Public-Key Encryption Schemes with Auxiliary Inputs

Yevgeniy Dodis[1,⋆], Shafi Goldwasser[2,⋆⋆], Yael Tauman Kalai[3], Chris Peikert[4,⋆⋆⋆], and Vinod Vaikuntanathan[5]

[1] New York University
[2] MIT and Weizmann Institute
[3] Microsoft Research
[4] Georgia Institute of Technology
[5] IBM Research

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 (DDH) and the Learning with Errors (LWE) problems.

As an independent technical contribution, we extend the Goldreich-Levin theorem to provide a hard-core (pseudorandom) value over *large* fields.

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 [5,15,18], a new goal has been set within the theory of cryptography community to build a general theory 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 [5,15,18,9,1,2,20,8,24,23,14,9,10] and the references therein).

Placing the current paper within this body of work, we focus on side channel attacks which result from "memory leakages" [1,8,2,20,16]. 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

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-3-642-11799-2_36

[⋆] Supported in part by NSF grants CNS-0831299 and CNS-0716690.

[⋆⋆] Supported in part by NSF grants CCF-0514167, CCF-0635297, NSF-0729011, the Israel Science Foundation 700/08 and the Chais Family Fellows Program.

[⋆⋆⋆] Supported in part by NSF grant CNS-0716786.

D. Micciancio (Ed.): TCC 2010, LNCS 5978, pp. 361–381, 2010.

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.

Akavia, Goldwasser and Vaikuntanathan [1] considered a model in which the leakage function h is an arbitrary polynomial-time computable function *with bounded output length*. Letting k denote the length (or more generally, the min-entropy) of the secret key SK, the restriction is that h outputs $\ell(k) < k$ bits. In particular, this ensures that the leakage does not fully reveal the secret key. Akavia et al. [1] show that the public-key encryption scheme of Regev [25] 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 [20] relaxed the restriction on h so that the leakage observed by the adversary may be longer than the secret key, but *the min-entropy of the secret drops by at most* $\ell(k)$ *bits* upon observing $h(SK)$; they call this the *noisy leakage* requirement. The work of [20] 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 universal hash-proof system [7]. We refer the reader to [20] for a detailed discussion of these results. (Finally, we note that the proof of [1] based on the LWE assumption generalizes to the case of noisy leakage.).

The bottom line is that both [1] and [20] (and the results that use the models therein) interpret the necessary restriction on the leakage function h by insisting that it is (information-theoretically) *impossible* to recover SK given the leakage $h(SK)$.

1.1 The Auxiliary Input Model

The natural question that comes out of the modeling in [1,20] is whether this restriction is essential. For example, is it possible to achieve security if the leakage function h is a one-way permutation? Such a function information-theoretically reveals the entire secret key SK, but still it is computationally infeasible to recover SK from $h(SK)$.

The focus of this work is the model of *auxiliary input* leakage functions, introduced by Dodis, Kalai, and Lovett [8], generalizing [1,20]. 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 *hard to compute* given $h(SK)$ (but not necessarily impossible to determine, as implied by the definitions of [1,20]). Formally, the restriction imposed by [8] on the leakage function h is 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 $\lambda(k)$ allows for a larger class of functions, 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]

The work of [8] constructed, based on a non-standard variant of the learning parity with noise (LPN) assumption, a symmetric-key encryption scheme that remains 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 [8], 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 [4,8]. 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 (computationally) reveal 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 [8] to the case of public-key encryption algorithms. 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 $h(SK, PK)$. In other words, we allow the leakage function to depend on PK.

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

1. 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, [20] showed that the BHHO scheme is secure w.r.t. bounded length leakage of size at most $k - k^\epsilon$ (and more generally, noisy leakages).
2. We show that the "dual" of Regev's encryption scheme [25], first proposed by Gentry, Peikert and Vaikuntanathan [12], 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 errors" (LWE) problem. This result, in particular, implies that the scheme is

[1] 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.

secure w.r.t. bounded length leakage of size at most $k - k^{\epsilon}$ (and more generally, noisy leakages). This improves on the previous bound of [1] for the [25] system.

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/DDH are secure against an adversary that runs in subexponential time.

Of course, the holy grail in this line of work would be a public-key encryption scheme secure against *polynomially hard auxiliary inputs*, that is functions that no polynomial-time adversary can invert with probability better than negligible (in k). Is this in fact achievable? We give two answers to this question.

A Negative Answer. We show that the holy grail is unattainable for public-key encryption schemes. In particular, for every *public-key encryption scheme*, we show an auxiliary input function h such that it is hard to compute SK given $h(PK, SK)$, and yet the scheme is completely insecure in the presence of the leakage given by h. The crux of the proof is to use the fact that the adversary that tries to break the encryption scheme gets *the public key PK in addition to the leakage $h(PK, SK)$*. Thus, it suffices to come up with a leakage function h such that:

- Given $h(PK, SK)$, it is hard to compute SK (i.e., h is uninvertible), and yet
- Given $h(PK, SK)$ and PK, it is easy to compute SK.

We defer the details of the construction of such a function to the full version.

A Weaker Definition and a Positive Answer. To complement the negative result, we construct a public key encryption schemes (under DDH and LWE) that are secure if the leakage function h is polynomially hard to invert *even given the public key*. This is clearly a weaker definition, but still it is advantageous in the context of composition. See Section 3.1 for definitional details, and the full version for the actual scheme.

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/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 $2^{-k^{\epsilon}}$. (In fact, this is the case for most of the known schemes, in particular [1,2,20,16].)

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, we first choose $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ uniformly at random, $\mathbf{x} \in \mathbb{Z}_q^m$ from a "narrow" error distribution, and $x' \in \mathbb{Z}_q$ from a "much wider" error distribution. The ciphertext is

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

Given the secret key $\mathbf{e}$, the decryption algorithm computes $\mathbf{e}^T(\mathbf{A}^T\mathbf{s} + \mathbf{x}) \approx (\mathbf{A}\mathbf{e})^T\mathbf{s} = \mathbf{u}^T\mathbf{s}$ (where the approximation holds because $\mathbf{e}^T\mathbf{x}$ 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) refers "almost entirely" to the "preamble" of the ciphertext $\mathbf{A}^T\mathbf{s} + \mathbf{x}$, and not to the secret key at all. This suggests considering an alternative encryption algorithm (used by the simulator) which generates a ciphertext using knowledge of the secret key $\mathbf{e}$ rather than the secret $\mathbf{s}$. The advantage of this in the context of leakage is that knowing the secret key enables the simulator to compute and reveal an arbitrary (polynomial-time computable) leakage function h of the secret key.[2] More specifically, we consider an alternate encryption algorithm that, given a preamble $\mathbf{y} = \mathbf{A}^T\mathbf{s} + \mathbf{x}$, encrypts the bit b using the secret key $\mathbf{e}$ as:

$$(\mathbf{y}, \mathbf{e}^T\mathbf{y} + x' + b\lfloor q/2 \rfloor).$$

The distribution thus produced is statistically "as good as" that of 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 (up to negligible statistical distance) like a sample from the "wide" error distribution when $\mathbf{e}^T\mathbf{x}$ is negligible compared to x'. (This is why we need to use a slightly superpolynomial q, and to choose $\mathbf{e}$ and $\mathbf{x}$ to be negligible relative to the magnitude of x').

Next, by the LWE assumption, we can replace $\mathbf{y}$ with a uniformly random vector over $\mathbb{Z}_q^m$. We would then like to show that the term $\mathbf{e}^T\mathbf{y} = \langle \mathbf{e}, \mathbf{y} \rangle$ in the second component of the ciphertext is pseudorandom given the rest of the adversary's view, namely $(\mathbf{A}, \mathbf{A}\mathbf{e}, h(\mathbf{A}, \mathbf{e}), \mathbf{y})$ where $\mathbf{y} \in \mathbb{Z}_q^m$ is uniformly random. Assuming that the function $h'(\mathbf{A}, \mathbf{e}) = (\mathbf{A}, \mathbf{A}\mathbf{e}, h(\mathbf{A}, \mathbf{e}))$ is uninvertible, this suggests using a *Goldreich-Levin type theorem over the large field* $GF(q)$. Providing such a theorem is the first technical contribution of this work.

The original Goldreich-Levin theorem over the binary field $GF(2)$ says that for an uninvertible function $h : GF(2)^m \to \{0,1\}^*$, the inner product $\langle \mathbf{e}, \mathbf{y} \rangle \in GF(2)$ is pseudorandom given $h(\mathbf{e})$ and uniformly random $\mathbf{y} \in GF(2)^m$. The later work of [13] extends this result to deal with inner products over $GF(q)$ for a general prime q. In particular, it shows that any PPT algorithm that distinguishes between $\langle \mathbf{y}, \mathbf{e} \rangle$ and

[2] This kind of technique for proving security of public-key encryption was already used in [12], in the context of leakage in [20,16], and to our knowledge, traces at least as far back as the Cramer-Shoup CCA-secure encryption scheme [6].

uniform, given $h(\mathbf{e})$ and $\mathbf{y}$, gives rise to a $\text{poly}(q)$-*time* algorithm that inverts $h(\mathbf{e})$ with probability $1/\text{poly}(q)$ (for a more detailed comparison of our result with [13], see Remark 2). When q is super-polynomial in the main security parameter n, 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 each coordinate of $\mathbf{e}$ comes from a much smaller subdomain $H \subseteq GF(q)$. 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 a $\text{poly}(m)$-sized subset of $GF(q)$. Let $h : H^m \to \{0,1\}^*$ be any (possibly randomized) function. If there is a PPT algorithm $\mathcal{D}$ that distinguishes between $\langle \mathbf{e}, \mathbf{y} \rangle$ and the uniform distribution over $GF(q)$ given $h(\mathbf{e})$ and $\mathbf{y} \leftarrow GF(q)^m$, then there is a PPT algorithm $\mathcal{A}$ that inverts h with probability $1/(q^2 \cdot \text{poly}(m))$.*

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 $(\mathbf{A}, \mathbf{Ae}, h(\mathbf{A}, \mathbf{e}))$; we call this 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 bit-length $n \log q = m^\epsilon \ll |SK|$, the reduction can simply guess the value of $PK = \mathbf{Ae}$ and lose only a factor of 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 any pesky error terms (and the resulting statistical deviation between encrypting with the public key versus the secret key), and (2) we use the Goldreich-Levin theorem over an exponentially large field $GF(q)$, rather than a superpolynomial one.

2 Preliminaries

Throughout this paper, we denote the security parameter by n. We write $\text{negl}(n)$ 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 [19] 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 1 **([19]).** *Under the DDH assumption on $\mathcal{G}$, for any positive integer m,*

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

The Learning with errors (LWE) Assumption. The LWE problem was introduced by Regev [25] 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 1. *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 [25] 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 the number of samples 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 use a compact matrix notation to describe the LWE problem $\mathsf{LWE}_{n,m,q,\chi}$: 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 a similar 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 two basic facts about Gaussians (tailored to the error distribution $\overline{\Psi}_\beta$); see, e.g. [11].

Lemma 2. *Let $\beta > 0$ and $q \in \mathbb{Z}$. Let the vector $\mathbf{x} \in \mathbb{Z}^n$ be arbitrary, and $\mathbf{y} \leftarrow \overline{\Psi}_\beta^n$. With overwhelming probability over the choice of $\mathbf{y}$, $\left|\mathbf{x}^T\mathbf{y}\right| \leq ||\mathbf{x}|| \cdot \beta q \cdot \omega(\sqrt{\log n})$.*

Lemma 3. *Let $\beta > 0$, $q \in \mathbb{Z}$ and $y \in \mathbb{Z}$. 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 [25], 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 [21] also gave a related *classical* reduction for similar parameters.

3 Security against Auxiliary Inputs

We start by defining security of public-key encryption schemes w.r.t. auxiliary input.

Definition 2. *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* $n \in \mathbb{N}$*,*

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

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

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

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 two 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 $\mathcal{H}_{\mathsf{bdd}}$ is the length-bounded family studied by the prior work [1,20],[3] while the second family $\mathcal{H}_{\mathsf{ow}}$ is the auxiliary-input family we introduce and study in this work, where we only assume that the secret key is "hard to compute" given the leakage.

- Let $\mathcal{H}_{\mathsf{bdd}}(\ell(k))$ be the class of all polynomial-time computable functions $h : \{0,1\}^{|SK|+|PK|} \to \{0,1\}^{\ell(k)}$, where $\ell(k) \leq k$ is the number of bits the attacker is allowed to learn. If a public-key encryption scheme Π is CPA secure w.r.t. this family of functions, it is called $\ell(k)$-`LB-CPA (length-bounded CPA) secure`.
- Let $\mathcal{H}_{\mathsf{ow}}(f(k))$ 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

[3] For simplicity, we do not define a more general family corresponding to the noisy leakage model of [20]. However, all the discussion, including Lemma 4, easily holds for noisy-leakage instead of length-bounded leakage.

generated (SK, PK)), no PPT algorithm can find SK with probability greater than $f(k)$, where $f(k) \geq 2^{-k}$ is the hardness parameter. If a public-key encryption scheme Π is CPA secure w.r.t. this family of functions, it is called $f(k)$-AI-CPA (auxiliary input CPA) secure. Our goal is to make $f(k)$ as large (i.e., as close to $\mathrm{negl}(k)$) as possible.

We also consider a weaker notion of auxiliary input security, called $f(k)$-wAI-CPA (weak auxiliary input CPA) security, where the class of functions under consideration is uninvertible *even given the public key of the scheme*. This notion is used as a stepping stone to achieving the stronger $f(k)$-AI-CPA notion.

- Let $\mathcal{H}_{\texttt{pk-ow}}(f(k))$ 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 $f(k) \geq 2^{-k}$ is the hardness parameter. If a public-key encryption scheme Π is CPA secure w.r.t. this family of functions, it is called $f(k)$-wAI-CPA (weak auxiliary input CPA) secure.

The following lemma shows various relations between these notions of security. The proof follows directly from the definitions, and is omitted.

Lemma 4. *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.

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

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, as shown in Part 3. of Lemma 4, it immediately implies that 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 [1,20]. Second, 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. Here, we further restrict the leakage functions h to ones where SK remains hard to compute *given both the leakage $y = h(PK, SK)$ 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. (In the full version, we also give more convincing examples.)

Although this shows that the wAI-CPA security should be taken with care, we show it is still very useful. First, Lemma 4 shows that it is useful *when the scheme has a short public key*. 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 signature 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. Finally, we are able to construct $f(k)$-wAI-CPA secure schemes with the *optimal* value of $f(k)$, namely $f(k) = \mathrm{negl}(k)$. We defer the details to the full version.

Public Parameters. For simplicity, when defining auxiliary input security, 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 also 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. $(PK, h(SK, PK, \texttt{params}))$).[4] Correspondingly, when applying Lemma 4, 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 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*

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

[4] 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.

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 1. 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.

Remark 2. We briefly compare our new variant of the GL Lemma for general q with the similar-looking extension of Goldreich, Rubinfeld and Sudan [13]. The extensions are incomparable, in the following sense. In [13], 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 [13] 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 [13] 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* $|H|$.

Overview of the Proof. As in the standard Goldreich-Levin proof, we concentrate on the vectors $\mathbf{s}$ on which the distinguisher has $\Omega(\epsilon)$-advantage for random $\mathbf{r}$. Let c be a parameter that depends on $|H|, n, \epsilon$ (this parameter will be specified precisely in the formal proof below). 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 its success probability in the process. Then, assuming all our c guessed inner products are correct, for each $i = 1 \ldots n$ and $a \in H$, we use the assumed distinguisher $\mathcal{D}$ to design an efficient procedure to test, with high probability, if $s_i = a$. The details of this test, which is the crux of the argument, is given in the formal proof below, but the above structure explains why our running time only depends on $|H|$ and not q.

Proof. We will actually prove a tighter version of the bound stated in Equation (1), and design an inverter $\mathcal{A}$ with success probability $\epsilon/4q^c$, 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

$$4q^c \leq 4\max(q^2, q \cdot (128|H|n/\epsilon^2)) = 4q \cdot \max(q, 128|H|n/\epsilon^2) \leq 512q^2 n/\epsilon^2.$$

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 $q^c \leq q \cdot (128|H|n/\epsilon^2)$, and thus the above probability is at least $\epsilon^3/512nq|H| \geq \epsilon^3/512nq^2$.

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

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

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 $m \overset{\text{def}}{=} 128|H|n/\epsilon^2$. 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, such that every two elements in S are linearly independent. This can be achieved for example by choosing $S \subseteq GF(q)^c \backslash \{0^c\}$ to be an arbitrary set of cardinality m such that the first coordinate of each element in S equals 1. 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 $g_1, \ldots, g_c \leftarrow GF(q)$.
   [Remark: Informally, the g_i are A's "guesses" 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 $p^{(i,a)} \geq \gamma_{\mathbf{s},y}$, set $s_i := a$ and move to the next $i+1$.
  Otherwise, move to the next guess $a \in H$.

5. Output $\mathbf{s} = s_1 s_2 \dots s_n$ (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 $h_{\bar{\rho}} = \langle \mathbf{r}_{\bar{\rho}}, \mathbf{s} \rangle$.

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

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 the 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 4 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*. This follows from their definition in Equation 3, since every two elements in S are linearly independent and we 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 4 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 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 $\mathbf{r}, u$ are pairwise independent.[5] 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 $1/(m(\epsilon/8)^2) = 1/2|H|n$.

This suffices to prove our result, since, by the union bound over all $i \in [1, \dots, n]$ 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.

[5] The argument is the same for $\mathbf{r}$ and for the values u, Claim 4 shows that they are in fact completely independent.

Claim. 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. 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 4. 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

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

Let $\mathbf{r} := \mathbf{r}_{\bar{\rho}} + \tau_{\bar{\rho}}^{(i,a)} \cdot \mathbf{e}_i$ and $u := h_{\bar{\rho}} + \tau_{\bar{\rho}}^{(i,a)} \cdot a$, 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 $t_i \neq 0$ and the value $\tau_{\bar{\rho}}^{(i,a)}$ was random in $GF(q)$, this means that $u = \langle \mathbf{r}, \mathbf{s} \rangle + \tau_{\bar{\rho}}^{(i,a)} \cdot t_i$ is random and independent of $\mathbf{r}$, as claimed. □

This concludes the proof of Theorem 1.

5 Auxiliary Input Secure Encryption Schemes

5.1 Construction Based on the DDH Assumption

We show that the BHHO encryption scheme [3] is secure against subexponentially hard-to-invert auxiliary input.

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

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

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

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

We now show that the BHHO scheme is secure against subexponentially hard auxiliary inputs, under the DDH assumption.

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

Remark. We can actually handle a richer class of auxiliary functions, namely, any $h(\mathbf{g}, \mathbf{s})$ that (given $\mathbf{g}$) is to hard invert with probability $1/2^k$, where k can be as small as $\text{polylog}(m)$. However, then the assumption we rely on is that DDH 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 DDH hardness assumption.

Proof (of Theorem 2). By Lemma 4 (Part 2) and because the length of "user-specific" public key y is $\log q$ bits, to prove the theorem it suffices to show that our encryption scheme is $(q 2^{-m^{\epsilon}})$-`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, letting $\mathsf{Adv}_{\mathcal{A},h}^{(i)}(n)$ denote the advantage of the adversary in experiment i.

Experiment 0*:* This is the experiment in Definition 2. 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 $C = \mathsf{Enc}(PK, M_b)$ where $b \in \{0, 1\}$ is uniformly random. $\mathcal{A}$ succeeds in the experiment if he succeeds in guessing b. By assumption, $\mathsf{Adv}_{\mathcal{A},h}^{(0)}(n) = \mathsf{Adv}_{\mathcal{A},h}(n) = \delta$.

Experiment 1*:* In this experiment, the challenge ciphertext C 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 $r \leftarrow \mathbb{Z}_q$, 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. The distribution produced by Enc' is identical to the distribution of a real ciphertext; in particular, $\mathsf{Adv}_{\mathcal{A},h}^{(0)}(n) = \mathsf{Adv}_{\mathcal{A},h}^{(1)}(n)$.

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:

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

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 the DDH assumption and by Lemma 1, it immediately follows that the advantage of the adversary changes by at most a negligible amount.

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

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

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

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

for large enough n. By Theorem 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 simulates the view of the adversary $\mathcal{A}$ as follows. On input $(PK = (\mathbf{g}, y), h(\mathbf{g}, \mathbf{s}), \mathbf{r}, z \in \mathbb{Z}_q)$, 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 $\mathsf{Adv}_{\mathcal{A},h}(n) \leq \text{negl}(n)$.

5.2 Constructions Based on LWE

First, we present (a modification of) the GPV encryption scheme [12]. 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.

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 4.

$\mathsf{Enc}(PK, b)$, 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 $b' \in [\frac{1}{4}, \frac{3}{4}] \bmod 1$, and 0 otherwise.

Remark 3. There are two main differences between the cryptosystem in [12] and the variant described here. First, we choose the error parameter β to be superpolynomially small in n (and the modulus q to be superpolynomially large), whereas in [12], both are polynomially related to n. Second, the secret-key distribution in our case is the uniform distribution over $\{0, 1\}^m$, whereas in [12], 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 is not inherent; using a discrete Gaussian distribution also results in an identity-based encryption scheme (as in [12]) secure against auxiliary inputs. We defer the details to the full version of this paper.

Remark 4. 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 [17,22]. However, we note that another optimization proposed in [22] 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 now show that the (modified) GPV scheme is secure against subexponentially hard auxiliary inputs, under the decisional LWE assumption.

Theorem 3. *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})-$\texttt{AI-CPA}$ secure (when $\mathbf{A}$ is viewed as a public parameter).*

Remark. We can actually 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 $\mathrm{polylog}(m)$. 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 3). By Lemma 4 (Part 2) and because the length of "user-specific" public key $\mathbf{u}$ is $n \log q$ bits, to show Theorem 3 it suffices to show that our encryption scheme is $(q^n 2^{-m^\epsilon})$-$\texttt{wAI-CPA}$ 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 2. 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. 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

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

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 3 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. 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. 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,

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

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 1). By Theorem 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.

References

1. Akavia, A., Goldwasser, S., Vaikuntanathan, V.: Simultaneous hardcore bits and cryptography against memory attacks. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 474–495. Springer, Heidelberg (2009)
2. Alwen, J., Dodis, Y., Wichs, D.: Leakage-resilient public-key cryptography in the bounded-retrieval model. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 36–54. Springer, Heidelberg (2009)
3. Boneh, D., Halevi, S., Hamburg, M., Ostrovsky, R.: Circular-secure encryption from decision diffie-hellman. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 108–125. Springer, Heidelberg (2008)
4. Boyen, X.: Reusable cryptographic fuzzy extractors. In: ACM Conference on Computer and Communications Security, pp. 82–91 (2004)
5. Canetti, R., Dodis, Y., Halevi, S., Kushilevitz, E., Sahai, A.: Exposureresilient functions and all-or-nothing transforms. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 453–469. Springer, Heidelberg (2000)
6. Cramer, R., Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998)
7. Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002)
8. Dodis, Y., Kalai, Y.T., Lovett, S.: On cryptography with auxiliary input. In: STOC, pp. 621–630 (2009)
9. Dziembowski, S., Pietrzak, K.: Leakage-resilient cryptography. In: FOCS, pp. 293–302 (2008)
10. Faust, S., Kiltz, E., Pietrzak, K., Rothblum, G.: Leakage-resilient signatures (2009), http://eprint.iacr.org/2009/282
11. Feller, W.: An Introduction to Probability Theory and Its Applications, vol. 1. Wiley, Chichester (1968)
12. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC, pp. 197–206 (2008)

13. Goldreich, O., Rubinfeld, R., Sudan, M.: Learning polynomials with queries: The highly noisy case. SIAM J. Discrete Math. 13(4), 535–570 (2000)
14. Goldwasser, S., Kalai, Y.T., Rothblum, G.N.: One-time programs. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 39–56. Springer, Heidelberg (2008)
15. Ishai, Y., Sahai, A., Wagner, D.: Private circuits: Securing hardware against probing attacks. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 463–481. Springer, Heidelberg (2003)
16. Katz, J., Vaikuntanathan, V.: Signature schemes with bounded leakage. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 703–720. Springer, Heidelberg (2009)
17. Kawachi, A., Tanaka, K., Xagawa, K.: Multi-bit cryptosystems based on lattice problems. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 315–329. Springer, Heidelberg (2007)
18. Micali, S., Reyzin, L.: Physically observable cryptography (extended abstract). In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 278–296. Springer, Heidelberg (2004)
19. Naor, M., Reingold, O.: Number-theoretic constructions of efficient pseudo-random functions. J. ACM 51(2), 231–262 (2004)
20. Naor, M., Segev, G.: Public-key cryptosystems resilient to key leakage. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 18–35. Springer, Heidelberg (2009)
21. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: STOC, pp. 333–342 (2009)
22. Peikert, C., Vaikuntanathan, V., Waters, B.: A framework for efficient and composable oblivious transfer. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 554–571. Springer, Heidelberg (2008)
23. Petit, C., Standaert, F.-X., Pereira, O., Malkin, T., Yung, M.: A block cipher based pseudo random number generator secure against side-channel key recovery. In: ASIACCS, pp. 56–65 (2008)
24. Pietrzak, K.: A leakage-resilient mode of operation. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 462–482. Springer, Heidelberg (2009)
25. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC, pp. 84–93 (2005)