Public-Key Encryption in the Bounded-Retrieval Model

Joël Alwen[1], Yevgeniy Dodis[1,⋆], Moni Naor[2,⋆⋆], Gil Segev[2,⋆⋆⋆], Shabsi Walfish[3], and Daniel Wichs[1]

[1] New York University (NYU), New York, USA
{jalwen,dodis,wichs}@cs.nyu.edu
[2] Weizmann Institute of Science, Rehovot, Israel
{moni.naor,gil.segev}@weizmann.ac.il
[3] Google Inc. Mountain View, USA
shabsi@google.com

Abstract. We construct the *first* public-key encryption scheme in the *Bounded-Retrieval Model* (BRM), providing security against various forms of adversarial "key leakage" attacks. In this model, the adversary is allowed to learn arbitrary information about the decryption key, subject only to the constraint that the overall amount of "leakage" is bounded by at most ℓ bits. The goal of the BRM is to design cryptographic schemes that can flexibly tolerate arbitrarily leakage bounds ℓ (few bits or many Gigabytes), by *only* increasing the size of secret key proportionally, but keeping *all the other parameters* — including the size of the public key, ciphertext, encryption/decryption time, and the number of secret-key bits accessed during decryption — *small and independent of* ℓ.

As our main technical tool, we introduce the concept of an *Identity-Based Hash Proof System* (IB-HPS), which generalizes the notion of hash proof systems of Cramer and Shoup [CS02] to the identity-based setting. We give three different constructions of this primitive based on: (1) bilinear groups, (2) lattices, and (3) quadratic residuosity. As a result of independent interest, we show that an IB-HPS almost immediately yields an Identity-Based Encryption (IBE) scheme which is secure against (small) partial leakage of the target identity's decryption key. As our main result, we use IB-HPS to construct public-key encryption (and IBE) schemes in the Bounded-Retrieval Model.

1 Introduction

Traditionally, the security of cryptographic schemes has been analyzed in an idealized setting, where an adversary only sees the specified "input/output behavior" of a scheme, but has no other access to its internal secret state. Unfortunately, in the real world, an adversary may often learn some partial information

⋆ Research supported by NSF grants CNS-0831299 and CNS-0716690.
⋆⋆ Research supported in part by a grant from the Israel Science Foundation.
⋆⋆⋆ Research supported by the Adams Fellowship Program of the Israel Academy of Sciences and Humanities, and by a grant from the Israel Science Foundation.

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 113–134, 2010.

about secret state via various *key leakage* attacks. Such attacks come in a large variety and include *side-channel attacks*, where the physical realization of a cryptographic primitive can leak additional information, such as the computation-time, power-consumption, radiation/noise/heat emission etc. The cold-boot attack of Halderman et al. [HSH+08] is another example of a key-leakage attack, where an adversary can learn (imperfect) information about memory contents of a machine, even after the machine is powered down. Lastly, and especially relevant to this work, we will also consider key-leakage attacks where a remote adversary hacks into a target computer, or infects it with some malware, allowing her to download large amounts of secret-key information from the system. Schemes that are proven secure in an idealized setting, without key leakage, may become completely insecure if the adversary learns even a small amount of information about the secret key. Indeed, even very limited leakage attacks have been shown to have devastating consequences for the security of many natural schemes.

In this work, we study the design of leakage-resilient public-key encryption schemes, which are provably secure even in the presence of some limited key-leakage attacks. In particular, we will assume that the attacker can learn *any efficiently computable function of the secret key*, subject only to the constraint that the total amount of information learned (i.e. the output size of the leakage function) is bounded by ℓ bits, where ℓ is some arbitrary "leakage parameter" of the system. Clearly, at this level of generality, the secret-key size s must be strictly greater than the leakage-parameter ℓ. In the literature, there seems to be a distinction between two related models of leakage, which differ in how they treat the leakage-parameter ℓ in relation to the secret-key size s.

Relative-Leakage Model. In the model of *relative leakage*, firs studied by Akavia Goldwasser and Vaikuntanathan, [AGV09], the key-size s is chosen in the same way as in standard (non leakage-resilient) cryptographic schemes: it is based on a security parameter, and is usually made as *small* as possible (e.g. 1024 bits) to give the system some sufficient level of security. Once the key-size s is determined, the allowed leakage ℓ should be *relatively large in proportion to* s so that e.g. up to 50% of the key can be leaked without compromising security. Therefore, the relative-leakage model implicitly assumes that, no matter what the key-size is, a leakage attack can reveal at most some *relatively small fraction* of the key. This assumption is very reasonable for some attacks, such as the cold-boot attack, where all memory contents decay uniformly over time.

Bounded-Retrieval Model (BRM). The *Bounded-Retrieval Model (BRM)* [Dzi06, CLW06, ADW09] is a generalization of the relative-leakage model. In this model, the leakage-parameter ℓ is an arbitrary and independent parameter of the system, which is based on practical considerations about how much leakage the system needs to tolerate on an *absolute scale*. The secret-key size s is then chosen flexibly, depending on the security parameter *and* the leakage parameter ℓ, so as to simultaneously provide a sufficient level of security while allowing up to ℓ bits of leakage. Therefore, we can tolerate settings where the leakage ℓ might be small (several bits) or huge (several Gigabytes) by flexibly increasing the secret-key

size s depending on (and necessarily exceeding) the leakage parameter ℓ.[1] Of course, the key-size s should be as small as possible otherwise, so that the allowed leakage ℓ is a large *relative portion* of s as well.

With the additional flexibility in secret-key size, the BRM imposes an added efficiency requirement: the *public-key size, ciphertext size, encryption-time and decryption-time* must remain small, only depending on the security parameter, *and essentially independent of the leakage-parameter* ℓ. In other words, ℓ could potentially grow to the order of Gigabytes, and still result in a usable system, where the secret key is huge, but the public-key size, ciphertext size and encryption/decryption times are not much different from those of standard cryptosystems. This also means that the number of secret-key bits accessed during decryption (called *locality* from now on) must remain small and essentially independent of the flexibly growing secret-key size.

The flexibility of the BRM seems necessary to protect against large classes of key-leakage attacks. For example, if the key size is (only) proportional to the security parameter, several consecutive side-channel readings of a handful of bits might already leak the entire secret key. Therefore, for natural side-channel attacks (such as radiation/heat/noise emission) it might already make sense to make ℓ moderately large (say on the order of Megabytes) to get security. The main intention of the BRM in prior works, which we also focus on here, is to offer a novel method for protecting systems against hacking/malware attacks, where an adversary can download large amounts of information from an attacked system. It is clear that no security can be achieved using standard-sized (e.g. 1,024 bit) secret keys, as the adversary can download such keys in their entirety. However, it may be conceivable that the adversary still cannot download *too much* (e.g. many Gigabytes) worth of information because: (1) the bandwidth between the attacker and the system may be too slow to allows this, (2) the operating-system security may detect such large levels of leakage, or (3) such attacks would simply not be cost-effective. Therefore we can conceivably protect against such attacks by just making the leakage-parameter ℓ large enough (e.g. potentially many Gigabytes), and using a proportionally larger secret-key-size s. Having a large secret key may, by itself, not be a major concern due to the increasing size and affordability of local storage. On the other hand, it is crucial that the other efficiency measures of the system – ciphertext and public-key sizes, encryption and decryption times – must not degrade with the growth of ℓ.

1.1 Our Results

As our main contribution, we construct the first leakage-resilient Public-Key Encryption (PKE) scheme in the BRM. Along the way, we develop new notions and get results of independent interest. In particular, we:

- Develop a new notion of an Identity-Based Hash Proof System (IB-HPS), which naturally yields Identity-Based Encryption (IBE) schemes.

[1] Historically, the BRM setting envisioned ℓ as being necessarily huge. Here we take a more general view of the BRM, insisting only that the key size can be set flexibly based on the leakage ℓ.

- Give three constructions of IB-HPS based on the ideas behind three prior IBE schemes: [Gen06, BGH07, GPV08]. In particular, we show that the notion of IB-HPS unifies these seemingly unrelated constructions under a single framework. As a result, we get constructions of IB-HPS under (1) a bilinear Diffie-Hellman type assumption (2) the quadratic-residuosity assumption (3) the Learning With Errors (LWE) assumption. The first scheme is secure in the standard model, while the latter two rely on Random Oracles or, alternatively, non-standard interactive assumptions.
- Show that an IBE based on IB-HPS can easily be made leakage-resilient, in the relative-leakage model.
- Show how to use IB-HPS to construct public-key encryption (PKE) schemes in the BRM, allowing for arbitrary large leakage-bounds, while preserving efficiency. Our techniques also naturally extend to allow for the construction of IBE schemes in the BRM.
- Develop new information-theoretic tools to analyze our construction of PKE in the BRM. Namely, we define a new notion of *approximate* hash functions (where only elements that are far in Hamming distance are unlikely to collide) and generalize the Leftover-Hash Lemma to approximate hashing.
- Show how to achieve CCA security for our leakage-resilient IBE and PKE in BRM constructions.

Before describing our construction of PKE in the BRM, it is instructive to understand why this problem is non-trivial, and therefore we begin with some naïve approaches, which we improve in several steps.

Naïve Approach: Inflating the Security Parameter. As the first step of getting a PKE in the BRM, we would like to simply design a leakage-resilient PKE scheme that allows for arbitrarily large leakage-bounds ℓ, without necessarily meeting the additional efficiency requirements of the BRM. Luckily, there are several recent PKE schemes in the *relative-leakage model* [AGV09, NS09] where the leakage-bound $\ell(\lambda)$ is a large portion of the key-size $s(\lambda)$ which, in turn, depends on a security parameter λ. Therefore, one simple solution is to simply artificially inflate the security parameter λ sufficiently, until $s(\lambda)$ and, correspondingly, $\ell(\lambda)$ reach the desired level of leakage we would like to tolerate. Unfortunately, it is clear that this approach gets extremely inefficient very fast – e.g. to allow for Gigabytes worth of leakage, we may need to perform exponentiations on group elements with Gigabyte-long description sizes.

Better Approach: Leakage-Amplification via Parallel Repetition. As an improvement over the previous suggestion, we propose an alternative which we call *parallel-repetition*. Assume we have a leakage-resilient PKE scheme in the relative-leakage model, tolerating ℓ-bits of leakage, for some small ℓ. We can create a new "parallel-repetition scheme", by taking n independent copies of the above PKE with key-pairs $(\mathsf{pk}_1, \mathsf{sk}_1), \ldots, (\mathsf{pk}_n, \mathsf{sk}_n)$ and setting the secret-key of the new scheme to be $\overline{\mathsf{sk}} = (\mathsf{sk}_1, \ldots, \mathsf{sk}_n)$ and the public key to be $\overline{\mathsf{pk}} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$. To encrypt under the repetition scheme, a user would n-out-of-n secret-share the message m, and, encrypt each share m_i under the public key

pk_i. One may hope to argue that, if an adversary learns fewer than $n\ell$ bits about the secret-key $\overline{\mathsf{sk}}$ of the repetition scheme, then there is at least one secret key sk_i about which the adversary learns fewer than ℓ bits, thus maintaining security. Therefore, the hope is that parallel-repetition *amplifies leakage-resilience* from ℓ bits to $n\ell$ bits, and thus lets us meet any leakage-bound just by increasing n sufficiently. In terms of efficiency, the parallel-repetition approach will usually be more efficient than artificially inflating the security parameter, but it is still far from the requirements of the BRM: the public-key size, ciphertext size, and encryption/decryption times are all proportional to n, and therefore must grow as we strive to tolerate more and more leakage.

Security of Parallel-Repetition? Surprisingly, we do not know how to formalize the hope that parallel-repetition amplifies leakage-resilience generically via a reduction. Such a reduction would need to use an attacker that expects a public key and $n\ell$ bits of leakage on its secret key in the repetition scheme, to break the original scheme with ℓ bits of leakage. Unfortunately, it does not seem like there is any way to embed a challenge public key pk_i into $\overline{\mathsf{pk}}$, and faithfully simulate the output of an arbitrary leakage-function $f(\overline{\mathsf{sk}})$ with $n\ell$-bit output, by only learning $g(\mathsf{sk}_i)$ for some $g(\cdot)$ with ℓ bit output. In fact, as a subject of future work, we believe that there is a black-box separation showing that no such reduction can succeed *in general*. Luckily, we show that (a variant of) parallel-repetition amplifies leakage for schemes of a special form, which we will discuss later. For now, let us get back to the issue of efficiency, which we still need to resolve.

Improvement I: Improved Efficiency via Random Selection. To decrease ciphertext size and encryption/decryption times, the encryptor selects some random subset $\{r_1, \ldots, r_t\} \subseteq \{1 \ldots n\}$ of t indices, and targets the ciphertext to the corresponding public keys $\mathsf{pk}_{r_1}, \ldots, \mathsf{pk}_{r_t}$ (e.g. t-out-of-t secret-shares the message m and encrypts each share m_i under the public key pk_{r_i}). Intuitively, if an adversary learns much less than $n\ell$ bits of leakage about $\overline{\mathsf{sk}}$, then there should be *many* component-keys sk_i for which the adversary learns less than ℓ bits. Therefore the encryptor should select at least one index corresponding to such a key with large probability, when t is made proportional to the security parameter, and potentially much smaller than n. Although the ciphertext size and encryption/decryption times (and locality) are now only proportional to the security parameter, the size of the public key still grows with n, and so this scheme is still not appropriate for the BRM in terms of efficiency.

Improvement II: Small Public-Key Size via IBE. A natural solution to having a short public key is to use *identity-based encryption* (IBE) instead of standard PKE. This way, the public key of the repetition scheme is simply a short *master public key* of an IBE scheme, while the secret key $\overline{\mathsf{sk}} = (\mathsf{sk}_1, \ldots, \mathsf{sk}_n)$ consists of secret-keys for some fixed "identities" $\mathsf{ID}_1, \ldots, \mathsf{ID}_n$. Together, the above two improvements yield a scheme which meets the efficiency requirements of the BRM: the public-key size, ciphertext size, encryption/decryption times are now only proportional to the security parameter and independent of n, which can grow flexibly.

Security of the IBE-Based PKE in BRM Construction? In order to show that the resulting scheme, utilizing the two proposed improvements, is a PKE in the BRM we need to show the following. If we start with a leakage-resilient IBE that allows for ℓ-bits of leakage, then the construction amplifies this to any desired amount ℓ' just by increasing the number of secret keys n sufficiently. Unfortunately, it turns out that this is not the case in general and, in the full version of this work [ADN+09], we construct a counterexaple. That is, we can construct an artificial IBE scheme which is leakage-resilient in the relative leakage model, with leakage ℓ, but the above construction does not amplify leakage-resilience beyond $\ell' = \ell$, no matter how large n is. The problem is that, conceivably, after observing *all* n secret keys for n identities, it might be possible to come up with a very short "compressed" key (e.g. whose size is independent of n) which allows one to decrypt ciphertexts for *each one* of the given n identities. Our main result is to show that (a variant of) the construction is secure, if the leakage-resilient IBE has some additional underlying structure, which we call an Identity-Based Hash Proof System ($\mathsf{IB\text{-}HPS}$).

Hash Proof Systems and Identity-Based Hash Proof Systems. Recently, Naor and Segev [NS09] showed how to use a *hash proof system (HPS)* to construct leakage-resilient PKE in the relative-leakage model. Following, [KPSY09, NS09], we view an HPS as a *key-encapsulation mechanism (KEM)* with special structure.[2] A KEM consists of a key-generation procedure $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$, an encapsulation procedure $(c, k) \leftarrow \mathsf{Encap}(\mathsf{pk})$ which produces ciphertext/randomness pairs (c, k), and a decapsulation procedure $k{=}\mathsf{Decap}(c, \mathsf{sk})$, which uses the secret key sk to recover the randomness k from a ciphertext c. A KEM allows a sender that knows pk, to securely agree on randomness k with a receiver that possesses sk, by sending an encapsulation-ciphertext c. A *hash proof system* is a KEM with the following two properties:

- There exists an *invalid-encapsulation procedure* $c \leftarrow \mathsf{Encap}^*(\mathsf{pk})$, so that ciphertexts generated by $\mathsf{Encap}^*(\mathsf{pk})$ are computationally indistinguishable from those generated by $\mathsf{Encap}(\mathsf{pk})$, *even given the secret key* sk.
- For a fixed pk and *invalid ciphertext* c generated by $\mathsf{Encap}^*(\mathsf{pk})$, the output of $\mathsf{Decap}(c, \mathsf{sk})$ is *statistically* uniform, over the randomness of sk. This property can only hold if a fixed pk leaves statistical entropy in sk.

Notice the difference between valid and invalid ciphertexts. For a fixed pk, a *valid* c, produced by $(c, k) \leftarrow \mathsf{Encap}(\mathsf{pk})$, always decapsulated to the same value k, no matter which secret key sk is used to decapsulate it. On other hand, an invalid c produced by $c \leftarrow \mathsf{Encap}^*(\mathsf{pk})$, decapsulated to a statistically random value based on the randomness of sk.

[2] Our informal description and definition of HPS here, which will also be a basis of our formal definition of $\mathsf{IB\text{-}HPS}$ in Section 3.1, is a simplified version of the standard one. Although the two are *not* technically equivalent, the standard definition implies ours, which is in-turn sufficient for leakage-resilience and captures the main essence of HPS.

The above two properties are sufficient to prove KEM security, showing that for $(c, k) \leftarrow \mathsf{Encap}(\mathsf{pk})$, an attacker given c cannot distinguish k from uniform. The proof proceeds in two steps:

1. We replace the honestly generated $(c, k) \leftarrow \mathsf{Encap}(\mathsf{pk})$ with $c' \leftarrow \mathsf{Encap}^*(\mathsf{pk})$ and $k' \leftarrow \mathsf{Decap}(c', \mathsf{sk})$.
2. The value $k' = \mathsf{Decap}(c', \mathsf{sk})$ is statistically uniform over the choice of sk, which is unknown to the adversary.

As Naor and Segev noticed in [NS09], this proof also works in the presence of leakage since step (1) holds even if the adversary saw *all of* sk, and step (2) is information-theoretic, so we can argue that ℓ bits of leakage about sk will only reduce the statistical entropy of k' by at most ℓ bits. To agree on a uniform value k in the presence of leakage, we just compose the KEM with a randomness extractor.

The main benefit of this proof strategy is that, after switching valid/invalid ciphertexts in the first step, we can argue about leakage using a purely information-theoretic analysis. We observe that it is therefore relatively easy to show that (a variant of) parallel repetition amplifies leakage-resilience, since it amplifies the statistical entropy of the secret key $\overline{\mathsf{sk}} = (\mathsf{sk}_1, \ldots, \mathsf{sk}_n)$. In this work, we generalize the notion of HPS to the identity-based setting by defining Identity-Based Hash Proof System (IB-HPS) in a natural way. First of all, this gives us a general framework for constructing leakage-resilient IBE schemes in the relative-leakage model. Second of all, it also allows us to prove that a variant of the previously proposed leakage-amplification technique (using an IB-HPS rather than just any IBE) can indeed be used to get PKE (and IBE) schemes in the BRM.

1.2 Related Work

Restricted Models of Leakage-Resilience. Several other models of leakage-resilience have appeared in the literature. They differ from the model we described in the that they restrict the *type*, as well as *amount*, of information that the adversary can learn. For example, the work on *exposure resilient cryptography* [CDH+00, DSS01, KZ03] studies the case where an adversary can only learn some small *subset of the physical bits of the secret key*. Similarly, [ISW03] studies how to implement arbitrary computation in the setting where an adversary can observe a small *subset of the physical wires of a circuity*. Unfortunately, these models fail to capture many meaningful side-channel attacks, such as learning the hamming-weight of the bits or their parity.

In their seminal work, Micali and Reyzin [MR04] initiated the formal modeling of side-channel attacks under the axiom that *"only computation leaks information"*, where each invocation of a cryptographic primitive leaks a function of *only* the bits accessed during that invocation. Several primitives have been constructed in this setting including stream ciphers [DP08, Pie09] and signatures [FKPR10]. On the positive side, this model only imposes a bound on the amount of information learned during each invocation of a primitive, but not

on the overall amount of information that the attacker can get throughout the lifetime of the system. On the negative side, this model fails to capture many leakage-attacks, such as the cold-boot attack of [HSH+08], where *all* memory contents leak information, even if they were never accessed.

Certainly, all of the restricted models fail to capture hacking/malware attacks, where it is very conceivable that an attacker can compute *even complicated functions* of *all* information stored on the system.

Relative-Leakage Model. Several constructions of primitives in the relative-leakage model have appeared recently. The works of [AGV09, NS09] construct public-key encryption schemes in this model, and [KV09] constructs signatures. The works of [DKL09, DGK+10] considers a yet-stronger model of leakage-resilience, called the *auxiliary input model*, where the leakage-function need only be one-way (and not necessarily length-bounded), and constructs symmetric-key and public-key encryption in this model.

BRM. The Bounded-Retrieval Model was (concurrently) proposed by Di Crescenzo et al. [CLW06] and Dziembowski [Dzi06]. The name serves as an analogy to the Bounded Storage Model (BSM) of [Mau92], which restricts the amount of data that an adversary can *store after observing a huge public random string*, rather than the amount of data an adversary can *retrieve from a huge secret key*. With the exception of [ADW09], all of the work on the BRM is in the symmetric-key setting, where two parties share a huge secret key. The recent work of Alwen et al. [ADW09] gave the first public-key results in the BRM, by constructing identification schemes, (variants of) signatures, and authenticated-key-agreement protocols. However, these primitives cannot be used to encrypt a message non-interactively, as is done in the current work. Moreover, the authenticated-key agreement protocols of [ADW09] required the use of Random Oracles, while we offer (some) constructions in the standard model. We note that many of the prior schemes in the BRM and BSM employ ideas similar to the "parallel repetition" and "random-subset selection" that we described in the introduction. However, the proof-techniques in this paper differ significantly from previous works.

2 Preliminaries

Notation. For an integer n, we use the notation $[n]$ to denote the set $[n] \stackrel{\text{def}}{=} \{1, \ldots, n\}$. For a randomized function f, we write $f(x; r)$ to denote the unique output of f on input x with random coins r. We write $f(x)$ to denote a random variable for the output of $f(x; r)$, over the random coins r. For a set S, we let U_S denote the uniform distribution over S. For an integer $v \in \mathbb{N}$, we let U_v denote the uniform distribution over $\{0,1\}^v$, the bit-strings of length v. For a distribution or random variable X we write $x \leftarrow X$ to denote the operation of sampling a random x according to X. For a set S, we write $s \leftarrow S$ as shorthand for $s \leftarrow U_S$.

Entropy. The *min-entropy* of a r.v. X is $\mathbf{H}_\infty(X) \stackrel{\text{def}}{=} -\log(\max_x \Pr[X = x])$. This is a standard notion of entropy used in cryptography, since it measures the

worst-case predictability of X. The *average conditional min-entropy* [DORS08] of X given Z is defined by $\widetilde{\mathbf{H}}_\infty(X|Z) \stackrel{\text{def}}{=} -\log\left(\mathbb{E}_{z\leftarrow Z}\left[2^{-\mathbf{H}_\infty(X|Z=z)}\right]\right)$. This measures the worst-case predictability of X by an adversary that may observe a correlated variable Z.

Statistical Distance and Extractors. The *statistical distance* between X, Y is defined by $\mathbf{SD}(X,Y) = \frac{1}{2}\sum_x |\Pr[X=x] - \Pr[Y=x]|$. We write $X \approx_\varepsilon Y$ to denote $\mathbf{SD}(X,Y) \leq \varepsilon$, and $X \approx Y$ to denote that the statistical distance is negligible. An extractor [NZ96] can be used to extract uniform randomness out of a weakly-random value which is only assumed to have sufficient min-entropy. Our definition follows that of [DORS08], which is defined in terms of conditional min-entropy.

Definition 1 (Extractors). *We say that an efficient randomized function* $\mathsf{Ext} : \{0,1\}^u \to \{0,1\}^v$ *is an* (m,ε)-extractor *if for all* X, Z *such that* X *is distributed over* $\{0,1\}^u$ *and* $\widetilde{\mathbf{H}}_\infty(X|Z) \geq m$*, we get* $(Z, R, \mathsf{Ext}(X;R)) \approx_\varepsilon (Z, R, U_v)$ *where* R *is a random variable for the coins of* Ext.

Due to space constraints, almost all the proofs are omitted from the conference version of this paper. Please see the full version [ADN+09] for proofs and additional details.

3 Identity-Based Hash Proof System (IB-HPS)

3.1 Definition

An *Identity-Based Hash Proof System* (IB-HPS) consists of PPT algorithms: Setup, KeyGen, Encap, Encap^*, Decap. The algorithms have the following syntax.

$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ **:** The setup algorithm takes as input a security parameter λ and produces the *master public key* mpk and the *master secret key* msk. The master public key defines an *identity set* $\mathcal{ID}$, and an *encapsulated-key set* $\mathcal{K}$. All other algorithms $\mathsf{KeyGen}, \mathsf{Encap}, \mathsf{Decap}, \mathsf{Encap}^*$ implicitly include mpk as an input.

$\mathsf{sk}_{\mathsf{ID}} \leftarrow \mathsf{KeyGen}(\mathsf{ID}, \mathsf{msk})$ **:** For any identity $\mathsf{ID} \in \mathcal{ID}$, the KeyGen algorithm uses the master secret key msk to sample an identity secret key $\mathsf{sk}_{\mathsf{ID}}$.

$(c,k) \leftarrow \mathsf{Encap}(\mathsf{ID})$ **:** The *valid* encapsulation algorithm creates pairs (c,k) where c is a valid ciphertext, and $k \in \mathcal{K}$ is the encapsulated-key.

$c \leftarrow \mathsf{Encap}^*(\mathsf{ID})$ **:** The alternative *invalid* encapsulation algorithm which samples an invalid ciphertext c.

$k \leftarrow \mathsf{Decap}(c, \mathsf{sk}_{\mathsf{ID}})$ **:** The decapsulation algorithm is deterministic, and takes an identity secret key $\mathsf{sk}_{\mathsf{ID}}$ and a ciphertext c and outputs the encapsulated key k.

We require that an Identity-Based Hash Proof System satisfies the following properties.

I. Correctness of Decapsulation. For any values of $\mathsf{mpk}, \mathsf{msk}$ produced by $\mathsf{Setup}(1^\lambda)$, any $\mathsf{ID} \in \mathcal{ID}$ we have:

$$\Pr\left[k \neq k' \;\middle|\; \begin{array}{c} \mathsf{sk}_{\mathsf{ID}} \leftarrow \mathsf{KeyGen}(\mathsf{ID}, \mathsf{msk}) \\ (c,k) \leftarrow \mathsf{Encap}(\mathsf{ID}) \quad , \quad k' = \mathsf{Decap}(c, \mathsf{sk}_{\mathsf{ID}}) \end{array}\right] \leq \mathrm{negl}(\lambda)$$

II. Valid/Invalid Ciphertext Indistinguishability. The valid ciphertexts generated by Encap and the invalid ciphertexts generated by Encap^* should be indistinguishable *even given the identity secret key.* In particular, we define the following distinguishability game between an adversary $\mathcal{A}$ and a challenger.

$\mathsf{VI\text{-}IND}(\lambda)$

Setup: The challenger computes $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and gives mpk to the adversary $\mathcal{A}$.

Test Stage 1: The adversary $\mathcal{A}$ adaptively queries the challenger with $\mathsf{ID} \in \mathcal{ID}$ and the challenger responds with $\mathsf{sk}_{\mathsf{ID}}$.

Challenge Stage: The adversary selects an *arbitrary* challenge identity $\mathsf{ID}^* \in \mathcal{ID}$. The challenger chooses $b \leftarrow \{0,1\}$.
If $b = 0$ the challenger computes $(c, k) \leftarrow \mathsf{Encap}(\mathsf{ID}^*)$.
If $b = 1$ the challenger computes $c \leftarrow \mathsf{Encap}^*(\mathsf{ID}^*)$.
The challenger gives c to the adversary $\mathcal{A}$.

Test Stage 2: The adversary $\mathcal{A}$ adaptively queries the challenger with $\mathsf{ID} \in \mathcal{ID}$ and the challenger responds with $\mathsf{sk}_{\mathsf{ID}}$.

Output: The adversary $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$ which is the output of the game. We say that $\mathcal{A}$ *wins* the game if $b' = b$.

Note: In test stages 1,2 the challenger computes $\mathsf{sk}_{\mathsf{ID}} \leftarrow \mathsf{KeyGen}(\mathsf{ID}, \mathsf{msk})$ the first time that ID is queried and responds to all future queries on the same ID with the same $\mathsf{sk}_{\mathsf{ID}}$.

Note that, during the challenge phase, the adversary can choose *any* identity ID^*, and possibly even one for which it has seen the secret key $\mathsf{sk}_{\mathsf{ID}^*}$ in Test Stage 1 (or the adversary can simply get $\mathsf{sk}_{\mathsf{ID}^*}$ in Test Stage 2). We define the advantage of $\mathcal{A}$ in distinguishing valid/invalid ciphertexts to be $\mathsf{Adv}^{\mathsf{VI\text{-}IND}}_{\mathsf{IB\text{-}HPS},\mathcal{A}}(\lambda) \stackrel{\text{def}}{=} |\Pr[\mathcal{A} \text{ wins }] - \frac{1}{2}|$. We require that $\mathsf{Adv}^{\mathsf{VI\text{-}IND}}_{\mathsf{IB\text{-}HPS},\mathcal{A}}(\lambda) = \mathrm{negl}(\lambda)$.

III. Universality/Smoothness/Leakage-Smoothness. Other than properties I and II, we will need one additional information theoretic property. Essentially, we want to ensure that there are many possibilities for the decapsulation of an *invalid* ciphertext, which are left undetermined by the public parameters of the system. We define three flavors of this property as follows.

Definition 2 (Universal $\mathsf{IB\text{-}HPS}$). *We say that an* $\mathsf{IB\text{-}HPS}$ *is* (m, ρ)**-universal** *if, for any fixed values of* $\mathsf{mpk}, \mathsf{msk}$ *produced by* $\mathsf{Setup}(1^\lambda)$*, and any fixed* $\mathsf{ID} \in \mathcal{ID}$ *the following two properties hold:*

1. *Let* $\mathsf{SK} \leftarrow \mathsf{KeyGen}(\mathsf{ID}, \mathsf{msk})$ *be a random variable. Then* $\mathbf{H}_\infty(\mathsf{SK}) \geq m$.
2. *For any fixed distinct values* $\mathsf{sk}_{\mathsf{ID}} \neq \mathsf{sk}'_{\mathsf{ID}}$ *in the support of* SK*, we have*

$$\Pr_{c \leftarrow \mathsf{Encap}^*(\mathsf{ID})}[\mathsf{Decap}(c, \mathsf{sk}_{\mathsf{ID}}) = \mathsf{Decap}(c, \mathsf{sk}'_{\mathsf{ID}})] \leq \rho.$$

Notice the significant difference between valid and invalid ciphertexts. For valid ciphertexts c, the correctness of decapsulation ensures that there is a single value

$k \in \mathcal{K}$ such that $\mathsf{Decap}(c, \mathsf{sk}_{\mathsf{ID}}) = k$ for (virtually) all choices of $\mathsf{sk}_{\mathsf{ID}}$ (of which there are many by (1)). On the other hand, for invalid ciphertexts c, (2) ensures that it is highly unlikely that any two distinct secret-keys $\mathsf{sk}_{\mathsf{ID}}$ will decapsulate c to the same value k.

Definition 3 (Smooth/Leakage-Smooth IB-HPS). *We say that an* IB-HPS *is* **smooth** *if, for any* fixed values *of* $\mathsf{mpk}, \mathsf{msk}$ *produced by* $\mathsf{Setup}(1^\lambda)$*, any* $\mathsf{ID} \in \mathcal{ID}$*, we have:*

$$\mathbf{SD}(\ (c,k)\ ,\ (c,k')\ ) \leq \mathrm{negl}(\lambda)$$

where $c \leftarrow \mathsf{Encap}^*(\mathsf{ID})$*,* $k' \leftarrow U_{\mathcal{K}}$ *and* k *is sampled by choosing* $\mathsf{sk}_{\mathsf{ID}} \leftarrow \mathsf{KeyGen}(\mathsf{ID}, \mathsf{msk})$ *and computing* $k = \mathsf{Decap}(c, \mathsf{sk}_{\mathsf{ID}})$*. We say that an* IB-HPS *is* **ℓ-leakage-smooth** *if, for any (possibly randomized) function* $f(\cdot)$ *with* ℓ*-bit output, we have:*

$$\mathbf{SD}(\ (c, f(\mathsf{sk}_{\mathsf{ID}}), k)\ ,\ (c, f(\mathsf{sk}_{\mathsf{ID}}), k')\ ) \leq \mathrm{negl}(\lambda)$$

where $c, k, \mathsf{sk}_{\mathsf{ID}}, k'$ *are sampled as above. Note, for this property,* f *need not be efficient.*

3.2 Relations between Universality, Smoothness and Leakage-Smoothness

The following theorem is a simple consequence of the leftover hash lemma.

Theorem 1. *Assume that an* IB-HPS*, with key set* $\mathcal{K} = \{0,1\}^v$*, is* (m, ρ)*-universal. Then it is also* ℓ*-leakage smooth as long as* $\ell \leq m - v - \omega(\log(\lambda))$ *and* $\rho \leq \frac{1}{2^v}(1 + \mathrm{negl}(\lambda))$*.*

We also show how to convert a *smooth* IB-HPS $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encap}, \mathsf{Encap}^*, \mathsf{Decap})$ into a *leakage-smooth* IB-HPS using an extractor $\mathsf{Ext} : \mathcal{K} \to \{0,1\}^v$. We define:

- $\mathsf{Encap}_2(\mathsf{ID})$: Choose $(c,k) \leftarrow \mathsf{Encap}(\mathsf{ID}), k' \leftarrow \mathsf{Ext}(k; r)$ where r is a random seed. Output $c' = (c, r), k'$.
- $\mathsf{Encap}_2^*(\mathsf{ID})$: Choose a random seed r and $c \leftarrow \mathsf{Encap}^*(\mathsf{ID})$. Output $c' = (c, r)$.
- $\mathsf{Decap}_2(c', \mathsf{msk})$: Parse $c' = (c, r)$. Compute $k = \mathsf{Decap}(c, \mathsf{msk}), k' = \mathsf{Ext}(k; r)$. Output k'.

Theorem 2. *Assume that an* IB-HPS *is* smooth *and that* $|\mathcal{K}| = 2^m$*. Let* $\mathsf{Ext} : \mathcal{K} \to \{0,1\}^v$ *be an* $(m - \ell, \varepsilon)$*-extractor for some* $\varepsilon = \mathrm{negl}(\lambda)$*. Then the above transformation produces an* ℓ-leakage-smooth IB-HPS*.*

4 Constructions of IB-HPS

4.1 A Construction of IB-HPS Based on Bilinear Groups

Background: Let $\mathbb{G}, \mathbb{G}_T$ be two (multiplicative) groups of prime order p and let g be a generator of $\mathbb{G}$. Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a map from $\mathbb{G}$ to the *target group* $\mathbb{G}_T$. We say that the group $\mathbb{G}$ is *bilinear* if we have

1. **Bilinearity:** For all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$ we have $e(u^a, v^b) = e(u, v)^{ab}$.
2. **Non-degeneracy:** For the generator g of $\mathbb{G}$, we get $e(g, g) \neq 1$.
3. **Efficiency:** Operations (multiplication, exponentiation) in $\mathbb{G}, \mathbb{G}_T$ and the map e can be computed efficiently.

We assume the existence of a group-generation algorithm $\mathcal{G}(1^\lambda)$ which outputs a tuple $(\mathbb{G}, \mathbb{G}_T, g, e(\cdot,\cdot), p)$ where $\mathbb{G}$ is a bilinear group of prime order p.

We will rely on the *truncated augmented bilinear Diffie-Hellman exponent assumption* (q-TABDHE) from [Gen06]. We define the two distributions

$$D^{(0)}_{\lambda,q} = \left(g, g^\alpha, g^{(\alpha^2)}, \ldots, g^{(\alpha^q)}, g', g'^{(\alpha^{q+2})}, e\left(g^{(q+1)}, g'\right)\right)$$

and

$$D^{(1)}_{\lambda,q} = \left(g, g^\alpha, g^{(\alpha^2)}, \ldots, g^{(\alpha^q)}, g', g'^{(\alpha^{q+2})}, Z\right)$$

where $(\mathbb{G}, \mathbb{G}_T, g, e(\cdot,\cdot), p) \leftarrow \mathcal{G}(1^\lambda)$, $g' \leftarrow \mathbb{G}$, $\alpha \leftarrow \mathbb{Z}_p$, and $Z \leftarrow \mathbb{G}_T$. For any algorithm $\mathcal{B}$, the *distinguishing advantage of $\mathcal{B}$ in the q-*TABDHE *problem* is $\mathsf{Adv}^{\mathsf{TABDHE}}_{\mathcal{B}}(\lambda, q) \stackrel{\text{def}}{=} \left|\Pr\left[\mathcal{B}\left(D^{(0)}_{\lambda,q}\right) = 0\right] - \Pr\left[\mathcal{B}\left(D^{(1)}_{\lambda,q}\right) = 0\right]\right|$.

Definition 4. *We say that the q-*TABDHE *assumption holds if, for any PPT $\mathcal{B}$, $\mathsf{Adv}^{\mathsf{TABDHE}}_{\mathcal{B}}(\lambda, q) = \mathrm{negl}(\lambda)$. We say that the* TABDHE *assumption holds if q-*TABDHE *holds for all polynomial q.*

Construction: We now present the construction of IB-HPS which is based directly on Gentry's IBE [Gen06].

$\mathsf{Setup}(1^\lambda)$ **:** Let $(\mathbb{G}, \mathbb{G}_T, g, e, p) \leftarrow \mathcal{G}(1^\lambda)$. Let $h \leftarrow \mathbb{G}$, $\alpha \leftarrow \mathbb{Z}_p$ and $g_1 := g^\alpha$. Set $\mathsf{mpk} = (\mathbb{G}, \mathbb{G}_T, g, e, p, g_1, h)$ and set $\mathsf{msk} = \alpha$. The identity set is $\mathcal{ID} = \mathbb{Z}_p \setminus \{\alpha\}$ and the encapsulated-key set is $\mathcal{K} = \mathbb{G}_T$. [a]

$\mathsf{KeyGen}(\mathsf{ID}, \mathsf{msk})$ **:** For $\mathsf{ID} \in \mathcal{ID}$, choose $r_{\mathsf{ID}} \leftarrow \mathbb{Z}_p$ and compute $h_{\mathsf{ID}} = (hg^{-r_{\mathsf{ID}}})^{1/(\alpha - \mathsf{ID})}$. Output $\mathsf{sk}_{\mathsf{ID}} = (r_{\mathsf{ID}}, h_{\mathsf{ID}})$.

$\mathsf{Encap}(\mathsf{ID})$ **:** Choose random $s \in \mathbb{Z}_p$ and compute $u = g_1^s g^{-s\mathsf{ID}}$, $v = e(g, g)^s$ and output $c = (u, v), k = e(g, h)^s$.

$\mathsf{Encap}^*(\mathsf{ID})$ **:** Choose a random pair $(s, s') \in \mathbb{Z}_p$ subject to the constraint $s \neq s'$. Let $u = g_1^s g^{-s\mathsf{ID}}$, $v = e(g, g)^{s'}$ and output $c = (u, v)$.

$\mathsf{Decap}(c, \mathsf{sk}_{\mathsf{ID}})$ **:** Parse $c = (u, v)$ and output $k = e(u, h_{\mathsf{ID}})v^{r_{\mathsf{ID}}}$.

[a] The set $\mathcal{ID}$ is defined in terms of the secret α. Given $\mathsf{ID} \in \mathbb{Z}_p$, one can efficiently check if $\mathsf{ID} \in \mathcal{ID}$ by checking if $g^{\mathsf{ID}} \stackrel{?}{=} g_1$.

Essentially, various parts of Gentry's proof already show that the scheme satisfies the properties of IB-HPS. We provide a moularized proof of the following theorem in the full version [ADN+09].

Theorem 3. *Under the* TABDHE *assumption, the above construction is an* IB-HPS *which is simultaneously* smooth *and* (m, ρ)-universal *for $\rho = 0$ and $m = \log(p)$. More precisely, the valid/invalid ciphertext indistinguishability property holds under the q-*TABDHE *assumption for any adversary making at most q secret-key and leakage queries.*

4.2 Parameters of Three IB-HPS Constructions

In the full version of this work [ADN+09], we give two additional constructions of IB-HPS based on the recent IBE schemes of [BGH07, GPV08]. Here we, just give a short note on each construction and explain its parameters. We will be interested in the following:

1. The *actual identity-key size* $\hat{m}$: the number of bits needed to efficiently represent an identity secret key $\mathsf{sk}_{\mathsf{ID}}$.
2. The *encapsulated-key size* $v = \log(|\mathcal{K}|)$: the size of the encapsulated key.
3. The min-entropy m and the universality ρ: the values for which the scheme is (m, ρ)-universal.

An important parameter is the ratio $\frac{m}{\hat{m}}$, which determines the amount of *relative leakage* that our IBE and PKE in BRM constructions can handle. We note that *all* of the schemes satisfy the definition of *smoothness*.

A Scheme Based on Bilinear Groups. The parameters of our construction from the previous section, based on Gentry's IBE, are:

$$\hat{m} = 2\log(p) + O(1) \quad , \quad m = \log(p) \quad , \quad \frac{m}{\hat{m}} \approx \frac{1}{2} \quad , \quad v = \log(p) \quad , \quad \rho = 0.$$

where p is the (prime) order of an appropriate bilinear-group $\mathbb{G}$.

A Scheme Based on Quadratic Residuosity. We show that the IBE scheme of Boneh, Gentry and Hamburg [BGH07] contains a IB-HPS. The construction and proof essentially follow [BGH07] (with a minor modification in how identity secret keys are chosen, to get universality). The scheme is secure under the Quadratic Residuosity assumption in the Random Oracle model, or under a non-standard *interactive quadratic residuosity assumption* in the standard model. The parameters of interest are:

$$\hat{m} = \log(N) \quad , \quad m = 1 \quad , \quad \frac{m}{\hat{m}} = \frac{1}{\log(N)} \quad , \quad v = 1 \quad , \quad \rho = 0.$$

where N is an appropriately sized RSA modulus. Unfortunately, it is not clear how to make the scheme leakage-smooth for any $\ell > 0$, since the secret-key entropy m is too small to extract even a single bit. This problem can be fixed, as will be done in the BRM, by using parallel-repetition to amplify the entropy. Still, the relative leakage of the scheme will be poor because of the poor ratio of the entropy m to actual-key-size $\hat{m}$.

A Scheme Based on Lattices. We show how to get a construction of IB-HPS using the IBE scheme of Gentry, Peikert and Vaikuntanathan [GPV08]. Note that this IBE construction was already observed to be leakage-resilient by [AGV09], but this does not imply that it is an IB-HPS. In fact, we need to make some simple modifications so that the scheme satisfies our definition. The security of the scheme is based on a (decisional) Learning With Errors (LWE) assumption, in the random oracle model. Note that this assumption can be reduced to the GapSVP problem for lattices, using the techniques of [Reg05, Pei09].[3] We show

[3] We note that our construction requires that we use some (slightly) super-polynomial modulus q in the LWE problem, which means that we need to assume GapSVP is hard against some (slightly) super-polynomial time adversaries.

that, for any constant $\varepsilon > 0$, there exists some setting of the actual-key-size $\hat{m}$ so that:

$$m = (1-\varepsilon)\hat{m} \quad , \quad \frac{m}{\hat{m}} = (1-\varepsilon) \quad , \quad v = 1 \quad , \quad \rho = \frac{1}{2}(1+\mathrm{negl}(\lambda)).$$

Note that, by Theorem 2, this construction is therefore *already* ℓ-leakage smooth, for any $\ell \leq m - \omega(\log(\lambda))$, without any need to apply an extractor.

5 Leakage-Resilient IBE from IB-HPS

We define what it means for an Identity-Based Encryption (IBE) scheme to be resistant to key leakage attacks and show how to use an IB-HPS to construct such an IBE scheme. Our notion of leakage-resilience only allows leakage-attacks against the secret keys of the various identities, but *not* the master secret key. Also, we only allow the adversary to perform leakage attacks before seing the challenge ciphertext. As noted by [AGV09, NS09, ADW09], this limitation is inherent to (non-interactive) encryption schemes since otherwise the leakage function can simply decrypt the challenge ciphertext and output its first bit.

Definition. Recall an IBE scheme consists of four PPT algorithms Setup, KeyGen, Encrypt, and Decrypt. We omit discussion of the usual correctness requirements. We define the *semantic security game*, parameterized by a security parameter λ and a leakage parameter ℓ as the following game between an adversary $\mathcal{A}$ and a challenger.

$\mathsf{IBE\text{-}SS}(\lambda, \ell)$

Setup: Challenger computes $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$, gives mpk to the adv. $\mathcal{A}$.

Test Stage 1: The adv. $\mathcal{A}$ adaptively makes the following queries:

- **Secret-Key Queries:** On input $\mathsf{ID} \in \mathcal{ID}$, the challenger replies with $\mathsf{sk}_{\mathsf{ID}}$.
- **Leakage Queries:** On input $\mathsf{ID} \in \mathcal{ID}$, a PPT function $f : \{0,1\}^* \to \{0,1\}$, the challenger replies with $f(\mathsf{sk}_{\mathsf{ID}})$.

Challenge Stage: The adversary selects two messages $\mathsf{m}_0, \mathsf{m}_1 \in \mathcal{M}$ and a challenge identity $\mathsf{ID}^* \in \mathcal{ID}$ which *never appeared* in a secret-key query and appeared in *at most* ℓ leakage queries. The challenger chooses $b \leftarrow \{0,1\}$ uniformly at random and computes $c \leftarrow \mathsf{Encrypt}(\mathsf{ID}^*, \mathsf{m}_b)$ and gives c to the adversary $\mathcal{A}$.

Test Stage 2: The adversary gets to make *secret-key queries* for arbitrary $\mathsf{ID} \neq \mathsf{ID}^*$. The challenger replies with $\mathsf{sk}_{\mathsf{ID}}$.

Output: The adversary $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$. We say that the adversary *wins* the game if $b' = b$.

Note: In test stages 1,2 the challenger computes $\mathsf{sk}_{\mathsf{ID}} \leftarrow \mathsf{KeyGen}(\mathsf{ID}, \mathsf{msk})$ the first time that ID is queried (in a secret-key or leakage query) and responds to all future queries on the same ID with the same $\mathsf{sk}_{\mathsf{ID}}$.

The *advantage* of an adversary $\mathcal{A}$ in the *semantic security game with leakage* ℓ is $\mathsf{Adv}^{\mathsf{IBE\text{-}SS}}_{\mathsf{IBE},\mathcal{A}}(\lambda,\ell) \stackrel{\text{def}}{=} \left|\Pr[\mathcal{A} \text{ wins }] - \frac{1}{2}\right|$.

Definition 5 (Leakage-Resilient IBE). *An* IBE *scheme is* ℓ-leakage-resilient*, if the advantage of any any PPT adversary* $\mathcal{A}$ *in the semantic security game*

with leakage ℓ, is $\mathsf{Adv}^{\mathsf{IBE\text{-}SS}}_{\mathsf{IBE},\mathcal{A}}(\lambda,\ell) = \mathrm{negl}(\lambda)$. *We define the* relative leakage *of the scheme to be $\alpha \stackrel{def}{=} \ell/\hat{m}$, where $\hat{m}$ is the number of bits needed to efficiently store identity secret keys* $\mathsf{sk}_{\mathsf{ID}}$.

Construction: The construction of a leakage-resilient IBE from a leakage-smooth IB-HPS is almost immediate, by simply using the encapsulated key as a one-time-pad to encrypt a message. In particular, given an IB-HPS where the encapsulated key set $\mathcal{K}$ has some group structure $(\mathcal{K},+)$ (e.g. bit-strings with $\oplus$), we construct an IBE scheme with the same identity set $\mathcal{ID}$ and message set $\mathcal{M} = \mathcal{K}$. The $\mathsf{Setup}, \mathsf{KeyGen}$ algorithms are the same for both primitives and $\mathsf{Encrypt}, \mathsf{Decrypt}$ are defined by:

$\mathsf{Encrypt}(\mathsf{ID},\mathsf{m})$**:** Choose $(c_1,k) \leftarrow \mathsf{Encap}(\mathsf{ID})$ and let $c_2 = k + \mathsf{m}$.
Output $c = (c_1,c_2)$.
$\mathsf{Decrypt}(c,\mathsf{sk}_{\mathsf{ID}})$**:** For $c = (c_1,c_2)$, compute $k = \mathsf{Decap}(c_1,\mathsf{sk}_{\mathsf{ID}})$.
Output $\mathsf{m} = c_2 - k$.

Note that the Encap^* algorithm of the IB-HPS is not used in the construction, but will be used to argue security.

Theorem 4. *Assume that we start with an ℓ-leakage-smooth* IB-HPS. *Then the above construction yields an ℓ-leakage-resilient* IBE.

6 Leakage Amplification of IB-HPS

We now show how to construct an ℓ-leakage-smooth IB-HPS, for arbitrarily large values of ℓ, meeting the efficiency requirements of the BRM. This will be the main step towards building PKE (and IBE) schemes in the BRM. We start with a IB-HPS scheme $\Pi_1 = (\mathsf{Setup}, \mathsf{KeyGen}_1, \mathsf{Encap}_1, \mathsf{Encap}_1^*, \mathsf{Decap}_1)$ and compile it into a new IB-HPS scheme $\Pi_2 = (\mathsf{Setup}, \mathsf{KeyGen}_2, \mathsf{Encap}_2, \mathsf{Encap}_2^*, \mathsf{Decap}_2)$, where the identity secret keys can be made arbitrarily large, so as to achieve ℓ-leakage-smoothness for a large ℓ. We will assume there is a one-to-one function $H : \mathcal{ID}_2 \times [n] \to \mathcal{ID}_1$ where $\mathcal{ID}_1, \mathcal{ID}_2$ are the identity sets of Π_1, Π_2 respectively. In the constructed scheme, the identity secret key of each $\mathsf{ID} \in \mathcal{ID}_2$ consists of n components $\mathsf{sk}_{\mathsf{ID}} = (\mathsf{sk}_{\mathsf{ID}}[1], \ldots, \mathsf{sk}_{\mathsf{ID}}[n])$, where each component $\mathsf{sk}_{\mathsf{ID}}[i]$ is an independently sampled identity secret key for an identity $H(\mathsf{ID},i) \in \mathcal{ID}_1$ of the original scheme. Here, n will be a key-size parameter, which gives us flexibility in the size of the identity secret key in the constructed scheme, and will depend on the desired leakage-parameter ℓ. The encapsulation procedure $\mathsf{Encap}_2(\mathsf{ID})$ will target only a small subset of t-out-of-n of the identities $H(\mathsf{ID},i)$, and decapsulation Decap_2 will only need to read the values $\mathsf{sk}_{\mathsf{ID}}[i]$ associated with these t identities. Here t will be a *locality-parameter* which can be much smaller than (and independent of) n. A formal description of the construction appears in Figure 1. It is described abstractly in terms of arbitrary parameters n, t, v. In the theorem that follows, we show how to instantiate these appropriately based on the setting of ℓ, λ.

Let $\Pi_1 = (\mathsf{Setup}, \mathsf{KeyGen}_1, \mathsf{Encap}_1, \mathsf{Encap}_1^*, \mathsf{Decap}_1)$ be a IB-HPS with encapsulated-key-set $\mathcal{K}$ and identity-set $\mathcal{ID}_1$.
Let $n, t, v \in \mathbb{Z}^+$. We call n a *key-size parameter*, t a *locality parameter* and v a *output-size* parameter.
Let $H : \mathcal{ID}_2 \times [n] \to \mathcal{ID}_1$ be a one-to-one function for some set $\mathcal{ID}_2$.[a]
Let $\mathcal{G}$ be a $\frac{1}{2^v}$-universal hash function family of functions $g : \mathcal{K}^t \to \{0,1\}^v$.

Define $\Pi_2 = (\mathsf{Setup}, \mathsf{KeyGen}_2, \mathsf{Encap}_2, \mathsf{Encap}_2^*, \mathsf{Decap}_2)$ as follows:

$\mathsf{Setup}(1^\lambda)$**:** The setup procedure is the same as that of Π_1.
$\mathsf{KeyGen}_2(\mathsf{ID}, \mathsf{msk})$**:** For $i \in [n]$, sample $\mathsf{sk}_{\mathsf{ID}}[i] \leftarrow \mathsf{KeyGen}_1(H(\mathsf{ID}, i), \mathsf{msk})$. Output $\mathsf{sk}_{\mathsf{ID}} = (\mathsf{sk}_{\mathsf{ID}}[1], \ldots, \mathsf{sk}_{\mathsf{ID}}[n])$.
$\mathsf{Encap}_2(\mathsf{ID})$**:** Choose t random indices $\overline{r} = (r_1, \ldots, r_t) \leftarrow [n]^t$. Choose $g \leftarrow \mathcal{G}$. For $i \in \{1, \ldots, t\}$, compute: $(c_i, k_i) \leftarrow \mathsf{Encap}_1(H(\mathsf{ID}, r_i))$. Let $\overline{c} = (c_1, \ldots, c_t)$. Output: $C = (\overline{r}, \overline{c}, g)$, $k = g(k_1, \ldots, k_t)$.
$\mathsf{Encap}_2^*(\mathsf{ID})$**:** Choose t random indices $\overline{r} = (r_1, \ldots, r_t) \leftarrow [n]^t$. Choose $g \leftarrow \mathcal{G}$. For $i \in \{1, \ldots, t\}$, compute: $c_i \leftarrow \mathsf{Encap}_1^*(H(\mathsf{ID}, r_i))$. Let $\overline{c} = (c_1, \ldots, c_t)$. Output: $C = (\overline{r}, \overline{c}, g)$.
$\mathsf{Decap}_2(C, \mathsf{sk}_{\mathsf{ID}})$**:** Parse $C = (\overline{r}, \overline{c}, g)$. Compute $k_i = \mathsf{Decap}_1(c_i, \mathsf{sk}_{\mathsf{ID}}[r_i])$ for $i \in \{1, \ldots, t\}$. Output $k = g(k_1, \ldots, k_t)$.

[a] A collision-resistant hash function (CRHF) would suffice here as well.

Fig. 1. Leakage-Amplification of an IB-HPS: Construction of Π_2 from Π_1

For the analysis of the construction, we need to define a new parameter called the *effective key size* m'. This is the minimal value such that, for any fixed $\mathsf{mpk}, \mathsf{msk}, \mathsf{ID}$, the number of values that $\mathsf{sk}_{\mathsf{ID}} \leftarrow \mathsf{KeyGen}(\mathsf{ID})$ can take on is bounded by $2^{m'}$. If the actual key size is $\hat{m}$ and the key entropy is m, then $\hat{m} \geq m' \geq m$. Note that in all of our constructions, m/m' is a constant (even when $m/\hat{m}$ is not, as is the case for our QR-based construction).

Theorem 5. *Assume Π_1 is an (m, ρ)-universal* IB-HPS *with effective key size m', where $\rho < 1$ and $m/m' > 0$ are constants. Then, for any constant $\varepsilon > 0$ and any polynomial $v(\lambda)$, there exists some $t = O(v + \lambda)$ so that, for any polynomial $n(\lambda)$, the above construction of Π_2 with parameters (n, t, v) is an ℓ-leakage-smooth* IB-HPS *where $\ell(\lambda) = (1-\varepsilon)nm - v - \lambda$. The encapsulated-key-set of Π_2 is $\mathcal{K} = \{0,1\}^v$.*

The full proof of the above theorem appears in [ADN+09]. We give some intuition here. It is easy to see that Π_2 satisfies correctness. Also, the valid/invalid ciphertext indistinguishability property of Π_2 follows by a simple hybrid argument. Therefore, we only need to show ℓ-leakage smoothness, for the ℓ given by the theorem statement. For a fixed $\mathsf{mpk}, \mathsf{msk}, \mathsf{ID}$ in Π_2, the entropy of the random variable $\mathsf{SK}_{\mathsf{ID}} \sim \mathsf{KeyGen}_2(\mathsf{ID}, \mathsf{msk})$, is amplified to $\mathbf{H}_\infty(\mathsf{SK}_{\mathsf{ID}}) \geq nm$, since it consists of n independently sampled secret keys of Π_1. If we could show that the scheme is also ρ'-universal, for some small $\rho' \leq (\frac{1}{2^v} + \mathrm{negl}(\lambda))$, then we could rely on Theorem 1 to show leakage-smoothness. Unfortunately, this is not the case. The problem is that, if two values $\mathsf{sk}_{\mathsf{ID}} \neq \mathsf{sk}'_{\mathsf{ID}}$ in the constructed scheme differ

in only one position j, then $\mathsf{Decap}_2(C, \mathsf{sk}_{\mathsf{ID}}) = \mathsf{Decap}(C, \mathsf{sk}'_{\mathsf{ID}})$ as long as the ciphertext C does not "select" j, which happens with large probability. Therefore, to analyze the leakage smoothness of the construction, we define a new notion called *approximately universal hashing*, where we only insist that values which are far from each other in Hamming distance (over some alphabet) are unlikely to collide. We then show a variant of the leftover-hash lemma, called the *approximate leftover-hash lemma* holds for approximate hashing. Lastly, we show that the decapsulation procedure $\mathsf{Decap}_2(C, \mathsf{sk}_{\mathsf{ID}})$ of the amplified scheme Π_2 is approximately universal, for appropriate parameters, when $C \leftarrow \mathsf{Encap}^*(\mathsf{ID})$.[4] Combining these results, we get the parameters of the theorem.

7 Public-Key Encryption and IBE in the BRM

A public-key encryption (PKE) scheme in the BRM consists of the algorithms (KeyGen, Encrypt, Decrypt), which are all parameterized by a security parameter λ *and a leakage parameter* ℓ. The syntax and the correctness property of an encryption scheme follow the standard notion of public-key encryption. We define the following *semantic-security game with leakage* ℓ between an adversary $\mathcal{A}$ and a challenger.

$\mathsf{SemS}(\lambda, \ell)$

Key Generation: The challenger computes $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda, 1^\ell)$ and gives pk to the adversary $\mathcal{A}$.

Leakage: The adversary $\mathcal{A}$ selects a PPT function $f : \{0,1\}^* \to \{0,1\}^\ell$ and gets $f(\mathsf{sk})$ from the challenger.

Challenge: The adversary $\mathcal{A}$ selects two messages $\mathsf{m}_0, \mathsf{m}_1$. The challenger chooses $b \leftarrow \{0,1\}$ uniformly at random and gives $c \leftarrow \mathsf{Encrypt}(\mathsf{m}_b, \mathsf{pk})$ to the adversary $\mathcal{A}$.

Output: The adversary $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$. We say that $\mathcal{A}$ wins the game if $b' = b$.

For any adversary $\mathcal{A}$, the *advantage of* $\mathcal{A}$ in the above game is defined as $\mathsf{Adv}^{\mathsf{SemS}}_{\mathsf{PKE},\mathcal{A}}(\lambda, \ell) \stackrel{\text{def}}{=} \left|\Pr[\mathcal{A} \text{ wins }] - \frac{1}{2}\right|$.

Definition 6 (Leakage-Resilient PKE). *A public-key encryption scheme* PKE *is leakage-resilient, if for any polynomial* $\ell(\lambda)$ *and any PPT adversary* $\mathcal{A}$*, we have* $\mathsf{Adv}^{\mathsf{SemS}}_{\mathsf{PKE},\mathcal{A}}(\lambda, \ell(\lambda)) = \mathrm{negl}(\lambda)$.

Definition 7 (PKE in the BRM). *We say that a leakage-resilient PKE scheme is a* PKE in the BRM*, if the public-key size, ciphertext size, encryption-time and decryption-time (and the number of secret-key bits read by decryption) are independent of the leakage-bound* ℓ*. More formally,* **there exist** *polynomials* $\mathsf{pksize}, \mathsf{ctsize}, \mathsf{encT}, \mathsf{decT}$*, such that,* **for any** *polynomial* ℓ *and any* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda, 1^{\ell(\lambda)})$*,* $\mathsf{m} \in \mathcal{M}$*,* $c \leftarrow \mathsf{Encrypt}(\mathsf{m}, \mathsf{pk})$*, the scheme satisfies:*

[4] For approximate universality, we think of the "big key" $\mathsf{sk}_{\mathsf{ID}}$ as consisting of n alphabet symbols, with one symbol for each component key $\mathsf{sk}_{\mathsf{ID}}[i]$.

1. *Public-key size is* $|\mathsf{pk}| \leq O(\mathsf{pksize}(\lambda))$*, ciphertext size is* $|c| \leq O(\mathsf{ctsize}(\lambda, |\mathsf{m}|))$.
2. *Run-time of* $\mathsf{Encrypt}(\mathsf{m}, \mathsf{pk})$ *is* $\leq O(\mathsf{encT}(\lambda, |\mathsf{m}|))$.
3. *Run-time of* $\mathsf{Decrypt}(c, \mathsf{sk})$*, and the number of bits of* sk *accessed, is* $\leq O(\mathsf{decT}(\lambda, |\mathsf{m}|))$.

The relative-leakage *of the scheme is* $\alpha \stackrel{def}{=} \ell/|\mathsf{sk}|$.

We can generalize the above definition to IBE schemes. A leakage-resilient IBE is an *IBE in the BRM* if the master-public-key size, master-secret-key size, ciphertext size and encryption/decryption times are bounded by polynomials independent of ℓ.

Theorem 6 (PKE and IBE in BRM). *Assume that we have an* (m, ρ)*-universal* IB-HPS *satisfying the conditions of Theorem 5 and having actual key size* $\hat{m}$*. Then, for any constant* $\varepsilon > 0$ *and any polynomial* v*, we get PKE (resp. IBE) schemes in the BRM with message space* $\mathcal{M} = \{0,1\}^v$ *and:*

1. *Public-key size (resp. master public/secret key size) is the same as that of the underlying* IB-HPS.
2. *The locality-parameter is* $t = O(v + \lambda)$*. The # of secret-key bits accessed during decryption is* $t\hat{m}$.
3. *Ciphertext-size/encryption-time/decryption-time differ by a factor of* t *from those of the underlying* IB-HPS.
4. *Relative leakage is* $\alpha \geq \frac{m}{\hat{m}}(1-\varepsilon)$*, for sufficiently large values of the leakage-parameter* ℓ*. In particular, for large enough* ℓ*, the secret-key size (resp. identity-secret-key size) is* $\leq \frac{\hat{m}}{m}(1+\varepsilon)\ell$.

Proof. Follows directly from leakage-amplification (Theorem 5). For any leakage-parameter ℓ, the key-size parameter n in the construction of Π_2 in Figure 1 is made just large enough so that $\ell \leq (1-\varepsilon)nm - v - \lambda$. Therefore, Π_2 is ℓ-leakage smooth. By Theorem 4, this yields an ℓ-leakage resilient IBE. The efficiency parameters are obvious from the construction, so it is easy to see that we get an IBE in the BRM. By ignoring all identities except for a single one, we naturally get a PKE in the BRM. The relative leakage is $\alpha = \frac{\ell}{\hat{m}n} \approx \frac{m}{\hat{m}}(1-\varepsilon)$, for ℓ large enough in relation to v, λ. □

8 Extensions

In the full version [ADN+09] of this work, we show several extensions to the results from the previous section. We describe them briefly here.

CCA Security. We show that the main ideas underlying our approach can be extended to deal with chosen-ciphertext attacks. We present constructions of encryption schemes that are resilient to leakage even under chosen-ciphertext attacks. That is, these schemes are semantically secure even against an adversary that is allowed to submit both leakage queries and decryption queries. We first consider identity-based encryption, and show that the CCA-secure variant

of Gentry's scheme [Gen06] can be generalized to deal with leakage. We then consider public-key encryption in the BRM, and observe that the generic transformation from chosen-plaintext security to chosen-ciphertext security, using the Naor-Yung paradigm [NY90], also applies in the BRM.

Shorter Ciphertexts via Anonymous Encapsulation. We notice that two of our IB-HPS constructions, based on lattices and quadratic residuosity, have additional structure, which allows for a more efficient version of our leakage-amplification construction. In the construction shown in Figure 1, the ciphertext C of the constructed scheme Π_2 contains t ciphertexts $c_1, \ldots, c_t$ of the underlying scheme Π_1, where $t = O(\lambda + v)$. We show how to reduce this to a single ciphertext if we start with an IB-HPS construction Π_1 that has an additional property, which we call *anonymous encapsulation*. Such a scheme has two additional procedures:

- $(c, s) \leftarrow \mathsf{EncapC}()$, which samples a ciphertext c together with a trapdoor s *without* knowing the target ID.
- $k = \mathsf{EcnapK}(c, s, \mathsf{ID})$, which (deterministically) computes k for any ID, given c and a trapdoor s.

Note that the procedures $\mathsf{EncapC}, \mathsf{EcnapK}$ (like Encap) are implicitly parameterized by the master public key mpk.

Definition 8 (Anonymous Encapsulation). *An* IB-HPS *has* anonymous encapsulation *if there exist efficient procedures* $\mathsf{EncapC}, \mathsf{EcnapK}$ *as above, such that, for any fixed* $\mathsf{mpk}, \mathsf{msk}, \mathsf{ID}$*, sampling* $(c, k) \leftarrow \mathsf{Encap}(\mathsf{ID})$ *is equivalent to sampling* $(c, s) \leftarrow \mathsf{EncapC}()$ *and computing* $k = \mathsf{EcnapK}(c, s, \mathsf{ID})$.

For the lattice and quadratic-residuosity based constructions, the procedures EncapC, EcnapK are already implicitly defined by Encap, which first samples c anonymously (independently of ID) and then computes k for a given ID using the randomness s that was used to generate c.

There are several advantages to IB-HPS schemes that have the anonymous-encapsulation property. Firstly, it's easy to see that the IBE constructed from such schemes has *anonymity*, in that the ciphertext does not reveal the target identity. Perhaps more importantly, anonymous encapsulation can be used to get an improved leakage-amplification scheme with shorter ciphertexts.[5] In particular, we modify the procedure $\mathsf{Encap}_2(\mathsf{ID})$ of the constructed Π_2 scheme, so that it samples a *single* ciphertext/trapdoor pair $(c, s) \leftarrow \mathsf{EncapC}_1()$ of the underlying scheme Π_1, and computes $k_i = \mathsf{EcnapK}_1(c, s, H(\mathsf{ID}, r_i))$ for each of of the t random indices $r_i \in [n]$. The ciphertexts of the constructed scheme therefore consist of $C = (\overline{r}, c, g)$, and contain only a single ciphertext c of the underlying scheme. To reduce the ciphertext size still further, we can employ the following optimizations:

1. Instead of sampling the indices $\overline{r} \leftarrow [n]^t$ uniformly at random, and communicating this choice in the ciphertext, we use use a *hitting sampler* to

[5] A similar technique is implicitly used to get shorter ciphertexts relative to the message length in the IBE constructions of [BGH07, GPV08].

sample $\overline{r} \in [n]^t$ efficiently. This choice can then be communicated using a seed of description size $\log(n) + O(\lambda + v)$, rather than the previous size $t \log(n) = O((\lambda + v) \log(n))$ needed to communicate $\overline{r}$ explicitly.

2. Use a γ-universal, instead of fully universal, hash function g, where $\gamma = \frac{1}{2^v}(1 + \mathrm{negl}(\lambda))$. As observed in [SZ99], such hash functions can have description sizes $O(v + \lambda)$, only proportional to the output size, and not the somewhat larger input size.

We show that leakage-amplification still holds for the modified constructions, by showing that $\mathsf{Decap}_2(C, \cdot)$ is an approximately-universal hash function with appropriate parameters, when $C \leftarrow \mathsf{Encap}^*(\mathsf{ID})$. Unfortunately, the setting of the parameters requires that $\rho \leq \frac{1}{2^v}$ in the original scheme, which is only the case for our QR-based scheme but *not* the lattice-based scheme.

9 Comparison of PKE (and IBE) in BRM Constructions

In Table 1, we compare the efficiency and relative-leakage of our various IBE and PKE in BRM constructions. We assume that the plaintext size is $v = O(\lambda)$.[6] In all of the schemes, the leakage-parameter ℓ can be arbitrarily large and the relative leakage column indicates the ratio of leakage to secret-key size. The public-key size of all schemes is the same as the master-public-key size of the corresponding IB-HPS and the encryption/decryption times (and the number of bits accessed) differ by a multiplicative factor of $t = O(\lambda)$ from those of the underlying IB-HPS. The "CT expansion" column indicates the ratio of the ciphertext size in the BRM to that of the underlying IB-HPS. The "CT size in BRM" column measures the size of the ciphertext in the BRM on an absolute scale.[7] The value $\varepsilon > 0$ can be an arbitrary constant.

Table 1. Comparison of Our PKE in BRM Constructions

Scheme	Assumption	Relative Leakage	CT Size in BRM	CT Expansion
Bilinear-Groups [Gen06]	TABDHE	$(\frac{1}{2} - \varepsilon)$	$O(\lambda^2)$	$O(\lambda)$
Quadratic Residuosity [BGH07]	QR †	$\frac{1}{O(\lambda)}$	$O(\lambda)$	$O(1)$
Lattices [GPV08]	LWE/GapSVP †	$(1 - \varepsilon)$	$O(\lambda^4)$	$O(\lambda)$

† = Random Oracle Model/Interactive Assumption

[6] To encrypt larger messages, it is sufficient to encrypt a short $O(\lambda)$ sized key for a symmetric-key encryption scheme.

[7] Note that, to make a fair comparison, we assume that RSA moduli and bilinear-group elements have description sizes $O(\lambda)$. For our LWE based construction, the modulus q needs to be (slightly) super-polynomial, and we are pessimistic by just bounding its description size by $O(\lambda)$.

Acknowledgements

We would like to thank Vinod Vaikuntanathan for many enlightening discussions, and especially for his invaluable help in answering our technical questions about his recent lattice-related results. We would also like to thank Craig Gentry for his helpful discussion and for pointing us to the IBE scheme of [BGH07].

References

[ADN+09] Alwen, J., Dodis, Y., Naor, M., Segev, G., Walfish, S., Wichs, D.: Public-key encryption in the bounded-retrieval model. Cryptology ePrint Archive, Report 2009/512 (2009), http://eprint.iacr.org/

[ADW09] 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)

[AGV09] 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)

[BGH07] Boneh, D., Gentry, C., Hamburg, M.: Space-efficient identity based encryption without pairings. In: FOCS, pp. 647–657 (2007)

[CDH+00] Canetti, R., Dodis, Y., Halevi, S., Kushilevitz, E., Sahai, A.: Exposure-resilient functions and all-or-nothing transforms. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 453–469. Springer, Heidelberg (2000)

[CLW06] Di Crescenzo, G., Lipton, R.J., Walfish, S.: Perfectly secure password protocols in the bounded retrieval model. In: Halevi and Rabin [HR06], pp. 225–244 (2006)

[CS02] 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)

[DGK+10] Dodis, Y., Goldwasser, S., Kalai, Y.T., Peikert, C., Vaikuntanathan, V.: Public-key encryption schemes with auxiliary inputs. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 361–381. Springer, Heidelberg (2010)

[DKL09] Dodis, Y., Kalai, Y.T., Lovett, S.: On cryptography with auxiliary input. In: STOC (2009)

[DORS08] Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. SIAM J. Comput. 38(1), 97–139 (2008)

[DP08] Dziembowski, S., Pietrzak, K.: Leakage-resilient cryptography. In: FOCS, pp. 293–302. IEEE Computer Society, Los Alamitos (2008)

[DSS01] Dodis, Y., Sahai, A., Smith, A.: On perfect and adaptive security in exposure-resilient cryptography. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 301–324. Springer, Heidelberg (2001)

[Dzi06] Dziembowski, S.: Intrusion-resilience via the bounded-storage model. In: Halevi and Rabin [HR06], pp. 207–224 (2006)

[FKPR10] Faust, S., Kiltz, E., Pietrzak, K., Rothblum, G.N.: Leakage-resilient signatures. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 343–360. Springer, Heidelberg (2010)

[Gen06] Gentry, C.: Practical identity-based encryption without random oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006)

[GPV08] Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC 2008: Proceedings of the 40th annual ACM symposium on Theory of computing, pp. 197–206. ACM, New York (2008)

[HR06] Halevi, S., Rabin, T. (eds.): TCC 2006. LNCS, vol. 3876. Springer, Heidelberg (2006)

[HSH+08] Alex Halderman, J., Schoen, S.D., Heninger, N., Clarkson, W., Paul, W., Calandrino, J.A., Feldman, A.J., Appelbaum, J., Felten, E.W.: Lest we remember: Cold boot attacks on encryption keys. In: van Oorschot, P.C. (ed.) USENIX Security Symposium. USENIX Association, pp. 45–60 (2008)

[ISW03] 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)

[KPSY09] Kiltz, E., Pietrzak, K., Stam, M., Yung, M.: A new randomness extraction paradigm for hybrid encryption. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 590–609. Springer, Heidelberg (2009)

[KV09] Katz, J., Vaikuntanathan, V.: Signature schemes with bounded leakage resilience. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 703–720. Springer, Heidelberg (2009)

[KZ03] Kamp, J., Zuckerman, D.: Deterministic extractors for bit-fixing sources and exposure-resilient cryptography. In: FOCS, pp. 92–101 (2003)

[Mau92] Maurer, U.M.: Conditionally-perfect secrecy and a provably-secure randomized cipher. J. Cryptology 5(1), 53–66 (1992)

[MR04] Micali, S., Reyzin, L.: Physically observable cryptography (extended abstract). In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 278–296. Springer, Heidelberg (2004)

[NS09] 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)

[NY90] Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, pp. 427–437 (1990)

[NZ96] Nisan, N., Zuckerman, D.: Randomness is linear in space. J. Comput. Syst. Sci. 52(1), 43–52 (1996)

[Pei09] Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: STOC, pp. 333–342 (2009)

[Pie09] Pietrzak, K.: A leakage-resilient mode of operation. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 462–482. Springer, Heidelberg (2009)

[Reg05] Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC, pp. 84–93 (2005)

[SZ99] Srinivasan, A., Zuckerman, D.: Computing with very weak random sources. SIAM J. Comput. 28(4), 1433–1459 (1999)