Public-Key Encryption Schemes with Auxiliary Inputs

Yevgeniy Dodis
New York University
dodis@cs.nyu.edu

Shafi Goldwasser
MIT and Weizmann
shafi@csail.mit.edu

Yael Kalai
Microsoft Research
yael@microsoft.com

Chris Peikert
Georgia Tech
cpeikert@cc.gatech.edu

Vinod Vaikuntanathan
IBM Research
vinodv@alum.mit.edu

Abstract

We construct *public-key* cryptosystems that remain secure even when the adversary is given any *computationally uninvertible function* of the secret key as auxiliary input (even one that may reveal the secret key information-theoretically). Our schemes are based on the decisional Diffie-Hellman and Learning with Errors problems.

As two independent technical contributions, we extend the Goldreich-Levin theorem to provide a hard-core (pseudorandom) value over *large* fields, and show the security of the learning with errors assumption in the presence of specific (linear) auxiliary information about the secrets.

1 Introduction

Modern cryptographic algorithms are designed under the assumption that keys are perfectly secret and independently chosen for the algorithm at hand. Still, in practice, information about secret keys does get compromised for a variety of reasons, including side-channel attacks on the physical implementation of the cryptographic algorithm, or the use of the same secret key, or the same source of randomness for keys across several applications.

In recent years, starting with the works of [5, 17, 20], a new goal has been set within the theory of cryptography community to build general theories of physical security against large classes of side channel attacks. A large body of work has accumulated by now in which different classes of side channel attacks have been defined and different cryptographic primitives have been designed to provably withstand these attacks (See [5, 17, 20, 11, 1, 2, 22, 9, 26, 25, 16, 11, 12] and the references therein).

Placing the current paper within this body of work, we focus on side channel attacks which result from "memory leakages" [1, 9, 2, 22, 18]. In this class of attacks, the attacker chooses an arbitrary, efficiently computable function h (possibly as a function of the public parameters of the system), and receives the result of h applied on the secret key SK. Clearly, to have some secrecy left, we must restrict the attacker to choose a function h that "does not fully reveal the secret". The challenge is to model this necessary constraint in a clean and general manner, which both captures real attacks and makes the definition achievable. As of now, several models have appeared trying to answer this question.

Let k denote the length (or more generally, the min-entropy) of SK. Akavia, Goldwasser and Vaikuntanathan [1] considered a model in which the (adversarially chosen) function h has bounded output-length $\ell(k) < k$. In particular, this restriction ensures that the leakage does not fully reveal the secret-key. Akavia et al. [1] show that the public-key encryption scheme of Regev [27] is secure against $\ell(k)$-length bounded leakage functions as long as $\ell(k) < (1-\epsilon)k$ for some constant $\epsilon > 0$, under the intractability of the learning with error problem (LWE). Subsequent work of Naor and Segev [22] relaxed the restriction on h so that the length of the leakage observed by the adversary may be longer than the length of the secret key, but the min-entropy of the secret drops by at most $\ell(k) < k$ bits upon observing $h(SK)$ (which they call the *noisy leakage* requirement). The work of [22] also showed how to construct a public-key encryption scheme which resists noisy leakage as long as $\ell(k) < k - k^{\epsilon}$ for some constant $\epsilon > 0$, under the decisional Diffie Hellman (DDH) assumption. They also showed a variety of other public-key encryption schemes tolerating different amounts of leakage, each under a different intractability assumption: Paillier's assumption, the quadratic residuosity assumption, and more generally, the existence of any hash-proof system [7]. We refer the reader to [22] for a detailed discussion of these results. (Finally, we note that the proof of [1] based on the LWE assumption also generalizes to the case of noisy leakage.).

The bottomline is that both [1] and [22] (and the results that use the models therein) interpret the restriction on the leakage function h thus:

> Given $h(SK)$, it is (information-theoretically) *impossible* to recover SK.

1.1 The Auxiliary Input Model

The natural question that comes out of the modeling in [1, 22] is whether this restriction is essential. For example, is it possible to achieve security against a leakage function h that is a one-way permutation? Such a function information-theoretically reveals the entire secret-key SK, but still it is computationally infeasible to recover the secret key from $h(SK)$.

The focus of this work is the model of *auxiliary input* leakage functions, introduced by Dodis, Kalai, and Lovett [9], generalizing [1, 22]. They consider the case of *symmetric encryption* and an adversary who can learn an arbitrary polynomial time computable function h of the secret key, provided that the secret key SK is *computationally* hard to compute given $h(SK)$ (but not necessarily information theoretic hard to determine, as implied by the definitions of [1, 22]). Formally, [9] require that any polynomial time algorithm attempting to invert $h(SK)$ will succeed with probability at most $2^{-\lambda(k)}$ for some function $\lambda(\cdot)$ (i.e., a smaller $\lambda(k)$ yields a larger class of functions allowed, and thus a stronger security result). The ultimate goal is to capture all polynomially uninvertible functions: namely, all functions for which the probability of inversion by a polynomial time algorithm is bounded by some *negligible* function in k.[1] [2]

The work of [9] constructed, based on a non-standard variant of the learning parity with noise (LPN) assumption, a symmetric-key encryption scheme that remains CCA-secure w.r.t. any auxiliary input $h(SK)$, as long as no polynomial time algorithm can invert h with probability more than $2^{-\epsilon k}$ for some $\epsilon > 0$.

In the same work [9], it was observed that in addition to generalizing the previous leakage models, the auxiliary input model offers additional advantages, the main one being composition. Consider a setting where a user prefers to use the same secret key for multiple tasks, as is the case when using biometric keys [4, 10]. Suppose we construct an encryption scheme that is secure w.r.t. any auxiliary input which is an uninvertible function of the secret key. Then, one can safely use his secret and public key pair, to run arbitrary protocols, as long as these protocols together do not reveal (computationally) the entire secret key.

1.2 Our Results

In this paper, we focus on designing *public key* encryption algorithms which are secure in the presence of auxiliary input functions. We adapt the definition of security w.r.t. auxiliary input from [9] to the case of public-key encryption algorithms.

1. To address the issue of whether the function h is chosen by the adversary after seeing the corresponding public-key PK (so called adaptive security in [1]) we allow the adversary to receive $h(SK, PK)$. This, in effect, amounts to choosing the function of SK to depend on PK.

2. Note that in the public key setting, the public key itself leaks some information about the secret key. Therefore, in the public key setting, it would be impossible to prove that the scheme remains secure w.r.t. any leakage function h that is an uninvertible function of the secret key. However, what we can hope for is to get security w.r.t. any leakage function h, such that the function that outputs both $h(SK, PK)$ and PK, is an uninvertible function. Indeed, we construct an encryption scheme that achieves such security guarantees; see item 3 below.

We prove auxiliary input security results for three public-key encryption schemes based on different assumptions.

[1] Here, $\mathrm{negl}(k)$ is defined as usual to be smaller than $\frac{1}{k^c}$ for every $c > 0$ for every sufficiently large k.

[2] It is instructive to contrast uninvertible functions with the standard notion of one-way functions. In the former, we require the adversary who is given $h(SK)$ to come up with the actual pre-image SK itself, whereas in the latter, the adversary need only output an SK' such that $h(SK') = h(SK)$. Thus, a function h that outputs nothing is an uninvertible function, but not one-way! The right notion to consider in the context of leakage and auxiliary input is that of uninvertible functions.

1. We show that the "dual" of Regev's encryption scheme [27], first proposed by Gentry, Peikert and Vaikuntanathan [14], when suitably modified is CPA-secure in the presence of auxiliary input functions h that can be inverted with probability *at most* 2^{-k^ϵ} for any $\epsilon > 0$. The underlying hard problem for our auxiliary-input CPA-secure scheme is the same as that for (standard) CPA-security, i.e, the "learning with error" (LWE) problem. This result, in particular, implies that the dual scheme is secure w.r.t. bounded length leakage of size at most $k - k^\epsilon$ (and more generally, to noisy leakages). This improves on the previous bound proved in [1] for the [27] system.

2. We show that the encryption scheme of Boneh, Halevi, Hamburg and Ostrovsky [3] (henceforth called the BHHO encryption scheme), suitably modified, is CPA-secure in the presence of auxiliary input functions h that can be inverted with probability *at most* 2^{-k^ϵ} for any $\epsilon > 0$. The underlying hard problem for our auxiliary-input CPA-secure scheme is again the same as that of the original BHHO scheme, i.e, the decisional Diffie-Hellman assumption. Previously, [22] showed that the BHHO scheme is secure w.r.t. bounded length leakage of size at most $k - k^\epsilon$ (or, more generally, noisy leakages).

   We note that we can prove security of both the dual Regev encryption scheme and the BHHO encryption scheme w.r.t. a richer class of auxiliary inputs, i.e., those that are hard to invert with probability $2^{-\mathrm{polylog}(k)}$. However, then the assumptions we rely on are that LWE (or DDH) are secure against an adversary that runs in subexponential time.

3. We introduce a new public-key encryption system, denoted sparse-GPV, which is essentially a sparse version of the dual Regev (a.k.a, GPV) scheme. We prove that assuming the intractability of the LWE problem, sparse-GPV is CPA-secure in the presence of all auxiliary input functions h for which the function $H(SK, PK) = (h(SK, PK), PK)$ is an uninvertible function. Sparse-GPV is a modification of the dual Regev encryption scheme, so that the secret key is chosen not uniformly from $\{0,1\}^m$ nor from a discrete Gaussian distribution, but rather is a 'sparse' random Boolean vector $\mathbf{e} \in \{0,1\}^m$ which has fewer 1s than the dimension of the underlying LWE instance (usually denoted by n).

   This in particular implies that sparse-GPV is CPA-secure w.r.t. bounded-length leakage (or more generally, noisy leakage) of length $(1-\epsilon)n$.

The security guarantees of the sparse-GPV scheme deserves some interpretation. At first glance, it seems that the fact that we can tolerate any auxiliary input $h(SK, PK)$, such that together with PK it is hard to find SK, is a very strong result in the context of leakage. However, taking a closer look, one can see that this intuition is misleading. The reason is that the public key PK may reveal substantial information about SK, and therefore it may be the case that even bounded-length leakage functions (as in [1]), together with the public key, are not one-way anymore. Indeed, as noted above, the sparse-GPV scheme is only secure w.r.t. bounded length leakage of size $(1-\epsilon)k$, whereas the other two schemes are secure w.r.t. bounded length leakage of size $k - k^\epsilon$.

However, the sparse-GPV scheme is advantageous in the context of composition. One can envision that to reduce the complexity of the public-key infrastructure, it may be desirable to design cryptographic systems such that the same public key (and corresponding secret key) may be used for several applications, including for example encryption, digital signatures, and identification. Interestingly, going back in time to the original Diffie-Hellman and RSA papers [8, 28], the term public-key cryptography implicitly referred to a system which offers at the same time both encryption and digital signatures using the same public/secret key pair. Efficiency aside, the question is whether such proposed practice offers *security* against polynomial time adversaries in

the sense that we have become accustomed to. Indeed, proving security against a large enough class of auxiliary input functions has direct implications on this question. In particular, a public-key encryption algorithm Π that is CPA-secure with any auxiliary input $h(SK, PK)$ (such that $H(SK, PK) = (h(SK, PK), PK)$ is a one-way function) implies that Π can be securely composed with any other cryptographic algorithm using the same public and secret keys that was previously proved secure in the standard (no leakage) sense.

We end this section by remarking that the complexity of all of the the encryption schemes above depends on the bound on the inversion probability of h which we desire to achieve. For example, in the case of the BHHO scheme, the size of the secret key (and the complexity of encrypting and decrypting) is $k^{1/\epsilon}$, where k is the length of the secret-key (or more generally, the min-entropy) and security is w.r.t. auxiliary inputs that are hard to invert with probability 2^{-k^ϵ}. (This is also the case for the results of [22].)

1.3 Overview of Techniques

We sketch the main ideas behind the auxiliary input security of the GPV encryption scheme (slightly modified). The scheme is based on the hardness of the learning with error (decisional LWE) problem, which states that for a security parameter n and any polynomially large m, given a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, the vector $\mathbf{A}^T\mathbf{s} + \mathbf{x}$ is pseudorandom where $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ is uniformly random, and each component of $\mathbf{x}$ is chosen from a "narrow error distribution".

Let us first recall how the GPV scheme works. The secret-key in the scheme is a vector $\mathbf{e} \in \{0,1\}^m$, and the public-key is a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ together with $\mathbf{u} = \mathbf{A}\mathbf{e} \in \mathbb{Z}_q^n$. Here n is the security parameter of the system, q is a prime (typically polynomial in n, but in our case slightly superpolynomial), and m is a sufficiently large polynomial in n and $\log q$. (The min-entropy of the secret-key k, in this case, is m.) The basic encryption scheme proceeds bit by bit. To encrypt a bit b using PK, we first sample $\mathbf{A}^T\mathbf{s} + \mathbf{x}$ from the LWE distribution: i.e, choose $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ uniformly at random and $\mathbf{x}$ from the error distribution, and output $\mathbf{A}^T\mathbf{s} + \mathbf{x}$. The encryption algorithm samples the number $x' \in \mathbb{Z}_q$ from the error distribution and outputs the ciphertext

$$(\mathbf{A}^T\mathbf{s} + \mathbf{x}, \mathbf{u}^T\mathbf{s} + x' + b\lfloor q/2 \rfloor) \in \mathbb{Z}_q^m \times \mathbb{Z}_q$$

Given the secret-key $\mathbf{e}$, the decryption algorithm computes $\mathbf{e}^T(\mathbf{A}^T\mathbf{s} + \mathbf{x}) \approx (\mathbf{A}\mathbf{e})^T\mathbf{s} = \mathbf{u}^T\mathbf{e}$ (since $\mathbf{e}^T\mathbf{x}$ and x' are all small compared to q) and uses this to recover b from the second component.

The first idea we use to show auxiliary input security for this scheme is that the intractability assumption (i.e, the hardness of LWE mentioned above) "almost entirely" refers to the pre-amble of the ciphertext $\mathbf{A}^T\mathbf{s} + \mathbf{x}$, and not to the secret-key at all. This suggests constructing the simulator in the security proof, such that it knows the secret-key while running the simulation: the added advantage in the context of leakage is that knowing the secret-key enables the simulator to simulate arbitrary (polynomial-time computable) leakage functions h. This technique for proving security of public-key encryption was already used in the context of leakage in [22, 18], and to our knowledge, traces as far back as the Cramer-Shoup CCA encryption [6]. In particular, we consider an alternate encryption scheme that, informally, encrypts the bit b using the secret-key $\mathbf{e}$ as follows:

$$\text{Output } (\mathbf{y} = \mathbf{A}^T\mathbf{s} + \mathbf{x}, \mathbf{e}^T\mathbf{y} + x' + b\lfloor q/2 \rfloor)$$

The distribution thus produced is "almost as good as" the original encryption algorithm: in particular, $\mathbf{e}^T\mathbf{y} + x' = \mathbf{u}^T\mathbf{s} + (\mathbf{e}^T\mathbf{x} + x')$, where $\mathbf{e}^T\mathbf{x} + x'$ is distributed almost like a sample from the noise distribution if $\mathbf{e}^T\mathbf{x}$ is negligible compared to x' (this is true since both $\mathbf{e}$ and $\mathbf{x}$ can be chosen

to be negligible compared to the magnitude of x'). Now, by the LWE assumption, we might as well $\mathbf{y}$ replace $\mathbf{y}$ with a uniformly random vector over $\mathbb{Z}_q^m$.

We would like to show that the second component of the ciphertext, which now consists of $\mathbf{e}^T\mathbf{y} = \langle \mathbf{e}, \mathbf{y} \rangle$ is pseudorandom, given the view of the adversary, namely $(\mathbf{A}, \mathbf{Ae}, h(\mathbf{A}, \mathbf{e}), \mathbf{y})$ (where $\mathbf{y} \in \mathbb{Z}_q^m$ is uniformly random). Assuming that the function $h'(\mathbf{A}, \mathbf{e}) = (\mathbf{A}, \mathbf{Ae}, h(\mathbf{A}, \mathbf{e}))$ is uninvertible, this suggests using a *Goldreich-Levin type theorem over the large field* $GF(q)$. Coming up with such a theorem is the first technical contribution of this work.

The original Goldreich-Levin theorem proves that for every uninvertible function h, $\langle \mathbf{e}, \mathbf{y} \rangle \pmod 2$ is pseudorandom, given $h(\mathbf{e})$ and $\mathbf{y}$ for a uniformly random $\mathbf{y} \in GF(2)^n$. In this setting, the later work of [15] extends this result to deal with inner-products over a general prime modulus q, rather than $q = 2$. In particular, they show that any PPT algorithm that distinguishes between $\langle \mathbf{y}, \mathbf{e} \rangle \pmod q$ and uniform, given $h(\mathbf{e})$ and $\mathbf{y}$, gives rise to an $\mathsf{poly}(q)$*-time* algorithm that inverts $h(\mathbf{e})$ with probability $1/\mathsf{poly}(q)$ (for a more detailed comparison of our result with [15], see Remark A.3). For superpolynomially large q, the running-time of the inverter is superpolynomial, which we wish to avoid. We consider a special class of functions (which is exactly what is needed in our applications) where $\mathbf{e}$ comes from a smaller domain $H^m \subseteq \mathbb{Z}_q^m$, i.e, each co-ordinate of $\mathbf{e}$ is chosen from a domain H that is much smaller than the entire $\mathbb{Z}_q$. For this class of functions, we show how to make the running-time of the inverter *polynomial in* n (and independent of q). We state the result informally below.

Informal Theorem 1. *Let q be prime, and let H be an polynomial size subset of $GF(q)$. Let $f : H^n \to \{0,1\}^*$ be any (possibly randomized) function. If there is a PPT algorithm $\mathcal{D}$ that distinguishes between $\langle \mathbf{e}, \mathbf{y} \rangle$ and uniform distribution on the range given $h(\mathbf{e})$ and $\mathbf{y}$, there is a PPT algorithm $\mathcal{A}$ that inverts $h(\mathbf{e})$ with probability roughly $1/(q^2 \cdot \mathsf{poly}(n, 1/\epsilon))$.*

Applying this variant of the Goldreich-Levin theorem over $GF(q)$, we get security against auxiliary input functions h that are hard to invert given $(\mathbf{A}, \mathbf{Ae}, h(\mathbf{A}, \mathbf{e}))$ (called weak auxiliary-input security in the rest of the paper). Obtaining strong auxiliary input security, i.e, security against functions h that are hard to invert given only $(\mathbf{A}, h(\mathbf{A}, \mathbf{e}))$, is very easy in our case: since the public key $\mathbf{Ae}$ has length $n \log q = m^\epsilon \ll |SK|$, one could simply guess $PK = \mathbf{Ae}$ and lose only 2^{-m^ϵ} in the inversion probability.

The proof of security for the BHHO encryption scheme follows precisely the same line of argument, but with two main differences: (1) the proof is somewhat simpler because one does not have to deal with pesky error terms as in the LWE case, and (2) we use the Goldreich-Levin theorem over an exponentially large field $GF(q)$, rather than a superpolynomial one.

Finally, we consider a variant of the GPV encryption scheme where the secret-key is a (random) sparse vector over $\{0,1\}^m$ (with exactly $n/2$ ones) rather than uniformly random in $\{0,1\}^m$. For this variant, we show security against any h that is *polynomially uninvertible* given $PK = (\mathbf{A}, \mathbf{Ae})$ and $h(\mathbf{A}, \mathbf{e})$. The main difference in the proof is in the first step where we "encrypt with the secret-key". In the GPV encryption scheme, we had to choose a superpolynomial modulus to make sure that this alternative encryption algorithm produces a distribution that is statistically close to the original encryption algorithm. We observe that in fact, we can perfectly simulate the distribution of the original encryption algorithm (while simultaneously encrypting with the secret-key) if the alternate encryption algorithm also had access to a pair $(\mathbf{e}, \mathbf{e}^T\mathbf{x})$, i.e, a (sparse) linear function of the error-vector $\mathbf{x}$ in the first component of the ciphertext. To make the proof go through with this observation, we prove the following lemma, which states that the LWE assumption is true even if the adversary gets (as auxiliary information) an arbitrary, sparse, linear function of the error-vector $\mathbf{x}$. This is the second independent technical contribution of this paper.

Informal Theorem 2. *Assuming that the LWE assumption for security parameter $n/2$ holds. Then,*

$$\{\mathbf{A}, \mathbf{A}^T\mathbf{s} + \mathbf{x}, \mathbf{e}, \mathbf{e}^T\mathbf{x}\} \approx \{\mathbf{A}, \mathbf{u}, \mathbf{e}, \mathbf{e}^T\mathbf{x}\}$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$, $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, $\mathbf{x}$ comes from the error distribution, $\mathbf{u} \leftarrow \mathbb{Z}_q^m$ is uniformly random, and $\mathbf{e} \leftarrow \{0,1\}^m$ such that the number of ones is exactly $n/2$.

Details are deferred to Section C.2 (See Corollary C.8).

2 Preliminaries

Throughout this paper, we denote the security parameter by n. We write $\mathrm{negl}(n)$ to denote an arbitrary negligible function, i.e., one that vanishes faster than the inverse of any polynomial.

The Decisional Diffie Hellman Assumption. Let $\mathcal{G}$ be a probabilistic polynomial-time "group generator" that, given the security parameter n in unary, outputs the description of a group G that has prime order $q = q(n)$. The decisional Diffie Hellman (DDH) assumption for $\mathcal{G}$ says that the following two ensembles are computationally indistinguishable:

$$\left\{(g_1, g_2, g_1^r, g_2^r) : g_i \leftarrow G, r \leftarrow \mathbb{Z}_q\right\} \approx_c \left\{(g_1, g_2, g_1^{r_1}, g_2^{r_2}) : g_i \leftarrow G, r_i \leftarrow \mathbb{Z}_q\right\}$$

We will use a lemma of Naor and Reingold [21] which states that a natural generalization of the DDH assumption which considers $m > 2$ generators is actually equivalent to DDH. The proof follows from the self-reducibility of DDH.

Lemma 2.1 ([21])**.** *Under the DDH assumption on $\mathcal{G}$,*

$$\left\{(g_1, \ldots, g_m, g_1^r, \ldots, g_m^r) : g_i \leftarrow G, r \leftarrow \mathbb{Z}_q\right\} \approx_c \left\{(g_1, \ldots, g_m, g_1^{r_1}, \ldots, g_m^{r_m} : g_i \leftarrow G, r_i \leftarrow \mathbb{Z}_q\right\}$$

Learning with Errors. We defer the definition of the learning with errors (LWE) assumption to Appendix B.

3 Security against Auxiliary Inputs

We start by defining a general notion of security of (public-key) encryption schemes w.r.t. auxiliary input.

Definition 1. A public-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M} = \{\mathcal{M}_n\}_{n\in\mathbb{N}}$ is `CPA secure w.r.t. auxiliary inputs from` $\mathcal{H}$ if for any PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, any function $h \in \mathcal{H}$, any polynomial p, and any sufficiently large $n \in \mathbb{N}$,

$$\mathsf{Adv}_{\mathcal{A},\Pi,h} \stackrel{def}{=} \left|\Pr[\mathrm{CPA}_0(\Pi, \mathcal{A}, n, h) = 1] - \Pr[\mathrm{CPA}_1(\Pi, \mathcal{A}, n, h) = 1]\right| < \frac{1}{p(n)},$$

where $\mathrm{CPA}_b(\Pi, \mathcal{A}, n, h)$ is the output of the following experiment:

$(SK, PK) \leftarrow \mathsf{Gen}(1^n)$
$(M_0, M_1, \mathsf{state}) \leftarrow \mathcal{A}_1(1^n, PK, h(SK, PK))$ s.t. $|M_0| = |M_1|$
$c_b \leftarrow \mathsf{Enc}(PK, M_b)$
Output $\mathcal{A}_2(c_b, \mathsf{state})$

3.1 Classes of Auxiliary Input Functions

Of course, we need to decide which function families $\mathcal{H}$ we are going to consider. We define three such families. For future convenience, we will parametrize these families by the min-entropy k of the secret key, as opposed to the security parameter n. (Note, in our schemes the secret key will be random, so k is simply the length of the secret key.) The first family $\mathcal{H}_{\mathtt{bdd}}$ is the length-bounded family studied by the prior work [1, 22],[3] while the last two families $\mathcal{H}_{\mathtt{ow}}, \mathcal{H}_{\mathtt{pk\text{-}ow}}$ are the auxiliary-input families we study and introduce in this work, where we only assume that the secret key is "hard to compute" given the leakage.

- Let $\mathcal{H}_{\mathtt{bdd}}(\ell(k))$ be the class of all polynomial-time computable functions $h : \{0,1\}^{|SK|+|PK|} \to \{0,1\}^{\ell(k)}$, where $\ell(k) \le k$ is the number of bits the attacker is allowed to learn.
- Let $\mathcal{H}_{\mathtt{ow}}(f(k))$ be the class of all polynomial-time computable functions $h : \{0,1\}^{|SK|+|PK|} \to \{0,1\}^*$, such that given $h(SK, PK)$ (for a randomly generated (SK, PK)), no PPT algorithm can find SK with probability greater than $f(k)$, where $f(k) \ge 2^{-k}$ is the hardness parameter. Our goal is to make $f(k)$ as large (i.e., as close to $\mathrm{negl}(k)$) as possible.
- Let $\mathcal{H}_{\mathtt{pk\text{-}ow}}(f(k))$ be the class of all polynomial-time computable functions $h : \{0,1\}^{|SK|+|PK|} \to \{0,1\}^*$, such that given $(PK, h(SK, PK))$ (for a randomly generated (SK, PK)), no PPT algorithm can find SK with probability greater than $f(k)$, where $f(k) \ge 2^{-k}$ is the hardness parameter. Our goal is to make $f(k)$ as large (i.e., as close to $\mathrm{negl}(k)$) as possible.

This leads to the following definition.

Definition 2. A public-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is said to be

- $\ell(k)$-`LB-CPA (length-bounded CPA) secure` if it is CPA secure w.r.t. family $\mathcal{H}_{\mathtt{bdd}}(\ell(k))$.
- $f(k)$-`AI-CPA (auxiliary input CPA) secure` if it is CPA secure w.r.t. family $\mathcal{H}_{\mathtt{ow}}(f(k))$.
- $f(k)$-`wAI-CPA (weak auxiliary input CPA) secure` if it is CPA secure w.r.t. family $\mathcal{H}_{\mathtt{pk\text{-}ow}}(f(k))$.

We show the following four relations between these definitions, whose proof is given in Appendix D.

Lemma 3.1. *Assume Π is a public-key encryption scheme whose public key is of length $t(k)$.*

1. *If Π is $f(k)$-`AI-CPA secure`, then Π is $f(k)$-`wAI-CPA secure`.*
2. *If Π is $f(k)$-`wAI-CPA secure`, then Π is $(2^{-t(k)} f(k))$-`AI-CPA secure`.*
3. *If Π is $f(k)$-`AI-CPA secure`, then Π is $(k - \log(1/f(k)))$-`LB-CPA secure`.*
4. *If Π is $f(k)$-`wAI-CPA secure`, then Π is $(k - t(k) - \log(1/f(k)))$-`LB-CPA secure`.*

We now examine our new notions or strong and weak auxiliary input security ($f(k)$-`AI-CPA` and $f(k)$-`wAI-CPA`, respectively).

[3] For simplicity, we do not define a more general family corresponding to noisy leakage model of [22]. However, all the discussion, including Lemma 3.1, easily holds for noisy-leakage instead of length-bounded leakage.

Strong Notion. We start with $f(k)$-AI-CPA security, which is the main notion we advocate. It states that as long as the leakage $y = h(SK, PK)$ did not reveal SK (with probability more than $f(k)$), the encryption remains secure. First, it cleanly generalizes the notion of auxiliary input security of [9] in the symmetric-key setting, since there k is simultaneously the security parameter, length of the secret key and its min-entropy. Second, as shown in Part 3. of Lemma 3.1, it immediately implies than it is safe to leak $(k - \log(1/f(k)))$ arbitrary bits about the secret key. Thus, if $\log(1/f(k)) \ll k$, it means that we can leak almost the entire (min-)entropy of the secret key! This motivates our convention of using k as the min-entropy of the secret key, making our notion intuitive to understand in the leakage setting of [1, 22]. Third, it implies very strong composition properties. As long as other usages of SK make it $f(k)$-hard to compute, these usages will not break the security of our encryption scheme.

Weak Notion. We next move to the more subtle notion of $f(k)$-wAI-CPA security. The difference from the strong security comes from the fact that we futher restrict the class of allowed leakage functions to that SK remains hard given both the leakage y and the public key PK. While this might sound natural, it has the following unexpected "anti-monotonicity" property. By making PK contain *more information* about the secret key, we could sometimes make the scheme *more secure* w.r.t. to this notion (i.e., the function $f(k)$ becomes larger), which seems unnatural. At the extreme, setting $PK = SK$ would make the scheme wAI-CPA "secure", since we now ruled out all "legal" auxiliary functions, making the notion vacuously true. While this might appear a syntactic problem, one can have more convincing examples, such as the following example below.

Example. Consider an encryption scheme Π which is $f(k)$-wAI-CPA secure, but insecure for slighly larger $f(k) < f'(k) \ll \sqrt{f(k)}$. Consider the scheme Π' taking two independent copies Π_1 and Π_2 of Π, and encrypting the message M by randomly splitting it as $M = M_1 \oplus M_2$ and encrypting each M_i with Π_i. This scheme might not be $f(k)$-wAI-CPAsecure, since the "direct product" leakage function $h(SK_1, SK_2) = (h_1(SK_1), h_2(SK_2))$ might be $f(k)$-hard to invert, despite both $h_1(SK_1)$ and $h_2(SK_2)$ being "$f'(k)$-easy". Thus, using such a "legal" h the attacker could recover both M_1 and M_2, and, hence, M. Now, consider a modified scheme Π'', where the public key explicitly contains the value SK_2. Here, conditioned on this public key, which includes SK_2, any $f(k)$-hard function of (SK_1, SK_2) must also be $f(k)$-hard function of SK_1. Thus, $f(k)$-wAI-CPA security of Π_1 implies that no information about M_1, and, hence, M is leaked. This means that Π'' is still $f(k)$-wAI-CPA secure, despite the fact that an intuitively "more secure" scheme Π' is not! □

Although the above property shows that the wAI-CPA security should be taken with care, we show it is still very useful. First, Lemma 3.1 shows that it is useful *when the scheme has a short public-key*. Specificially, when $t \ll \log(1/f(k))$, wAI-CPA security implies pretty good AI-CPA and LB-CPA security. In particular, this will be the case for all the schemes that we construct, where we will first show good wAI-CPA security, and then deduce almost the same AI-CPA security. Second, even if the scheme does not have a very short public-key, wAI-CPA security might be useful in composing different schemes *sharing the same public-key infrastructure*. For example, assume we have a signature and an encryption scheme having the same pair (PK, SK). And assume that the signature scheme is shown to be $f(k)$-secure against key recovery attacks. Since the auxiliary information obtained by using the signature scheme certainly includes the public key, we can conclude that our $f(k)$-wAI-CPA secure encryption scheme is still secure, despite being used together with the signatures scheme. In other words, while strong auxiliary input security would allow us to safely compose with any $f(k)$-secure signature scheme, even using a different PK, weak auxiliary input security is still enough when the PKI is shared, which is one of the motivating

settings for auxiliary input security.

Still, one might ask why to go for wAI-CPA security and not for the stronger AI-CPA security. The answer is that we migth be able to achieve a much smaller value of $f(k)$ in the latter case. Indeed, the best AI-CPA security we will manage to achieve in this work is $f(k) = 2^{-k^\epsilon}$,[4] while we will construct a wAI-CPA secure scheme with optimal value $f(k) = \text{negl}(k)$!

Public Parameters. For simplicity, in Definition 2 we did not consider the case when the encryption schemes might depend on system-wide parameters params. However, the notions of strong and weak auxiliary input security naturally generalize to this setting, as follows. First, to allow realistic attacks, the leakage function h can als depend on the parameters. Second, for both AI-CPA and wAI-CPA notions, SK should be hard to recover given params and $h(SK, PK, \texttt{params})$ (resp. $(PK, h(SK, PK, \texttt{params}))$).[5] Correspondingly, when applying Lemma 3.1, the length of the parameters is not counted towards the length $t(k)$ of the public-key.

4 Goldreich-Levin Theorem for Large Fields

In this section, we prove a Goldreich-Levin theorem over any field $GF(q)$ for a prime q. In particular, we show:

Theorem 4.1. *Let q be prime, and let H be an arbitrary subset of $GF(q)$. Let $f : H^n \to \{0,1\}^*$ be any (possibly randomized) function. If there is a distinguisher $\mathcal{D}$ that runs in time t such that*

$$\Big|\Pr[\mathbf{s} \leftarrow H^n, y \leftarrow f(\mathbf{s}), \mathbf{r} \leftarrow GF(q)^n : \mathcal{D}(y, \mathbf{r}, \langle \mathbf{r}, \mathbf{s}\rangle) = 1] - \Pr[\mathbf{s} \leftarrow H^n, y \leftarrow f(\mathbf{s}), \mathbf{r} \leftarrow GF(q)^n, u \leftarrow GF(q) : \mathcal{D}(y, \mathbf{r}, u) = 1]\Big| = \epsilon$$

then there is an inverter $\mathcal{A}$ that runs in time $t' = t \cdot \mathsf{poly}(n, |H|, 1/\epsilon)$ such that

$$\Pr[\mathbf{s} \leftarrow H^n, y \leftarrow f(\mathbf{s}) : \mathcal{A}(y) = \mathbf{s}] \geq \frac{\epsilon^3}{512 \cdot n \cdot q^2} \tag{1}$$

Remark 4.2. Assume that the distinguisher $\mathcal{D}$ is a PPT algorithm and the distinguishing advantage ϵ is non-negligible in n. When q is polynomial in n, the running time of $\mathcal{A}$ is polynomial, and the success probability is inverse-polynomial in n, irrespective of H. When q is super-polynomial in n, but $|H|$ *is polynomial in* n, the running time of $\mathcal{A}$ *remains polynomial in* n, but the success-probability is dominated by the $1/q^2$ factor.

Overview of the Proof. We describe the main ideas behind the proof and defer the full proof to Appendix A. As in the standard proof, we can concentrate on the vectors $\mathbf{s}$ on which the distinguisher has $\Omega(\epsilon)$-advantage for random $\mathbf{r}$. As a manner of simple algebraic manipulation, it is also easy to see (see Appendix A) that it suffices to build an inverter $\mathcal{A}$ for such vectors

[4] Moreover, it appears hard to break this barrier using standard polynomial-type assumptions. Informally, under such assumptions the public key must leak at least k^ϵ bits of information about the secret key. This suggests that some leakage function h could leak precisely the remaining $k - k^\epsilon$ bits of information, allowing one to break the CPA security (using also PK), while keeping this leakage 2^{-k^ϵ}-hard in the AI-CPA setting.

[5] Notice, unlike the case of wAI-CPA security, the inclusion of params as part of the leakage does not result in the "anti-monotonicity" problem discussed earlier, since params are independent of the secret key.

$\mathbf{s}$ which succeeds with probability at least $1/2q^c$, where $c \geq 2$ is the smallest integer such that $q^c > 128|H|n/\epsilon^2$. Also as in the standard proof, our inverter $\mathcal{A}$ will guess c inner products $\langle \mathbf{s}, \mathbf{z}_i \rangle$, for random vectors $\mathbf{z}_1 \ldots \mathbf{z}_c$, losing $1/q^c$ factor in the process. Thus, we only need to show how to compute $\mathbf{s}$ with probability $1/2$, provided: (a) the inverter $\mathcal{A}$ knows c inner products of $\mathbf{s}$ with random vectors; and (b) the distinguisher $\mathcal{D}$ has advantage $\Omega(\epsilon)$ in distinguishing correct inner product values from random vectors u.

This is where we differ from the standard proof, and use the fact that $|H|$ might be smaller than q. For each $i = 1 \ldots n$ and $a \in H$, we design an iterative procedure to test, with high probability if $s_i = a$. This is why our running time only depends on $|H|$ and not q. The details of this test $T_{i,a}$, which is the crux of the argument (in particular, why the existence of the *distinguisher* is enough to design $T_{i,a}$), is given in Appendix A.

5 Auxiliary Input Secure Encryption Schemes

We show that the BHHO encryption scheme is secure against subexponentially hard-to-invert auxiliary input in section 5. The proof of auxiliary input security for the GPV and the sparse GPV cryptosystems are given in Appendix C, for lack of space.

This section is organized as follows: first, we present the cryptosystem of Boneh, Halevi, Hamburg and Ostrovsky [3]. We then show that the scheme is secure against subexponentially hard auxiliary inputs, under the decisional Diffie-Hellman (DDH) assumption.

5.1 The BHHO Cryptosystem

Let n be the security parameter. Let $\mathcal{G}$ be a probabilistic polynomial-time "group generator" that, given the security parameter n in unary, outputs the description of a group G that has prime order $q = q(n)$.

$\mathsf{KeyGen}(1^n, \epsilon)$: Let $m := (4 \log q)^{1/\epsilon}$, and let $G \leftarrow \mathcal{G}(1^n)$. Sample m random generators $g_1, \ldots, g_m \leftarrow G$. Let $\mathbf{g} = (g_1, \ldots, g_m)$. Choose a uniformly random m-bit string $\mathbf{s} = (s_1, \ldots, s_m) \in \{0,1\}^m$, and define

$$y := \prod_{i=1}^{m} g_i^{s_i} \in G$$

Let the secret key $SK = \mathbf{s}$, and let the public key $PK = (\mathbf{g}, y)$ (plus the description of G). Note, $\mathbf{g}$ can be viewed as public parameters, meaning only y can be viewed as the "user-specific" public key.

$\mathsf{Enc}(PK, M)$: Let the message $M \in G$. Choose a uniformly random $r \in \mathbb{Z}_q$. Compute $f_i := g_i^r$ for each i, and output the ciphertext

$$C := (f_1, \ldots, f_m, y^r \cdot m) \in G^{m+1}.$$

$\mathsf{Dec}(SK, C)$: Parse the ciphertext C as $(f_1, \ldots, f_m, c)$, and the secret key $SK = (s_1, \ldots, s_m)$. Output

$$M' := c \cdot \left(\prod_{i=1}^{m} f_i^{s_i} \right)^{-1} \in G$$

To see the correctness of the encryption scheme, observe that if $f_i = g_i^r$ for all i, then the decryption algorithm outputs

$$M' = c \cdot \left(\prod_{i=1}^{m} f_i^{s_i} \right)^{-1} = c \cdot \left(\prod_{i=1}^{m} g_i^{s_i} \right)^{-r} = c \cdot y^{-r} = M$$

5.2 Security for Subexponentially Hard-to-Invert Auxiliary Inputs

Theorem 5.1. *Assuming that the Decisional Diffie-Hellman problem is hard for $\mathcal{G}$, the encryption scheme described above is (2^{-m^ϵ})-*`AI-CPA` *secure (when $\mathbf{g}$ is viewed as a public parameter).*

Remark. We can actually handle a richer class of auxiliary inputs. We can prove security even for auxiliary functions $h(\mathbf{g}, \mathbf{s})$ that (given $\mathbf{g}$) are hard to invert with probability $1/2^k$, where k can be as small as polylog(m). However, then the assumption we rely on is that DDH is hard for adversaries that runs in subexponential time. For the sake of simplicity, we only state the theorem for $k = m^\epsilon$ in which case we can rely on the standard DDH hardness assumption.

Proof of Theorem 5.1. By Lemma 3.1 (Part 2.) and because the length of "user-specific" public-key y is $\log q$ bits, to show Theorem 5.1 it suffices to show that our encryption scheme is $(q2^{-m^\epsilon})$-`wAI-CPA` secure. Fix any auxiliary-input function h, so that $\mathbf{s}$ is still $(q \cdot 2^{-m^\epsilon})$-hard given $(\mathbf{g}, y, h(\mathbf{g}, \mathbf{s}))$, and a PPT adversary $\mathcal{A}$ with advantage $\delta = \delta(n) = \mathsf{Adv}_{\mathcal{A},h}(n)$.

We consider a sequence of experiments, and let $\mathsf{Adv}^{(i)}_{\mathcal{A},h}(n)$ denote the advantage of the adversary in experiment i.

Experiment 0: This is the experiment in Definition 1. The adversary $\mathcal{A}$ gets as input $PK = (\mathbf{g}, y)$ and the auxiliary input $h(\mathbf{g}, \mathbf{s})$. $\mathcal{A}$ chooses two messages M_0 and M_1, and receives $\mathsf{Enc}(PK, M_b)$ where $b \in \{0, 1\}$ is uniformly random. $\mathcal{A}$ succeeds in the experiment if he succeeds in guessing b. By assumption, $\mathsf{Adv}^{(0)}_{\mathcal{A},h}(n) = \mathsf{Adv}_{\mathcal{A},h}(n) = \delta$.

Experiment 1: In this experiment, the challenge ciphertext M_b is generated by "encrypting with the secret key," rather than with the usual $\mathsf{Enc}(PK, M_b)$ algorithm. In particular, define the algorithm $\mathsf{Enc}'(\mathbf{g}, \mathbf{s}, M_b)$ as follows.

1. Choose $r \leftarrow \mathbb{Z}_q$ uniformly at random, and compute the first m components of the ciphertext $(f_1, \ldots, f_m) = (g_1^r, \ldots, g_m^r)$.
2. Compute the last component of the ciphertext as $c = \prod_{i=1}^{m} f_i^{s_i} \cdot M_b$.

Claim 5.2. *The distribution produced by Enc' is identical to the distribution of a real ciphertext; in particular, $\mathsf{Adv}^{(0)}_{\mathcal{A},h}(n) = \mathsf{Adv}^{(1)}_{\mathcal{A},h}(n)$.*

Proof. Fix $\mathbf{g}$ and $\mathbf{s}$ (and hence y). Then for uniformly random $r \in \mathbb{Z}_q$, both Enc and Enc' compute the same $f_1, \ldots, f_m$, and their final ciphertext components also coincide:

$$c = \prod_{i=1}^{m} f_i^{s_i} \cdot M = \prod_{i=1}^{m} g_i^{r s_i} \cdot M = \Big(\prod_{i=1}^{m} g_i^{s_i} \Big)^r \cdot M = y^r \cdot M. \qquad \square$$

Experiment 2: In this experiment, the vector $(f_1, \ldots, f_m)$ in the ciphertext is taken from the *uniform* distribution over G^m, i.e., each $f_i = g^{r_i}$ for uniformly random and independent $r_i \in \mathbb{Z}_q$ (where g is some fixed generator of G), and $c = \prod_i f_i^{s_i} \cdot M_b$ as before. Under DDH assumption and by Lemma 2.1, it immediately follows that the advantage of the adversary changes by at most a negligible amount.

Claim 5.3. *If the DDH problem is hard for $\mathcal{G}$, then for every PPT algorithm $\mathcal{A}$ and for every function $h \in \mathcal{H}$, $\left|\mathsf{Adv}^{(1)}_{\mathcal{A},h}(n) - \mathsf{Adv}^{(2)}_{A,h}(n)\right| \leq \mathrm{negl}(n)$.*

Experiment 3: In this experiment, the final component of the ciphertext is replaced by a uniformly random element $u \leftarrow G$. Namely, the ciphertext is generated as $(g^{r_1}, \ldots, g^{r_m}, g^u)$, where $r_i \in \mathbb{Z}_q$ and $u \in \mathbb{Z}_q$ are all uniformly random and independent.

Claim 5.4. *For every PPT algorithm $\mathcal{A}$ and for every $h \in \mathcal{H}$, $\left|\mathsf{Adv}^{(2)}_{\mathcal{A},h}(n) - \mathsf{Adv}^{(3)}_{\mathcal{A},h}(n)\right| \leq \mathrm{negl}(n)$.*

Proof. We reduce the task of inverting h (with suitable probability) to the task of gaining some non-negligible $\delta = \delta(n)$ distinguishing advantage between experiments 2 and 3.

We wish to construct an efficient algorithm that, given $PK = (\mathbf{g}, y)$ and $h(\mathbf{g}, \mathbf{s})$, outputs $\mathbf{s} \in H^m = \{0,1\}^m$ with probability at least

$$q \cdot \frac{\delta^3}{512n \cdot q^3} > q \cdot \frac{1}{512n \cdot 2^{3m^\epsilon/4} \cdot \mathrm{poly}(n)} > q \cdot 2^{-m^\epsilon},$$

for large enough n. By Theorem 4.1, it suffices to reduce δ-distinguishing

$$(PK, h(\mathbf{g}, \mathbf{s}), \mathbf{r} \in \mathbb{Z}_q^m, \langle \mathbf{r}, \mathbf{s} \rangle) \text{ from } (PK, h(\mathbf{g}, \mathbf{s}), \mathbf{r}, u \in \mathbb{Z}_q)$$

to δ-distinguishing between experiments 2 and 3.

The reduction $\mathcal{B}$ that accomplishes this, given $(PK = (\mathbf{g}, y), h(\mathbf{g}, \mathbf{s}), \mathbf{r}, z \in \mathbb{Z}_q)$, simulates the view of the adversary $\mathcal{A}$ as follows. Give PK to $\mathcal{A}$ and get back two messages M_0, M_1; choose a bit $b \in \{0,1\}$ at random and give $\mathcal{A}$ the ciphertext $(g^{r_1}, \ldots, g^{r_m}, g^z \cdot M_b)$. Let b' be the output of $\mathcal{A}$; if $b = b'$ then $\mathcal{B}$ outputs 1, and otherwise $\mathcal{B}$ outputs 0.

By construction, it may be checked that when $\mathcal{B}$'s input component $z = \langle \mathbf{r}, \mathbf{s} \rangle \in \mathbb{Z}_q$, $\mathcal{B}$ simulates experiment 2 to $\mathcal{A}$ perfectly. Likewise, when z is uniformly random and independent of the other components, $\mathcal{B}$ simulates experiment 3 perfectly. It follows that $\mathcal{B}$'s advantage equals $\mathcal{A}$'s. □

Now the ciphertext in experiment 3 is independent of the bit b that selects which message is encrypted. Thus, the adversary has no advantage in this experiment, i.e, $\mathsf{Adv}^{(3)}_{\mathcal{A},h}(n) = 0$. Putting together the claims, we get that $\mathsf{Adv}_{\mathcal{A},h}(n) \leq \mathrm{negl}(n)$. This concludes the proof of Theorem 5.1.

Scheme with better weak security. While the natural version of our encryption scheme above would require us to work with an ϵ-distinguisher for the inner product $\langle \mathbf{r}, \mathbf{s} \rangle$, we can make less efficient variant of our scheme[6] which would allow us to work with an ϵ-predictor for the value $g^{\langle \mathbf{r}, \mathbf{s} \rangle}$, as opposed to the inner product $\langle \mathbf{r}, \mathbf{s} \rangle$ itself. Luckily, the GL inversion algorithm of [15] (which needs a predictor) can be easily modified to work in this case, but with the the caveat that it will output the tuple $(g^{s_1}, \ldots, g^{s_n})$. However, when each $s_i \in H$, we can compute discrete logs in time $O(|H|)$ (which is constant for our case $|H| = 2$). This means we can have the following variant of [15]. Given an ϵ-predictor for the value $g^{\langle \mathbf{r}, \mathbf{s} \rangle}$, we can compute $\mathbf{s}$ in time $\mathsf{poly}(n, |H|, 1/\epsilon)$ and success probability $\mathsf{poly}(\epsilon, 1/n)$, both of which do not depend on q.

Applying this to our scheme, where $|H| = 2$, we get a DDH-based encryption scheme which is $\mathrm{negl}(n)$-`wAI-CPA` secure. The details will appear in the full version.

[6] Which will apply so called *reconstructive extractor* to the value $g^{\langle \mathbf{r}, \mathbf{s} \rangle}$ and then XOR this with the message. This variant is less efficient because the output of the extractor will only be at most logarithmic in the security parameter.

References

[1] Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In *TCC*, pages 474–495, 2009. 1, 2, 3, 7, 8

[2] Joel Alwen, Yevgeniy Dodis, and Daniel Wichs. Leakage-resilient public-key cryptography in the bounded-retrieval model. In *CRYPTO*, pages ??–??, 2009. 1

[3] Dan Boneh, Shai Halevi, Michael Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision diffie-hellman. In *CRYPTO*, pages 108–125, 2008. 3, 10

[4] Xavier Boyen. Reusable cryptographic fuzzy extractors. In *ACM Conference on Computer and Communications Security*, pages 82–91, 2004. 2

[5] 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. 1

[6] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO*, pages 13–25, 1998. 4

[7] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *EUROCRYPT*, pages 45–64, 2002. 1

[8] Whitfield Diffie and Martin E. Hellman. Multiuser cryptographic techniques. In *AFIPS National Computer Conference*, pages 109–112, 1976. 3

[9] Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett. On cryptography with auxiliary input. In *STOC*, pages 621–630, 2009. 1, 2, 8

[10] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *EUROCRYPT*, pages 523–540, 2004. 2

[11] Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *FOCS*, pages 293–302, 2008. 1

[12] Sebastian Faust, Eike Kiltz, Krzysztof Pietrzak, and Guy Rothblum. Leakage-resilient signatures, 2009. Available at `http://eprint.iacr.org/2009/282`. 1

[13] William Feller. *An Introduction to Probability Theory and Its Applications, Volume 1.* Wiley, 1968. 17

[14] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206, 2008. 3, 18, 21

[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. 5, 12, 17

[16] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. One-time programs. In *CRYPTO*, pages 39–56, 2008. 1

[17] Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In *CRYPTO*, pages 463–481, 2003. 1

[18] Jonathan Katz and Vinod Vaikuntanathan. Signature schemes with bounded leakage. In *ASIACRYPT*, pages ??–??, 2009. 1, 4

[19] Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Multi-bit cryptosystems based on lattice problems. In *Public Key Cryptography*, pages 315–329, 2007. 19

[20] Silvio Micali and Leonid Reyzin. Physically observable cryptography (extended abstract). In *TCC*, pages 278–296, 2004. 1

[21] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. *J. ACM*, 51(2):231–262, 2004. 6

[22] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In *CRYPTO*, pages ??–??, 2009. 1, 2, 3, 4, 7, 8

[23] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *STOC*, pages 333–342, 2009. 18

[24] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In *CRYPTO*, pages 554–571, 2008. 19

[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. 1

[26] Krzysztof Pietrzak. A leakage-resilient mode of operation. In *EUROCRYPT*, pages 462–482, 2009. 1

[27] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93, 2005. 1, 3, 17, 18

[28] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978. 3

A Goldreich-Levin Theorem for Large Fields

We will actually prove a tighter version of the bound stated in Equation 1, where the success probability of $\mathcal{A}$ is $\epsilon/4q^c$, where $c \geq 2$ is the smallest integer such that $q^c > 128|H|n/\epsilon^2$. The general bound in Equation 1 follows since $4q^c = 4\max(q^2, q \cdot (128|H|n/\epsilon^2)) = 4q \cdot \max(q, 128|H|n/\epsilon^2) \leq 512q^2n/\epsilon^2$. Thus,

- If $q > 128|H|n/\epsilon^2$, then $c = 2$ and the above probability is at least $\epsilon/4q^2$.
- If $q \leq 128|H|n/\epsilon^2$, then $c = \lceil \log_q(128n|H|/\epsilon^2) \rceil$ and the above probability is at least $\epsilon^3/512nq|H| = \mathsf{poly}(\epsilon, 1/n, 1/|H|)$.

Proof. As mentioned above, we will design an inverter $\mathcal{A}$ with a stronger success probability $\epsilon/4q^c$, where $c \geq 2$ is the smallest integer such that $q^c > 128|H|n/\epsilon^2$. Without loss of generality, we will drop the absolute value from the condition on $\mathcal{D}$, by flipping the decision of $\mathcal{D}$, if needed. Also, for a fixed value $\mathbf{s} \in H^\ell$ and fixed randomness of f (in case f is randomized), let $y = f(\mathbf{s})$ and let

$$\begin{aligned} \alpha_{\mathbf{s},y} &= \Pr[\mathbf{r} \leftarrow GF(q)^n : \mathcal{D}(y, \mathbf{r}, \langle \mathbf{r}, \mathbf{s} \rangle) = 1] \\ \beta_{\mathbf{s},y} &= \Pr[\mathbf{r} \leftarrow GF(q)^n, u \leftarrow GF(q) : \mathcal{D}(y, \mathbf{r}, u) = 1] \end{aligned}$$

Thus, we know that $\mathbb{E}_{\mathbf{s}}[\alpha_{\mathbf{s},y} - \beta_{\mathbf{s},y}] \geq \epsilon$ (note, this expectation also includes possible randomness of f, but we ignore it to keep the notation uncluttered). Let us call the pair $(\mathbf{s}, y)$ *good* if $\alpha_{\mathbf{s},y} - \beta_{\mathbf{s},y} \geq \epsilon/2$. Since $\alpha_{\mathbf{s},y} - \beta_{\mathbf{s},y} \leq 1$, a simple averaging argument implies that

$$\Pr[\mathbf{s} \leftarrow H^n, y \leftarrow f(\mathbf{s}) : (\mathbf{s}, y) \text{ is good}] \geq \epsilon/2 \tag{2}$$

Below, we will design an algorithm $\mathcal{A}(y)$ which will succeed to recover $\mathbf{s}$ from y with probability $1/2q^c$, whenever the pair $(\mathbf{s}, y)$ is good. Coupled with Equation 2, this will establish that $\mathcal{A}$'s overall probability of success (for random $\mathbf{s}$ and y) is at least $\epsilon/4q^c$, as required. Thus, in the discussion below, we will assume that $(\mathbf{s}, y)$ is fixed and good.

Before describing $\mathcal{A}(y)$, we will also assume that $\mathcal{A}(y)$ can compute, with overwheliming probability, a number $\gamma_{\mathbf{s},y}$ such that $(\alpha_{\mathbf{s},y} - \epsilon/8 \geq \gamma_{\mathbf{s},y} \geq \beta_{\mathbf{s},y} + \epsilon/8)$. Indeed, by sampling $O(n/\epsilon^2)$ random and independent vectors $\mathbf{r}$ and u, $\mathcal{A}(y)$ can compute an estimate e for $\beta_{\mathbf{s},y}$, such that $\Pr(|e - \beta_{\mathbf{s},y}| > \epsilon/8) \leq 2^{-n}$ (by the Chernoff's bound), after which one can set $\gamma_{\mathbf{s},y} = e + \epsilon/4$. So we will assume that $\mathcal{A}(y)$ can compute such an estimate $\gamma_{\mathbf{s},y}$.

Let $m \stackrel{\mathsf{def}}{=} 128|H|n/\epsilon^2$. By assumption, $c \geq 2$ is such that $q^c > m$. Let us fix an arbitrary subset $S \subseteq GF(q)^c \backslash \{0^c\}$ of cardinality m. The algorithm $\mathcal{A}(y)$ works as follows.

1. Compute the value $\gamma_{\mathbf{s},y}$ such that $(\alpha_{\mathbf{s},y} - \epsilon/8 \geq \gamma_{\mathbf{s},y} \geq \beta_{\mathbf{s},y} + \epsilon/8)$, as described above.
2. Choose c random vectors $\mathbf{z}_1, \ldots, \mathbf{z}_c \leftarrow GF(q)^n$, and c random elements $g_0, \ldots, g_c \leftarrow GF(q)$.
   [Remark: Informally, the g_i are the "guesses" of the algorithm A for the values of $\langle \mathbf{z}_i, \mathbf{s} \rangle$.]
3. For every tuple $\bar{\rho} = (\rho_1, \ldots, \rho_c) \in S$, compute
$$\mathbf{r}_{\bar{\rho}} := \sum_{j=1}^{c} \rho_j \mathbf{z}_j \quad \text{and} \quad h_{\bar{\rho}} := \sum_{j=1}^{c} \rho_j g_j \tag{3}$$
   [Remark: If the guesses g_i are all correct, then for every $\bar{\rho}$, we have $h_{\bar{\rho}} = \langle \mathbf{r}_{\bar{\rho}}, \mathbf{s} \rangle$.
   Also notice that the vectors $\mathbf{r}_{\bar{\rho}}$ are pairwise independent since $c \geq 2$ and $\bar{\rho} \neq 0^c$.]
4. For each $i \in [n]$, do the following:
   - For each $a \in H$, guess that $s_i = a$, and run the following procedure to check if the guess is correct:
     - For each $\bar{\rho} \in S$, choose a random $\tau_{\bar{\rho}}^{(i,a)} \in GF(q)$ and run
$$\mathcal{D}(y, \mathbf{r}_{\bar{\rho}} + \tau_{\bar{\rho}}^{(i,a)} \cdot \mathbf{e}_i, h_{\bar{\rho}} + \tau_{\bar{\rho}}^{(i,a)} \cdot a)$$
       where $\mathbf{e}_i$ is the i^{th} unit vector. Let $p^{(i,a)}$ be the fraction of $\mathcal{D}$'s answers which are 1.
     - If the $p^{(i,a)} \geq \gamma_{\mathbf{s},y}$, set $s_i := a$ and move to the next $i + 1$.
       Otherwise, move to the next guess $a \in H$.
5. Output $\mathbf{s} = s_1 s_2 \ldots s_n$ (or fail if some s_i was not found).

The procedure invokes the distinguisher $\mathcal{D}$ at most $O(nm|H|)$ times (not counting the estimation step for $\gamma_{\mathbf{s},y}$ which is smaller), and thus the running time is $O(t \cdot nm|H|) = t \cdot \mathsf{poly}(n, |H|, 1/\epsilon)$, where t is the running time of $\mathcal{D}$. Let us now analyze the probability that the procedure succeeds.

First, define the event E to be the event that for all $\bar{\rho} \in S$, we have $h_{\bar{\rho}} = \langle \mathbf{r}_{\bar{\rho}}, \mathbf{s} \rangle$.

$$\Pr[E] = \Pr[\forall \bar{\rho} \in S, h_{\bar{\rho}} = \langle \mathbf{r}_{\bar{\rho}}, \mathbf{s} \rangle] \geq \Pr[\forall i \in [1, \ldots, c], g_i = \langle \mathbf{z}_i, \mathbf{s} \rangle] = \frac{1}{q^c}$$

where the last equality follows from the fact that A's random guess of g_i are all correct with probability $1/q^c$. For the rest of the proof, we condition on the event E (and, of course, on he goodness of $(\mathbf{s}, y)$), and show that $\mathcal{A}$'s success is at least $1/2$ in this case, completing the proof.

We next prove two claims. First, in Claim A.1 we show that if A's guess a for s_i is correct, then each individual input to $\mathcal{D}$ is distributed like $(y, \mathbf{r}, \langle \mathbf{r}, \mathbf{s} \rangle)$, for a random $\mathbf{r}$. Thus, the probability that $\mathcal{D}$ answers 1 on these inputs is exactly $\alpha_{\mathbf{s},y}$. Moreover, the inputs to $\mathcal{D}$ are *pairwise independent* (which is obvious from their definition in Equation 3, since we also excluded $\bar{\rho} = 0^c$). Thus, by Chebyshev's inequality, the probability that the average $p^{i,a}$ of m pairwise independent estimations of $\alpha_{\mathbf{s},y}$ is smaller than $\gamma_{\mathbf{s},y}$, which is more than $\epsilon/8$ smaller than the true average $\alpha_{\mathbf{s},y}$, is at most $1/(m(\epsilon/8)^2) = 1/2|H|n$, where we recall that $m = 128|H|n/\epsilon^2$.

Secondly, in Claim A.2 we show that for every incorrect guess a for s_i, each individual input to $\mathcal{D}$ is distributed like $(y, \mathbf{r}, u)$ for a random $\mathbf{r}$ and u. Thus, the probability that $\mathcal{D}$ answers 1 on these inputs is exactly $\beta_{\mathbf{s},y}$. And, as before, different values $\mathbf{r}$ and u are pairwise independent.[7] By an argument similar to the above, in this case the probability that the average $p^{i,a}$ of m pairwise independent estimations of $\beta_{\mathbf{s},y}$ is larger than $\gamma_{\mathbf{s},y}$, which is more than $\epsilon/8$ larger than the true average $\beta_{\mathbf{s},y}$, is at most $1/(m(\epsilon/8)^2) = 1/2|H|n$, where we recall that $m = 128|H|n/\epsilon^2$.

This suffices to prove the our result, since, by the union bound over all $i \in [1, \ldots, n]$ and $a \in |H|$, the chance that $\mathcal{A}$ will incorrectly test any pair (i, a) (either as false positive or false negative) is at most $|H|n \cdot 1/(2|H|n) = 1/2$. Thus, it suffices to prove the two claims.

Claim A.1. *If A's guess is correct, i.e, $s_i = a$, the inputs to $\mathcal{D}$ are distributed like $(y, \mathbf{r}, \langle \mathbf{r}, \mathbf{s} \rangle)$.*

Proof: Each input to $\mathcal{D}$ is of the form $(y, \mathbf{r}_{\bar{\rho}} + \tau_{\bar{\rho}}^{(i,a)} \cdot \mathbf{e}_i, h_{\bar{\rho}} + \tau_{\bar{\rho}}^{(i,a)} \cdot a)$. Since we already conditioned on E, we know that $h_{\bar{\rho}} = \langle \mathbf{r}_{\bar{\rho}}, \mathbf{s} \rangle$. Also, we assumed that $s_i = a$. Thus,

$$h_{\bar{\rho}} + \tau_{\bar{\rho}}^{(i,a)} \cdot a = \langle \mathbf{r}_{\bar{\rho}}, \mathbf{s} \rangle + \tau_{\bar{\rho}}^{(i,a)} \cdot s_i = \langle \mathbf{r}_{\bar{\rho}}, \mathbf{s} \rangle + \langle \tau_{\bar{\rho}}^{(i,a)} \mathbf{e}_i, \mathbf{s} \rangle = \langle \mathbf{r}_{\bar{\rho}} + \tau_{\bar{\rho}}^{(i,a)} \cdot \mathbf{e}_i, \mathbf{s} \rangle$$

Since $\mathbf{r}_{\bar{\rho}} + \tau_{\bar{\rho}}^{(i,a)} \cdot \mathbf{e}_i$ is uniformly random by itself (see Equation 3 and remember $\bar{\rho} \neq 0^c$), the input of $\mathcal{D}$ is indeed of the form $(y, \mathbf{r}, \langle \mathbf{r}, \mathbf{s} \rangle)$, where $\mathbf{r} := \mathbf{r}_{\bar{\rho}} + \tau_{\bar{\rho}}^{(i,a)} \cdot \mathbf{e}_i$ is uniformly random. □

Claim A.2. *If A's guess is incorrect, i.e, $s_i \neq a$, the inputs to $\mathcal{D}$ are distributed like $(y, \mathbf{r}, u)$ for a uniformly random $u \in GF(q)$.*

Proof: The proof proceeds similar to Claim A.1. As before, each input to $\mathcal{D}$ is of the form $(y, \mathbf{r}_{\bar{\rho}} + \tau_{\bar{\rho}}^{(i,a)} \cdot \mathbf{e}_i, h_{\bar{\rho}} + \tau_{\bar{\rho}}^{(i,a)} \cdot a)$. Now, however, $s_i \neq a$, so suppose $a - s_i = t_i \neq 0$. Then

$$h_{\bar{\rho}} + \rho_{\bar{\rho}}^{(i,a)} \cdot a = \langle \mathbf{r}_{\bar{\rho}}, \mathbf{s} \rangle + \rho_{\bar{\rho}}^{(i,a)} \cdot (s_i + t_i) = \langle \mathbf{r}_{\bar{\rho}} + \rho_{\bar{\rho}}^{(i,a)} \cdot \mathbf{e}_i, \mathbf{s} \rangle + \rho_{\bar{\rho}}^{(i,a)} \cdot t_i$$

Let $\mathbf{r} := \mathbf{r}_{\bar{\rho}} + \tau_{\bar{\rho}}^{(i,a)} \cdot \mathbf{e}_i$ and $u := h_{\bar{\rho}} + \rho_{\bar{\rho}}^{(i,a)} \cdot a$, so that the input to $\mathcal{D}$ is $(y, \mathbf{r}, u)$. By the equation above, we have $u = \langle \mathbf{r}, \mathbf{s} \rangle + \rho_{\bar{\rho}}^{(i,a)} \cdot t_i$. Also, since $\mathbf{r}_{\bar{\rho}}$ is uniformly random, it perfectly hides $\tau_{\bar{\rho}}^{(i,a)}$ in the definition of $\mathbf{r}$. Thus, $\mathbf{r}$ is independent from $\tau_{\bar{\rho}}^{(i,a)}$. Finally, since we assumed that $t_i \neq 0$

[7] The argument is the same for $\mathbf{r}$, and for the values u, Claim A.2 shows that they are in fact completely independent.

and the value $\tau_{\tilde{\rho}}^{(i,a)}$ was random in $GF(q)$, this means that $u = \langle \mathbf{r}, \mathbf{s} \rangle + \rho_{\tilde{\rho}}^{(i,a)} \cdot t_i$ is random and independent of $\mathbf{r}$, as claimed. □

This concludes the proof of Theorem 4.1. □

Remark A.3. We briefly compare our new variant of the GL Lemma for general q with the similar-looking extension of Goldreich, Rubinefeld and Sudan [15]. The etensions are incomparable, in the following sense. In [15], the authors assume an ϵ-*predictor* for the inner product $\langle \mathbf{r}, \mathbf{s} \rangle$, which is a stronger assumption than the existence of an ϵ-*distinguisher*, especially as q grows. On the other, in [15] both the running time and the inversion probability of the inverter they construct depends only on n/ϵ and is *independent of* q (and, hence, $|H|$, if one considers restricting the domain as we do). Unfortunately, if one generically converts a distinguisher into a predictor, this conversion makes the prediction advantage equal to ϵ/q, which means that applying [15] would make both the inversion probability and the *running time of the inverter* depend on q. In contrast, we directly work with the distinguisher, and manage to only make the inversion probability dependent on q, while the *running time dependent only on* $|H|$.

B The Learning with Errors Assumption

Learning with errors (LWE). The LWE problem was introduced by Regev [27] as a generalization of the "learning noisy parities" problem. For positive integers n and $q \geq 2$, a vector $\mathbf{s} \in \mathbb{Z}_q^n$, and a probability distribution χ on $\mathbb{Z}_q$, let $A_{\mathbf{s},\chi}$ be the distribution obtained by choosing a vector $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random and a noise term $x \leftarrow \chi$, and outputting $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + x) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.

Definition 3. For an integer $q = q(n)$ and an error distribution $\chi = \chi(n)$ over $\mathbb{Z}_q$, the (worst-case) learning with errors problem $\mathsf{LWE}_{n,m,q,\chi}$ in n dimensions is defined as follows. Given m independent samples from $A_{\mathbf{s},\chi}$ (where $\mathbf{s} \in \mathbb{Z}_q^n$ is arbitrary), output $\mathbf{s}$ with noticeable probability.

The (average-case) decisional variant of the LWE problem, denoted $\mathsf{DLWE}_{n,m,q,\chi}$, is to distinguish (with non-negligible advantage) m samples chosen according to $A_{\mathbf{s},\chi}$ for *uniformly random* $\mathbf{s} \in \mathbb{Z}_q^n$, from m samples chosen according to the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

For cryptographic applications we are primarily interested in the average-case decision problem DLWE. Fortunately, Regev [27] showed that for a prime modulus q, the (worst-case) LWE and (average-case) DLWE problems are equivalent, up to a $q \cdot \mathrm{poly}(n)$ factor in m. We say that $\mathsf{LWE}_{n,m,q,\chi}$ (respectively, $\mathsf{DLWE}_{n,m,q,\chi}$) is hard if no PPT algorithm can solve it for infinitely many n.

At times, we find it convenient to describe the LWE problem $\mathsf{LWE}_{n,m,q,\chi}$ using a compact matrix notation: given $(\mathbf{A}, \mathbf{A}^T\mathbf{s} + \mathbf{x})$ where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ is uniformly random, $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ is the LWE secret, and $\mathbf{x} \leftarrow \chi^m$, find $\mathbf{s}$. We also use similar matrix notation for the decision version DLWE.

Gaussian error distributions. We are primarily interested in the LWE and DLWE problems where the error distribution χ over $\mathbb{Z}_q$ is derived from a Gaussian. For any $r > 0$, the density function of a one-dimensional Gaussian probability distribution over $\mathbb{R}$ is given by $D_r(x) = 1/r \cdot \mathsf{exp}(-\pi(x/r)^2)$. For $\beta > 0$, define $\overline{\Psi}_\beta$ to be the distribution on $\mathbb{Z}_q$ obtained by drawing $y \leftarrow D_\beta$ and outputting $\lfloor q \cdot y \rceil \pmod q$. We write $\mathsf{LWE}_{n,m,q,\beta}$ as an abbreviation for $\mathsf{LWE}_{n,m,q,\overline{\Psi}_\beta}$.

Here we state some basic facts about Gaussians (tailored to the error distribution $\overline{\Psi}_\beta$); see, e.g. [13]. (In what follows, $\|\mathbf{x}\|$ denotes the ℓ_2 norm of a vector $\mathbf{x}$.)

Lemma B.1. *Let $\beta > 0$ and $q \in \mathbb{Z}$.*

1. *Let $y \leftarrow \overline{\Psi}_\beta$. With overwhelming probability, $|y| \leq \beta q \cdot \sqrt{n}$.*

2. *Let the vector* $\mathbf{x} \in \mathbb{Z}^n$ *be arbitrary, and the vector* $\mathbf{y} \leftarrow \overline{\Psi}_\beta^n$. *With overwhelming probability over the choice of* $\mathbf{y}$,
$$|\langle \mathbf{x}, \mathbf{y} \rangle| \leq ||\mathbf{x}|| \cdot \beta q \cdot \omega(\sqrt{\log n})$$

3. *Let* $y \in \mathbb{R}$ *be arbitrary. The statistical distance between the distributions* $\overline{\Psi}_\beta$ *and* $\overline{\Psi}_\beta + y$ *is at most* $|y|/(\beta q)$.

Evidence for the hardness of $\mathsf{LWE}_{n,m,q,\beta}$ follows from results of Regev [27], who gave a *quantum* reduction from approximating certain problems on n-dimensional lattices in the worst case to within $\widetilde{O}(n/\beta)$ factors to solving $\mathsf{LWE}_{n,m,q,\beta}$ for any desired $m = \mathrm{poly}(n)$, when $\beta \cdot q \geq 2\sqrt{n}$. Recently, Peikert [23] also gave a related *classical* reduction for similar parameters.

C Constructions Based on LWE

C.1 Strong Auxiliary Input Security for the GPV Cryptosystem

First, we present (a modification of) the GPV encryption scheme [14]. We then show that the system is secure against sub-exponentially hard auxiliary input functions, assuming the hardness of the learning with error (LWE) problem.

C.1.1 The GPV Cryptosystem

Let n denote the security parameter, and let $0 < \epsilon \leq 1$. Let $f(n) = 2^{\omega(\log n)}$ be some superpolynomial function. Let the prime $q \in (f(n), 2 \cdot f(n)]$, the integer $m = ((n+3)\log q)^{1/\epsilon}$ and the error-distributions $\overline{\Psi}_\beta$ and $\overline{\Psi}_\gamma$ where $\beta = 2\sqrt{n}/q$ and $\gamma = 1/(8 \cdot \omega(\sqrt{\log n}))$ be parameters of the system.

$\mathsf{Gen}(1^n)$: Choose a uniformly random matrix $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ and a random vector $\mathbf{e} \leftarrow \{0,1\}^m$. Compute $\mathbf{u} = \mathbf{A}\mathbf{e}$. The public-key $PK := (\mathbf{A}, \mathbf{u})$ and the secret-key $SK := \mathbf{e}$. We notice that the matrix $\mathbf{A}$ could be viewed as a public parameter, making $\mathbf{u}$ the only "user-specific" part of PK for the purposes of Lemma 3.1.

$\mathsf{Enc}(PK, b)$, where b is a bit, works as follows. Choose a random vector $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, a vector $\mathbf{x} \leftarrow \overline{\Psi}_\beta^m$ and $x' \leftarrow \overline{\Psi}_\gamma$. Output the ciphertext
$$\left(\mathbf{A}^T\mathbf{s} + \mathbf{x}, \mathbf{u}^T\mathbf{s} + x' + b\left\lfloor \frac{q}{2} \right\rfloor \right)$$

$\mathsf{Dec}(SK, c)$: Parse the ciphertext as $(\mathbf{y}, c) \in \mathbb{Z}_q^m \times \mathbb{Z}_q$ and compute $b' = (c - \mathbf{e}^T\mathbf{y})/q$. Output 1 if $\left|b' - \frac{1}{2}\right| \leq \frac{1}{4}$, and 0 otherwise.

Remark C.1. Two remarks about the above cryptosystem are in order.

1. The main difference between the cryptosystem in [14] and the variant described here is two-fold. (1) we choose the error-parameter β to be superpolynomially small in n (and the modulus q to be superpolynomially large), whereas in [14], both are polynomially related to n. (2) the secret-key distribution in our case is the uniform distribution over $\{0,1\}^m$, whereas in [14], it is the discrete Gaussian distribution $D_{\mathbb{Z}^m, r}$ for some $r > 0$. The first modification is essential to our proof of auxiliary-input security. The second modification can be done away with, and doing so results in an identity-based encryption scheme (as in [14]) secure against auxiliary inputs. We defer the details to the full version of this paper.

2. Although the encryption scheme described here is a bit-encryption scheme, it can be modified to encrypt $O(\log q)$ bits with one invocation of the encryption algorithm, using the ideas of [19, 24]. However, we note that another optimization proposed in [24] that achieves *constant* ciphertext expansion does not seem to lend itself to security against auxiliary inputs. Roughly speaking, the reason is that the optimization enlarges the secret-key by repeating the secret-key of the basic encryption scheme polynomially many times; this seems to adversely affect auxiliary input security.

We first show the correctness of the encryption scheme. The decryption algorithm computes

$$b' = (c - \mathbf{e}^T\mathbf{y})/q = \left(\mathbf{u}^T\mathbf{s} + x' + b\left\lfloor\frac{q}{2}\right\rfloor - \mathbf{e}^T(\mathbf{A}^T\mathbf{s} + \mathbf{x})\right)/q = b \cdot \frac{1}{2} + (x' - \mathbf{e}^T\mathbf{x} - (1/2)/q)$$

Now, by Lemma B.1, $|\mathbf{e}^T\mathbf{x}| \leq \sqrt{m} \cdot \beta q \cdot \omega(\sqrt{\log n})$ and $|x'| \leq \gamma q \cdot \omega(\sqrt{\log n})$ with all but negligible probability. The difference between b' and $b \cdot 1/2$ is at most $|x' - \mathbf{e}^T\mathbf{x} + 1/2q|$ which, by plugging in the values of β and γ and using the triangle inequality, is at most $q/4$.

C.1.2 Security for Subexponentially Hard-to-Invert Auxiliary Inputs

Theorem C.2. *Let the superpolynomial function $f(n)$ and the parameters m, q, β and γ be as in the encryption scheme described above. Assuming that the $\mathsf{DLWE}_{n,m,q,\beta}$ problem is hard, the encryption scheme above is (2^{-m^ϵ})-$\mathtt{AI\text{-}CPA}$ secure (when $\mathbf{A}$ is viewed as a public parameter).*

Remark. We can actually handle a richer class of auxiliary inputs. We can prove security even for auxiliary functions $h(\mathbf{A}, \mathbf{e})$ that (given $\mathbf{A}$) are hard to invert with probability 2^{-k}, where k can be as small as polylog(m). However, then the assumption we rely on is that LWE is hard for adversaries that run in subexponential time. For the sake of simplicity, we only state the theorem for $k = m^\epsilon$ in which case we can rely on the standard LWE hardness assumption.

Proof of Theorem C.2. By Lemma 3.1 (Part 2.) and because the length of "user-specific" public-key $\mathbf{u}$ is $n \log q$ bits, to show Theorem C.2 it suffices to show that our encryption scheme is $(q^n 2^{-m^\epsilon})$-$\mathtt{wAI\text{-}CPA}$ secure. Fix any auxiliary-input function h, so that $\mathbf{e}$ is still $(q^n \cdot 2^{-m^\epsilon})$-hard given $(\mathbf{A}, \mathbf{u}, h(\mathbf{A}, \mathbf{e}))$, and a PPT adversary $\mathcal{A}$ with advantage $\delta = \delta(n) = \mathsf{Adv}_{\mathcal{A},h}(n)$.

We consider a sequence of experiments, and let $\mathsf{Adv}^{(i)}_{\mathcal{A},h}(n)$ denote the advantage of the adversary in experiment i.

Experiment 0: This is the experiment in Definition 1. The adversary $\mathcal{A}$ gets as input $PK = (\mathbf{A}, \mathbf{u})$ and the auxiliary input $h(\mathbf{A}, \mathbf{e})$. $\mathcal{A}$ receives $\mathsf{Enc}(PK, b)$ where $b \in \{0, 1\}$ is uniformly random. $\mathcal{A}$ succeeds in the experiment if he succeeds in guessing b. By assumption, $\mathsf{Adv}^{(0)}_{\mathcal{A},h}(n) = \mathsf{Adv}_{\mathcal{A},h}(n) = \delta$.

Experiment 1: In this experiment, the challenge ciphertext is generated by "encrypting with the secret key," rather than with the usual $\mathsf{Enc}(PK, b)$ algorithm. In particular, define the algorithm $\mathsf{Enc}'(\mathbf{A}, \mathbf{e}, b)$ as follows.

1. Choose $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ at random and $\mathbf{x} \leftarrow \overline{\Psi}_\beta^m$, and compute the first component of the ciphertext $\mathbf{y} = \mathbf{A}^T\mathbf{s} + \mathbf{x}$.

2. Choose $x' \leftarrow \overline{\Psi}_\gamma$ and compute the second component of the ciphertext as

$$c = \mathbf{e}^T\mathbf{y} + x' + b\lfloor q/2 \rfloor$$

Claim C.3. *The distribution produced by* Enc' *is statistically close to the distribution of a real ciphertext; in particular, there is a negligible function* negl *such that*

$$\left|\mathsf{Adv}^{(0)}_{\mathcal{A},h}(n) - \mathsf{Adv}^{(1)}_{\mathcal{A},h}(n)\right| \leq \mathsf{negl}(n).$$

Proof. Fix $\mathbf{A}$ and $\mathbf{e}$ (and hence $\mathbf{u}$). Then for uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$ and $\mathbf{x} \leftarrow \overline{\Psi}_\beta^m$, both Enc and Enc' compute the same $\mathbf{y}$. Given $\mathbf{y}$, the second component of the ciphertext produced by Enc' is

$$c = \mathbf{e}^T\mathbf{y} + x' + b\lfloor q/2 \rfloor = \mathbf{e}^T A\mathbf{s} + (\mathbf{e}^T\mathbf{x} + x') + b\lfloor q/2 \rfloor = \mathbf{u}^T\mathbf{s} + (\mathbf{e}^T\mathbf{x} + x') + b\lfloor q/2 \rfloor$$

It suffices to show that the distribution of $\mathbf{e}^T\mathbf{x} + x'$ is statistically indistinguishable from $\overline{\Psi}_\gamma$. This follows from Lemma B.1 and the fact that

$$\mathbf{e}^T\mathbf{x}/(\gamma q) \leq ||\mathbf{e}|| \cdot ||\mathbf{x}||/(\gamma q) \leq \sqrt{m} \cdot \beta q \cdot \omega(\sqrt{\log n})/(\gamma q) = 2 \cdot \sqrt{mn} \cdot \omega(\log n)/q$$

is a negligible function of n. □

Experiment 2: In this experiment, the vector $\mathbf{y}$ in the ciphertext is taken from the uniform distribution over $\mathbb{Z}_q^m$. Assuming the $\mathsf{DLWE}_{n,m,q,\beta}$ problem is hard, it immediately follows that the advantage of the adversary changes by at most a negligible amount.

Claim C.4. *If the* $\mathsf{DLWE}_{n,m,q,\beta}$ *problem is hard, then for every PPT algorithm* $\mathcal{A}$ *and for every function* $h \in \mathcal{H}$*, there is a negligible function* negl *such that* $\left|\mathsf{Adv}^{(1)}_{\mathcal{A},h}(n) - \mathsf{Adv}^{(2)}_{A,h}(n)\right| \leq \mathsf{negl}(n)$.

Experiment 3: In this experiment, the second component of the ciphertext is replaced by a uniformly random element $r \leftarrow \mathbb{Z}_q$. Namely, the ciphertext is generated as $(\mathbf{y}, r)$, where $\mathbf{y} \leftarrow \mathbb{Z}_q^m$ is uniformly random, and $r \leftarrow \mathbb{Z}_q$ is uniformly random.

Claim C.5. *For every PPT algorithm* $\mathcal{A}$ *and for every function* $h \in \mathcal{H}$*, there is a negligible function* negl *such that* $\left|\mathsf{Adv}^{(2)}_{\mathcal{A},h}(n) - \mathsf{Adv}^{(3)}_{\mathcal{A},h}(n)\right| \leq \mathsf{negl}(n)$.

Proof. We reduce the task of inverting h to the task of gaining a non-negligible distinguishing advantage between experiments 2 and 3. Suppose for the sake of contradiction that there exists a PPT algorithm $\mathcal{A}$, a function $h \in \mathcal{H}$, and a polynomial p such that for infinitely many n's, $\left|\mathsf{Adv}^{(2)}_{\mathcal{A},h}(n) - \mathsf{Adv}^{(3)}_{\mathcal{A},h}(n)\right| \geq 1/p(n)$. We show that this implies that there exists a PPT algorithm $\mathcal{B}$ so that for infinitely many n's,

$$\left|\Pr[\mathcal{B}(\mathbf{A}, \mathbf{u}, h(\mathbf{A}, \mathbf{e}), \mathbf{y}, \mathbf{e}^T\mathbf{y}) = 1] - \Pr[\mathcal{B}(\mathbf{A}, \mathbf{u}, h(\mathbf{A}, \mathbf{e}), \mathbf{y}, r) = 1]\right| \geq 1/p(n) \tag{4}$$

The adversary $\mathcal{B}$ will simulate $\mathcal{A}$, as follows. On input $(\mathbf{A}, \mathbf{u}, h(\mathbf{A}, \mathbf{e}), \mathbf{y}, c)$, algorithm $\mathcal{B}$ will choose a random bit $b \in \{0, 1\}$ and will start emulating $\mathcal{A}(PK, h(\mathbf{A}, \mathbf{e}))$, where $PK = (\mathbf{A}, \mathbf{u})$. The algorithm $\mathcal{B}$ will then sample $x' \leftarrow \overline{\Psi}_\gamma$ and a uniformly random bit $b \leftarrow \{0, 1\}$ and feed $\mathcal{A}$ the ciphertext $(\mathbf{y}, r + x' + b\lfloor q/2 \rfloor \pmod q)$. Let b' be the output of $\mathcal{A}$. If $b = b'$ then $\mathcal{B}$ outputs 1, and otherwise $\mathcal{B}$ outputs 0.

By definition

$$\Pr[\mathcal{B}(\mathbf{A}, \mathbf{u}, h(\mathbf{A}, \mathbf{e}), \mathbf{y}, \mathbf{e}^T\mathbf{y}) = 1] = \mathsf{Adv}^{(2)}_{\mathcal{A},h}(n),$$

and

$$\Pr[\mathcal{B}(\mathbf{A}, \mathbf{u}, h(\mathbf{A}, \mathbf{e}), \mathbf{y}, r) = 1] = \mathsf{Adv}^{(3)}_{\mathcal{A},h}(n).$$

This, together with the assumption that $\left|\mathsf{Adv}^{(2)}_{\mathcal{A},h}(n) - \mathsf{Adv}^{(3)}_{\mathcal{A},h}(n)\right| \geq 1/p(n)$, implies that Equation (4) holds. Now, we use Goldreich-Levin theorem over the (large) field $\mathbb{Z}_q$ and $H = \{0,1\} \subseteq \mathbb{Z}_q$ (Theorem 4.1). By Theorem 4.1, there is an algorithm that, given $PK = (\mathbf{A}, \mathbf{u})$, inverts $h(\mathbf{A}, \mathbf{e})$ with probability greater than

$$\frac{\delta^3}{512 \cdot n \cdot q^2} = q^n \cdot \frac{\delta^3 \cdot q}{512 \cdot n \cdot q^{n+3}} > q \cdot 2^{-m^\epsilon}$$

since $q^{n+3} = 2^{m^\epsilon}$ and $512 \cdot n/\delta^3 \cdot q < 1$ for large enough n. This provides the desired contradiction. □

The ciphertext in experiment 3 contains no information about the message. Thus, the adversary has no advantage in this experiment, i.e, $\mathsf{Adv}^{(3)}_{\mathcal{A},h}(n) = 0$. Putting together the claims, we get that $\mathsf{Adv}_{\mathcal{A},h}(n) \leq \mathsf{negl}(n)$. This concludes the proof of Theorem C.2. □

C.2 Weak Security for Polynomially Hard-to-Invert Auxiliary Inputs

In this section, we present a variant of the GPV cryptosystem that we call sparse GPV, and show that the system is weakly secure against any polynomially uninvertible auxiliary input function (i.e, $(\mathsf{negl}(n))$-`wAI-CPA` secure).

C.2.1 The Sparse GPV Cryptosystem

We present yet another variant of the dual Regev (aka, GPV) encryption scheme from [14]. As in Section C, we let n denote the security parameter. Let the prime $q = \mathsf{poly}(n)$, the integer $m = \mathsf{poly}(q)$ and the error-distribution $\overline{\Psi}_\beta$ where $\beta = 1/mathsfpoly(n)$ for some polynomials $\mathsf{poly}(\cdot)$ be parameters of the system.

$\mathsf{Gen}(1^n)$: Choose a uniformly random matrix $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ and a random vector $\mathbf{e} \leftarrow \{0,1\}^m$, such that $\mathbf{e}$ has exactly $n/2$ ones and $m - n/2$ zeros. Compute $\mathbf{u} = \mathbf{A}\mathbf{e}$. The public-key is $PK := (\mathbf{A}, \mathbf{u})$ and the secret-key is $SK := \mathbf{e}$. As before, $\mathbf{A}$ can be considered a public parameter.

$\mathsf{Enc}(PK, b)$, where b is a bit, works as follows. Choose a random vector $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ and a vector $\mathbf{x} \leftarrow \overline{\Psi}_\beta^m$. Output the ciphertext

$$\left(\mathbf{A}^T\mathbf{s} + \mathbf{x}, \mathbf{u}^T\mathbf{s} + b\left\lfloor \frac{q}{2} \right\rfloor\right)$$

$\mathsf{Dec}(SK, c)$: The decryption algorithm is exactly as in Section C.

The scheme here differs from the scheme in [14] (as well as the scheme in Section C) in two respects: first and most significantly, the secret-key is a *sparse* vector from $\{0,1\}^m$ (rather than drawn from a discrete Gaussian distribution $D_{\mathbb{Z}^m,r}$ for some $r > 0$ in GPV and a uniformly random vector in $\{0,1\}^m$ in Section C), and secondly, the final component of the ciphertext does not use an extra noise component.

Correctness of the encryption scheme follows by an argument analogous to the one in Section C.

C.2.2 New Lemmas on the LWE Problem

Before we prove the auxiliary-input security of the encryption scheme described above, we show a lemma on the hardness of the $\mathsf{DLWE}_{q,\beta}$ problem if the adversary is given (specific types of) auxiliary information about the secret $\mathbf{s}$ (as well as the error $\mathbf{x}$). The lemma states that $\mathsf{DLWE}_{n,m,q,\beta}$ remains hard even if the distinguisher is given any (sufficiently short) *linear function* of the secret $\mathbf{s}$.

Lemma C.6. *If the* $\mathsf{DLWE}_{n-k,m,q,\beta}$ *problem is hard, then for every rank-*k *matrix* $\mathbf{B} \in \mathbb{Z}_q^{n\times k}$*,*

$$(\mathbf{A}, \mathbf{B}^T\mathbf{s}, \mathbf{A}^T\mathbf{s} + \mathbf{x}) \approx (\mathbf{A}, \mathbf{u}_1, \mathbf{u}_2).$$

The proof of the lemma relies on the following claim, which can be thought of as a special case of the lemma where the distinguisher is given k coordinates of the secret $\mathbf{s}$ (instead of an arbitrary linear function of $\mathbf{s}$).

Claim C.7. *If the* $\mathsf{DLWE}_{n-k,m,q,\beta}$ *problem is hard, then for every set of indices* $I \subseteq [n]$ *of size* $|I| = k$*,*

$$(\mathbf{A}, (s_i)_{i\in I}, \mathbf{A}^T\mathbf{s} + \mathbf{x}) \approx (\mathbf{A}, (s_i)_{i\in I}, \mathbf{u}),$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$*,* $\mathbf{s} \leftarrow \mathbb{Z}_q^n$*,* $\mathbf{x} \leftarrow \overline{\Psi}_\beta^m$*, and* $\mathbf{u} \leftarrow \mathbb{Z}_q^m$*.*

Proof. Assume for the sake of simplicity of notation that $I = \{1, \ldots, k\}$. Denote by $\mathbf{A}_1$ the first k rows of $\mathbf{A}$, and denote by $\mathbf{A}_2$ the last $n-k$ rows of $\mathbf{A}$. Similarly, denote by $\mathbf{s}_1$ the first k coordinates of $\mathbf{s}$, and denote by $\mathbf{s}_2$ the last $n-k$ coordinates of $\mathbf{s}$. Then,

$$\mathbf{A}^T\mathbf{s} + \mathbf{x} = \mathbf{A}_1^T\mathbf{s}_1 + \mathbf{A}_2^T\mathbf{s}_2 + \mathbf{x}.$$

Using this notation, we need to prove that

$$(\mathbf{A}_1, \mathbf{A}_2, \mathbf{s}_1, \mathbf{A}_1^T\mathbf{s}_1 + \mathbf{A}_2^T\mathbf{s}_2 + \mathbf{x}) \approx (\mathbf{A}_1, \mathbf{A}_2, \mathbf{s}_1, \mathbf{u}). \tag{5}$$

The hardness of $\mathsf{DLWE}_{n-k,m,q,\beta}$ implies that

$$(\mathbf{A}_2, \mathbf{A}_2^T\mathbf{s}_2 + \mathbf{x}) \approx (\mathbf{A}_2, \mathbf{u}).$$

This, together with the fact that $\mathbf{A}_1$ and $\mathbf{s}_1$ can be sampled efficiently given $(\mathbf{A}_2, \mathbf{A}_2\mathbf{s}_2 + \mathbf{x})$, proves Equation (5). □

Proof of Lemma C.6. We first note that if each column of $\mathbf{B}$ was a vector in the standard basis (i.e., each column of $\mathbf{B}$ was of the form $(0, \ldots, 0, 1, 0, \ldots, 0)$), then the proof of Lemma C.6 would follow immediately from Claim C.7. For the case of a general matrix $\mathbf{B}$, the idea is to do a change of basis, and reduce Lemma C.6 to Claim C.7.

Let $\mathbf{C}$ be an (arbitrary) invertible matrix in $\mathbb{Z}_q^{n\times n}$ such that the first k columns of $\mathbf{C}$ are equal to $\mathbf{B}$. Denote by $\mathbf{s}' \triangleq \mathbf{C}^T\mathbf{s}$. Namely $\mathbf{s}'$ is the vector $\mathbf{s}$ presented in a different basis. Then,

$$\mathbf{A}^T\mathbf{s} = \mathbf{A}^T(\mathbf{C}^{-1})^T\mathbf{C}^T\mathbf{s} = (\mathbf{C}^{-1}\mathbf{A})^T(\mathbf{C}^T\mathbf{s}) = (\mathbf{C}^{-1}\mathbf{A})^T\mathbf{s}'$$

Denote by $\mathbf{s}'_1$ the first k coordinates of $\mathbf{s}'$, and denote by $\mathbf{s}'_2$ the last $n-k$ coordinates of $\mathbf{s}'$. Then,

$$(\mathbf{A}, \mathbf{B}^T\mathbf{s}, \mathbf{A}^T\mathbf{s} + \mathbf{x}) = (\mathbf{A}, \mathbf{s}'_1, (\mathbf{C}^{-1}\mathbf{A})^T\mathbf{s}' + \mathbf{x}). \tag{6}$$

The fact that $\mathbf{s}$ is uniformly distributed in $\mathbb{Z}_q^n$, together with the fact that $\mathbf{C}$ is invertible, implies that $\mathbf{s}'$ is also uniformly distributed in $\mathbb{Z}_q^n$. This, together with Claim C.7, and together with the fact that $\mathbf{C}$ is fixed and known in advance, implies that

$$(\mathbf{A}, \mathbf{s}'_1, (\mathbf{C}^{-1}\mathbf{A})^T\mathbf{s}' + \mathbf{x}) \approx (\mathbf{A}, \mathbf{u}_1, \mathbf{u}_2).$$

This, together with Equation (6), implies that

$$(\mathbf{A}, \mathbf{B}^T\mathbf{s}, \mathbf{A}^T\mathbf{s} + \mathbf{x}) \approx (\mathbf{A}, \mathbf{u}_1, \mathbf{u}_2),$$

as desired. This concludes the proof of Lemma C.6. □

In the proof of auxiliary input security, we use the following corollary of Lemma C.6, which states that the $\mathsf{DLWE}_{n,m,q,\beta}$ problem remains hard even given a *linear function* of the error-term $\mathbf{x}$.

Corollary C.8. *If the* $\mathsf{DLWE}_{n-k,m,q,\beta}$ *problem is hard, then*

$$\{\mathbf{A}, \mathbf{A}^T\mathbf{s} + \mathbf{x}, \mathbf{e}, \mathbf{e}^T\mathbf{x}\} \approx \{\mathbf{A}, \mathbf{u}, \mathbf{e}, \mathbf{e}^T\mathbf{x}\}$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$, $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, $\mathbf{x} \leftarrow \overline{\Psi}_\beta^m$, $\mathbf{u} \leftarrow \mathbb{Z}_q^m$, *and* $\mathbf{e} \leftarrow \{0,1\}^m$ *such that the number of ones is exactly* k.

Proof. Let $I \triangleq \{i : e_i = 1\}$. By definition $|I| = k$. In what follows we prove that Lemma C.6 implies even the stronger statement that

$$(\mathbf{A}, \mathbf{A}^T\mathbf{s} + \mathbf{x}, \mathbf{e}, \{x_i\}_{i\in I}) \approx (\mathbf{A}, \mathbf{u}, \mathbf{e}, \{x_i\}_{i\in I}) \tag{7}$$

Assume for the sake of simplicity of notation that $I = \{1, \ldots, k\}$. Denote by $\mathbf{A}_1$ the first k columns in $\mathbf{A}$ and denote by $\mathbf{A}_2$ the remaining $m-k$ columns of $\mathbf{A}$. Similarly, we denote by $\mathbf{x}_1$ the first k coordinates of the vector $\mathbf{x}$, and by $\mathbf{x}_2$ the remaining $m-k$ coordinates of $\mathbf{x}$. Thus, Equation (7) can be rewritten as

$$(\mathbf{A}, \mathbf{A}_1^T\mathbf{s} + \mathbf{x}_1, \mathbf{A}_2^T\mathbf{s} + \mathbf{x}_2, \mathbf{e}, \mathbf{x}_1) \approx (\mathbf{A}, \mathbf{u}_1, \mathbf{u}_2, \mathbf{e}, \mathbf{x}_1), \tag{8}$$

where $\mathbf{u}_1 \leftarrow \mathbb{Z}_q^k$ and $\mathbf{u}_2 \leftarrow \mathbb{Z}_q^{m-k}$.

The fact that $\mathbf{e}$ and $\mathbf{x}_1$ are efficiently sampleable implies that it suffices to prove that

$$(\mathbf{A}, \mathbf{A}_1^T\mathbf{s}, \mathbf{A}_2^T\mathbf{s} + \mathbf{x}_2) \approx (\mathbf{A}, \mathbf{u}_1, \mathbf{u}_2),$$

which follows immediately from Lemma C.6.

□

C.2.3 Proof of Auxiliary Input Security

Theorem C.9. *Let the parameters* m, q, β *and* γ *be as in the encryption scheme in Section C.2.1. Assuming that the* $\mathsf{DLWE}_{n/2,m,q,\beta}$ *problem is hard, the encryption scheme above is* $(\mathsf{negl}(n))$-$\mathtt{wAI\text{-}CPA}$ *secure. (since the min-entropy* k *of the secret key is polynomial in* n*, the scheme is* $(\mathsf{negl}(k))$-$\mathtt{wAI\text{-}CPA}$ *secure as well).*

Proof of Theorem C.9. Fix any auxiliary-input function h, so that $\mathbf{e}$ is still polynomially-hard to invert $(\mathbf{A}, \mathbf{u}, h(\mathbf{A}, \mathbf{e}))$, and a PPT adversary $\mathcal{A}$ with $\delta = \delta(n) = \mathsf{Adv}_{\mathcal{A},h}(n)$. We consider a sequence of experiments, and let $\mathsf{Adv}^{(i)}_{\mathcal{A},h}(n)$ denote the advantage of the adversary in experiment i.

Experiment 0**:** This is the experiment in Definition 1. The adversary $\mathcal{A}$ gets as input $PK = (\mathbf{A}, \mathbf{u})$ and the auxiliary input $h(\mathbf{A}, \mathbf{e})$. $\mathcal{A}$ receives $\mathsf{Enc}(PK, b)$ where $b \in \{0,1\}$ is uniformly random. $\mathcal{A}$ succeeds in the experiment if he succeeds in guessing b. By assumption, $\mathsf{Adv}^{(0)}_{\mathcal{A},h}(n) = \mathsf{Adv}_{\mathcal{A},h}(n) = \delta$.

Experiment 1**:** In this experiment, the challenge ciphertext is generated by encrypting with the secret key, rather than with the usual $\mathsf{Enc}(PK, b)$ algorithm. In particular, define the algorithm $\mathsf{Enc}'(PK, \mathbf{e}, b)$ as follows.

$$\mathsf{Enc}'(PK, \mathbf{e}, b) \triangleq (\mathbf{y}, \mathbf{e}^T\mathbf{y} - \mathbf{e}^T\mathbf{x} + b\lfloor q/2 \rfloor)$$

where $\mathbf{y} = \mathbf{A}^T\mathbf{s} + \mathbf{x}$. Clearly, the distribution produced by Enc' is identical to the distribution of a real ciphertext, and thus $\mathsf{Adv}^{(1)}_{\mathcal{A},h}(n) = \mathsf{Adv}^{(0)}_{\mathcal{A},h}(n)$.

Experiment 2**:** In this experiment, the vector $\mathbf{y}$ in the ciphertext is taken from the uniform distribution over $\mathbb{Z}_q^m$. Using Corollary C.8, we show that the advantage of the adversary changes by at most a negligible amount. More precisely,

Claim C.10. *Assume that the* $\mathsf{DLWE}_{n/2,m,q,\beta}$ *problem is hard. Then for every PPT algorithm* $\mathcal{A}$ *and for every function* $h \in \mathcal{H}$*, there is a negligible function* negl *such that*

$$\left|\mathsf{Adv}^{(2)}_{\mathcal{A},\Pi,h}(n) - \mathsf{Adv}^{(1)}_{\mathcal{A},\Pi,h}(n)\right| \leq \mathsf{negl}(n)$$

Proof. The proof works in two stages. First, we consider two distributions D_1 and D_2 (defined below), and show that any adversary $\mathcal{A}$ that distinguishes between experiments 1 and 2 with non-negligible probability can be used to construct an adversary $\mathcal{B}$ that distinguishes between D_1 and D_2.

D_1 : Output $(\mathbf{A}, \mathbf{A}^T\mathbf{s} + \mathbf{x}, \mathbf{e}, \mathbf{e}^T\mathbf{x}) : \mathbf{s} \leftarrow \mathbb{Z}_q^n; \mathbf{x} \leftarrow \overline{\Psi}_\beta; \mathbf{e} \leftarrow \{0,1\}^m$ s.t. $\mathbf{e}$ has exactly $n/2$ ones

D_2 : Output $(\mathbf{A}, \mathbf{u}, \mathbf{e}, \mathbf{e}^T\mathbf{x}) : \mathbf{u} \leftarrow \mathbb{Z}_q^m; \mathbf{x} \leftarrow \overline{\Psi}_\beta; \mathbf{e} \leftarrow \{0,1\}^m$ s.t. $\mathbf{e}$ has exactly $n/2$ ones

We then invoke Corollary C.8 which says that distinguishing between D_0 and D_1 is at least as hard as solving $\mathsf{DLWE}_{n/2,m,q,\beta}$. □

The rest of the proof is similar to the proof of Theorem C.2.

Experiment 3**:** In this experiment, we again change the way the ciphertext is generated. In particular, the component $\mathbf{e}^T(\mathbf{y} - \mathbf{x})$ in the ciphertext is replaced by a uniformly random element $u \leftarrow \mathbb{Z}_q$. i.e, the ciphertext is generated as $\left(\mathbf{y}, u + b\lfloor q/2 \rfloor\right)$, where $\mathbf{y} \leftarrow \mathbb{Z}_q^m$ is uniformly random, and $u \leftarrow \mathbb{Z}_q$ is uniformly random.

Claim C.11. *For every PPT algorithm* $\mathcal{A}$ *and for every function* $h \in \mathcal{H}$*, there is a negligible function* negl *such that* $\left|\mathsf{Adv}^{(2)}_{\mathcal{A},h}(n) - \mathsf{Adv}^{(3)}_{\mathcal{A},h}(n)\right| \leq \mathsf{negl}(n)$.

Proof: Analogous to Claim C.5, we can show that any PPT adversary $\mathcal{A}$ that distinguishes between experiments 2 and 3 with advantage $\delta \geq 1/p(n)$ (for some polynomial $p(n)$) can be used to construct an inverter $\mathcal{B}$ that, given $PK = (\mathbf{A}, \mathbf{u})$, inverts $h(\mathbf{A}, \mathbf{s})$ with probability

$$\frac{\delta^3}{512 \cdot n \cdot q^2} \stackrel{\mathsf{def}}{=} \frac{1}{p'(n)}$$

where $p'(n) \stackrel{\mathsf{def}}{=} 512 \cdot n \cdot p(n)^3 \cdot q^2$ is polynomial in n since q is polynomial in n. Since the min-entropy k of the secret key is polynomial in n, the inversion probability is polynomial in k as well. □

The ciphertext in experiment 3 contains no information about the message. Thus, the adversary has no advantage in this experiment, i.e, $\mathsf{Adv}^{(3)}_{\mathcal{A},h}(n) = 0$. Putting together the claims, we get that $\mathsf{Adv}_{\mathcal{A},h}(n) \leq \mathsf{negl}(n)$. □

D Proof of Lemma 3.1

Proof. The first part is obvious, since auxiliary input CPA security protects against a strictly larger class of functions than weak auxiliary input CPA security. Indeed, if SK is $f(k)$-hard to compute given $(PK, h(SK, PK))$, then it is certainly $f(k)$-hard given $h(SK, PK)$ alone. In other words, $\mathcal{H}_{\mathtt{pk\text{-}ow}}(f(k)) \subseteq \mathcal{H}_{\mathtt{ow}}(f(k))$.

The second part follows from the fact that PK is only $t = t(k)$ bits long, so it is always possible to guess it with probability 2^{-t}. Namely, if SK is "$f(k)$-easy" given $(PK, h(SK, PK))$, then it is certainly "$(2^{-t}f(k))$-easy" given $h(SK, PK)$ alone, by simply guessing the value PK. In other words, $\mathcal{H}_{\mathtt{ow}}(2^{-t}f(k)) \subseteq \mathcal{H}_{\mathtt{pk\text{-}ow}}(f(k))$.

To show the third part, let $\ell(k) = k - \log(1/f(k))$. The claim will follow if we show that $\mathcal{H}_{\mathtt{bdd}}(\ell(k)) \subseteq \mathcal{H}_{\mathtt{ow}}(2^{\ell(k)-k})$; namely, learning at most $\ell(k)$ bits of information $h(SK, PK)$ about a secret key SK of min-entropy k, makes it hard to predict with probability at least $2^{\ell(k)}/2^k$. Indeed, this is so since learning $\ell(k)$ bits can only increase the best prediction probability $1/2^k$ by at most a factor $2^{\ell(k)}$.

The fourth part follows immediately by combining the second and third parts. □