}(n)}$, $\Pr[D_{k'}(E_k(m)) \neq \bot] \leq \mathsf{negl}(n)$.*

Lemma 5. *[9] Let $\mathbb{E} = (G, E, D)$ be an encryption scheme with CPA security for $\alpha(n)$-weak keys and having the* wrong-key detection *property. We define the obfuscator $\mathcal{O}$ which, on input $I_{(k,m)}$, computes a ciphertext $c = E_k(m)$ and outputs the circuit $C_c(\cdot)$ (with hard-coded ciphertext c) defined by $C_c(x) = D_x(c)$. Then, $\mathcal{O}$ is a self-composable multi-bit point function obfuscator with $\alpha(n)$-entropic security.*

The proof of Theorem 7 follows immediately from Lemma 5, Theorem 6, and the following claim.

Claim 3. *The encryption scheme $\mathbb{E} = (G, E, D)$, defined in Section 4, has the wrong-key detection property.*

5.1 Proof of Claim 3

Let $\mathbf{s}, \mathbf{s}' \in \{0,1\}^n$ be any two distinct secret keys, and let $\mathbf{w} \in \{0,1\}^m$ be any message.

$$\Pr[D_{\mathbf{s}'}(E_{\mathbf{s}}(\mathbf{w})) \neq \perp] = \Pr\left[\mathbf{A}(\mathbf{s}-\mathbf{s}') + \mathbf{x} \in \left(\frac{-q}{8}, \frac{q}{8}\right)^m\right] \leq$$

where $\mathbf{A} \in_R \mathbb{Z}_q^{m \times n}$ and $\mathbf{x} \leftarrow (\overline{\Psi}_\beta)^m$. The fact that $\mathbf{s} \neq \mathbf{s}'$ implies that the vector $\mathbf{A}(\mathbf{s}-\mathbf{s}')$ is uniformly distributed in $\mathbb{Z}_q^m$, and thus the vector $\mathbf{A}(\mathbf{s}-\mathbf{s}')+\mathbf{x}$ is uniformly distributed in $\mathbb{Z}_q^m$. This implies that

$$\begin{aligned} \Pr\left[\mathbf{A}(\mathbf{s}-\mathbf{s}') + \mathbf{x} \in \left(-\frac{q}{8}, \frac{q}{8}\right)^m\right] &\leq \left(\frac{1}{4}\right)^m \\ &= \mathsf{negl}(n) \end{aligned}$$

as desired. □

References

[1] Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In *TCC*, pages 474–495, 2009.

[2] Michael Alekhnovich. More on average case vs approximation complexity. In *FOCS*, pages 298–307, 2003.

[3] Joel Alwen, Yevgeniy Dodis, and Daniel Wichs. Leakage-resilient public-key cryptography in the bounded-retrieval model. In *CRYPTO*, pages ??–??, 2009.

[4] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular secure encryption based on hard learning problems. In *CRYPTO*, pages ??–??, 2009.

[5] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In *CRYPTO*, pages 1–18, 2001.

[6] Victor Boyko. On the security properties of oaep as an all-or-nothing transform. In *CRYPTO*, pages 503–518, 1999.

[7] Ran Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In *CRYPTO*, pages 455–469, 1997.

[8] Ran Canetti, Yevgeniy Dodis, Shai Halevi, Eyal Kushilevitz, and Amit Sahai. Exposure-resilient functions and all-or-nothing transforms. In *EUROCRYPT*, pages 453–469, 2000.

[9] Ran Canetti, Yael Kalai, Mayank Varia, and Daniel Wichs. On symmetric encryption and point function obfuscation, 2009. Manuscript in Submission.

[10] Don Coppersmith. Finding a small root of a bivariate integer equation; factoring with high bits known. In *EUROCRYPT*, pages 178–189, 1996.

[11] Yevgeniy Dodis, Shafi Goldwasser, Yael Kalai, Chris Peikert, and Vinod Vaikuntanathan. Public-key encryption schemes with auxiliary inputs, 2009. Manuscript in Submission.

[12] Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett. On cryptography with auxiliary input. In *STOC*, pages 621–630, 2009.

[13] Yevgeniy Dodis, Shien Jin Ong, Manoj Prabhakaran, and Amit Sahai. On the (im)possibility of cryptography with imperfect randomness. In *FOCS*, pages 196–205, 2004.

[14] Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *FOCS*, pages 293–302, 2008.

[15] Oded Goldreich, Ronitt Rubinfeld, and Madhu Sudan. Learning polynomials with queries: The highly noisy case. *SIAM J. Discrete Math.*, 13(4):535–570, 2000.

[16] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. One-time programs. In *CRYPTO*, pages 39–56, 2008.

[17] Nadia Heninger and Hovav Shacham. Reconstructing rsa private keys from random key bits. In *CRYPTO*, pages 1–17, 2009.

[18] Russell Impagliazzo, Leonid A. Levin, and Michael Luby. Pseudo-random generation from one-way functions (extended abstracts). In *STOC*, pages 12–24, 1989.

[19] Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing

attacks. In *CRYPTO*, pages 463–481, 2003.

[20] Jonathan Katz and Vinod Vaikuntanathan. Signature schemes with bounded leakage. In *ASIACRYPT*, pages ??–??, 2009.

[21] Silvio Micali and Leonid Reyzin. Physically observable cryptography (extended abstract). In *TCC*, pages 278–296, 2004.

[22] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In *CRYPTO*, pages ??–??, 2009.

[23] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *STOC*, pages 333–342, 2009.

[24] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In *CRYPTO*, pages 554–571, 2008.

[25] Christophe Petit, François-Xavier Standaert, Olivier Pereira, Tal Malkin, and Moti Yung. A block cipher based pseudo random number generator secure against side-channel key recovery. In *ASIACCS*, pages 56–65, 2008.

[26] Krzysztof Pietrzak. A leakage-resilient mode of operation. In *EUROCRYPT*, pages 462–482, 2009.

[27] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93, 2005.

[28] Ronald L. Rivest. All-or-nothing encryption and the package transform. In *FSE*, pages 210–218, 1997.

[29] Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. Cryptology ePrint Archive, Report 2008/116, 2008.