Public-Key Encryption in the Bounded-Retrieval Model

Joël Alwen [*] Yevgeniy Dodis[†] Moni Naor [‡] Gil Segev [§] Shabsi Walfish [¶] Daniel Wichs[‖]

October 28, 2009

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 about secret state via various *key leakage* attacks. Such attacks come in a large variety and include *side-channel attacks* [Koc96, BDL97, BS97, KJJ99, QS01, GMO01], 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.

Unfortunately, it is unrealistic to assume that we can foresee, let alone block, all of the possible means through which key leakage can occur in real-world implementations of cryptographic schemes. Therefore, the cryptographic

[*]Department of Computer Science, NYU. Email: jalwen@cs.nyu.edu.

[†]Department of Computer Science, NYU. Email: dodis@cs.nyu.edu.

[‡]Incumbent of the Judith Kleeman Professorial Chair, Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot 76100, Israel. Email: moni.naor@weizmann.ac.il. Research supported in part by a grant from the Israel Science Foundation.

[§]Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot 76100, Israel. Email: gil.segev@weizmann.ac.il. Research supported by the Adams Fellowship Program of the Israel Academy of Sciences and Humanities, and by a grant from the Israel Science Foundation.

[¶]Google Inc. Email: shabsi@google.com.

[‖]Department of Computer Science, NYU. Email: wichs@cs.nyu.edu.

community has recently initiated the investigation of increasingly general (formally modeled) classes of leakage attacks, with the aim of constructing *leakage-resilient* cryptographic schemes that remain provably secure even in the presence of such attacks. Of course, if an adversary can get unrestricted information about the secret key (say, of an encryption scheme), then she can learn the key in its entirety and the security of the system is necessarily compromised. Therefore, we must first place some "upper bound" on the type or amount of information that the adversary can learn. The nature of such bounds varies in the literature, as we survey later. For this work, we only restrict the *amount*, but not the *type*, of information that an adversary can learn through a key-leakage attack. 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* [AGV09, NS09, DKL09, KV09], 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, CDD+07, DP07, 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

[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.

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.
- 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 Appendix A, 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 (IB-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 IB-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 Section 3, we generalize the notion of HPS to the identity-based setting by defining Identity-Based Hash Proof System (IB-HPS) in a natural way. Then, in Section 4, we show how to construct leakage-resilient IBE in the relative-leakage model using IB-HPS. In Section 5, we show that a variant of our parallel-repetition idea, and random-subset selection ideas, indeed amplify leakage-resilience of IB-HPS-based constructions. Finally, in Section 6, show how this leads to constructions of 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 [MR04a] 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 [FKPR09]. 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 work of [DKL09] 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 encryption in this model.

BRM. The Bounded-Retrieval Model was (concurrently) proposed by Di Crescenzo et. al [CLW06] and Dziembowski [Dzi06], and later studied by [CDD$^+$07, DP07, ADW09]. The name serves as an analogy to the Bounded Storage Model (BSM) of [Mau92, AR99, ADR02, Lu02, Vad04], 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 random variable 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. We also review a generalization from [DORS08], called *average conditional min-entropy* defined by

$$\widetilde{\mathbf{H}}_\infty(X|Z) \stackrel{\text{def}}{=} -\log\left(\mathbb{E}_{z \leftarrow Z}\left[\ \max_x \Pr[X = x|Z = z]\ \right]\right) = -\log\left(\mathbb{E}_{z \leftarrow Z}\left[2^{-\mathbf{H}_\infty(X|Z=z)}\right]\right).$$

This measures the worst-case predictability of X by an adversary that may observe a correlated variable Z. We will use the following lemmas to reason about entropy.

Lemma 2.1 ([DORS08]) *Let X, Y, Z be random variables where Z takes on values in a set of size at most 2^ℓ. Then $\widetilde{\mathbf{H}}_\infty(X|(Y,Z)) \geq \widetilde{\mathbf{H}}_\infty((X,Y)|Z) - \ell \geq \widetilde{\mathbf{H}}_\infty(X|Z) - \ell$ and, in particular, $\widetilde{\mathbf{H}}_\infty(X|Y) \geq \mathbf{H}_\infty(X) - \ell$.*

STATISTICAL DISTANCE AND EXTRACTORS. The *statistical distance* between two random variables 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 2.1 (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.*

We now recall the definition of universal-hashing and the leftover-hash lemma, which states that universal hash functions are also good extractors.

Definition 2.2 (ρ-Universal Hashing) *A family $\mathcal{H}$, consisting of (deterministic) functions $h : \{0,1\}^u \to \{0,1\}^v$, is a ρ-*universal hash family *if for any $m_1 \neq m_2 \in \{0,1\}^u$ we have $\Pr_{h \leftarrow \mathcal{H}}[h(m_1) = h(m_1)] \leq \rho$.*

Lemma 2.2 (Leftover-Hash Lemma [NZ96]) *Assume that the family $\mathcal{H}$ of functions $h : \{0,1\}^u \to \{0,1\}^v$, is a ρ-universal hash family. Then the randomized extractor $\mathsf{Ext}(x; h) = h(x)$, where h is uniform over $\mathcal{H}$, is an (m, ε)-extractor as long as $m \geq v + 2\log(1/\varepsilon) - 1$ and $\rho \leq \frac{1}{2^v}(1 + \varepsilon^2)$.*

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

3.1 Definition

An *Identity-Based Hash Proof System* (IB-HPS) consists of PPT algorithms: $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encap}, \mathsf{Encap}^*, \mathsf{Decap})$. 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:

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

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 3.1 (Universal IB-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* SK *be a random variable for the output of* $\mathsf{KeyGen}(\mathsf{ID}, \mathsf{msk})$*. 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 3.2 (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.

We show two simple observations about the relationships between universality, smoothness and leakage-smoothness. First, we show that a universal IB-HPS is leakage smooth for appropriate parameters.

Theorem 3.1 *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* $\rho \leq \frac{1}{2^v}\,(1 + \mathrm{negl}(\lambda))$*.*

Proof. Follows by the leftover-hash lemma (Lemma 2.2). □

We now 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 $(c, k) \leftarrow \mathsf{Encap}(\mathsf{ID}), k' \leftarrow \mathsf{Ext}(k; r)$. 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'.

We show that the transformed system $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encap}_2, \mathsf{Encap}_2^*, \mathsf{Decap}_2)$ is leakage-smooth for appropriate parameters in the next theorem.

Theorem 3.2 *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*.*

Proof. The correctness and valid/invalid ciphertext indistinguishability properties of the modified scheme follow from those of the original. For leakage-smoothness, let us fix $\mathsf{mpk}, \mathsf{msk}, \mathsf{ID}$. Let f be any function with ℓ bit output. We define the following (correlated) random variables: $\mathsf{SK}_{\mathsf{ID}}$ is distributed according to $\mathsf{KeyGen}(\mathsf{ID}, \mathsf{msk})$, C is distributed according to $\mathsf{Encap}^*(\mathsf{ID})$, K is distributed according to $\mathsf{Decap}(C, \mathsf{SK}_{\mathsf{ID}})$ and R is a random and independent extractor seed. Also, we define a (probabilistic, and possibly inefficient) function $f'(c, k)$ which samples $\mathsf{sk}_{\mathsf{ID}}$ from the conditional distribution $(\mathsf{SK}_{\mathsf{ID}} \mid C = c, K = k)$ and outputs $f(\mathsf{sk}_{\mathsf{ID}})$. Then:

$$\begin{aligned}
\langle\, C' = (C, R), f(\mathsf{SK}_{\mathsf{ID}}), K' = \mathsf{Ext}(K; R)\,\rangle &\equiv \langle\, C' = (C, R), f'(C, K), K' = \mathsf{Ext}(K; R)\,\rangle \\
&\approx \langle\, C' = (C, R), f'(C, U_{\mathcal{K}}), K' = \mathsf{Ext}(U_{\mathcal{K}}; R)\,\rangle \quad (1) \\
&\approx \langle\, C' = (C, R), f'(C, U_{\mathcal{K}}), U_v\,\rangle \quad (2) \\
&\approx \langle\, C' = (C, R), f'(C, K), U_v\,\rangle \quad (3) \\
&\equiv \langle\, C' = (C, R), f(\mathsf{SK}_{\mathsf{ID}}), U_v\,\rangle
\end{aligned}$$

Equation (1) follows by the definition of smoothness. For equation (2), notice that

$$\widetilde{\mathbf{H}}_\infty(U_{\mathcal{K}} | C, f'(C, U_{\mathcal{K}})) \geq \widetilde{\mathbf{H}}_\infty(U_{\mathcal{K}} | C) - \ell \geq m - \ell$$

by Lemma 2.1, and the fact that $C, U_{\mathcal{K}}$ are independent. Then (2) follows from the definition of an extractor, where U_v is independent of the other variables. Lastly, equation (3) follows by another application of smoothness. □

3.3 Constructions

We show that the idea of an IB-HPS implicitly forms the backbone of the recent IBE constructions of [Gen06, BGH07, GPV08]. This gives us three constructions of IB-HPS, which are explicitly described and proven in the appendices. 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: this is $v = \log(|\mathcal{K}|)$, where $\mathcal{K}$ is the encapsulated-key set.
3. The min-entropy m and the universality ρ. These are 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. In Appendix B, we show that the IBE scheme of Gentry [Gen06], implicitly contains an IB-HPS construction. The scheme and the proof are essentially the same as those of [Gen06], and rely on the "truncated augmented bilinear Diffie-Hellman exponent" (TABDHE) assumption. It does *not* require the use of Random Oracles. The scheme is extremely efficient, requiring only a constant (2 or 3) number of group elements in the master public key, master secret key, identity secret key, and ciphertexts. The parameters of interest 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. In Appendix C, 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* (see Appendix C) 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. In Appendix D, 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 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 3.2, this construction is therefore *already* ℓ-leakage smooth, for any $\ell \leq m - \omega(\log(\lambda))$, without any need to apply an extractor.

4 Leakage-Resilient IBE

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

[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.

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.

4.1 Definition

Recall an IBE scheme consists of PPT algorithms $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$, an identity set $\mathcal{ID}$ and a message space $\mathcal{M}$. The syntax of $\mathsf{Setup}, \mathsf{KeyGen}$ is the same as that in $\mathsf{IB\text{-}HPS}$, and $\mathsf{Encrypt}, \mathsf{Decrypt}$ have the following syntax:

$c \leftarrow \mathsf{Encrypt}(\mathsf{ID}, \mathsf{m})$ **:** The encryption algorithm encrypts $\mathsf{m} \in \mathcal{M}$, and produces a ciphertext c.
$\mathsf{m} \leftarrow \mathsf{Decrypt}(c, \mathsf{sk}_{\mathsf{ID}})$ **:** The decryption algorithm decrypts a ciphertext c using the identity secret key $\mathsf{sk}_{\mathsf{ID}}$.

I. CORRECTNESS OF DECRYPTION. For any $(\mathsf{mpk}, \mathsf{msk})$ produced by $\mathsf{Setup}(1^\lambda)$, any $\mathsf{ID} \in \mathcal{ID}$, any $\mathsf{m} \in \mathcal{M}$, we have

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

II. SEMANTIC SECURITY WITH LEAKAGE. We define the *semantic security game*, parametrized 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: 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 $\mathcal{A}$ can adaptively ask the challenger for 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 4.1 (Leakage-Resilient IBE) *An* IBE *scheme is* ℓ-leakage-resilient*, if (1) it satisfies the correctness of decryption property, and (2) 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* $\mathsf{sk}_{\mathsf{ID}}$.

Remark on Stateful vs. Stateless Key Authority. In the semantic-security game with leakage, we assume that $\mathsf{sk}_{\mathsf{ID}} \leftarrow \mathsf{KeyGen}(\mathsf{ID}, \mathsf{msk})$ is computed only once per identity ID and reused subsequently. In reality, this requires that the key-authority that issues identity secret keys is stateful, and caches the secret keys that it computes. As noted in [Gen06, BGH07, GPV08], this requirement can be overcome easily and generically to get a stateless key-authority. We simply add a pseudo-random function $f \in_R \mathcal{F}$, from a PRF family $\mathcal{F}$, to the master secret key msk and always run $\mathsf{KeyGen}(\mathsf{ID}, \mathsf{msk})$ using random coins derived from $f(\mathsf{ID})$. That way the output is consistent each time KeyGen is called.

4.2 Construction of Leakage-Resilient IBE

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 $c = (c_1, c_2)$.
$\mathsf{Decrypt}(c, \mathsf{sk}_{\mathsf{ID}})$**:** Parse $c = (c_1, c_2)$ and 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 4.1 *Assume that we start with an ℓ-leakage-smooth* IB-HPS*. Then the above construction yields an ℓ-leakage-resilient* IBE*.*

Proof. The correctness of decryption, follows by the correctness of decapsulation. For the security analysis, we us a series of games argument:

Game 0: Define Game 0 to be the semantic security game with leakage ℓ. Notice that, in the challenge stage of Game 0, the challenger computes $c \leftarrow \mathsf{Encrypt}(\mathsf{ID}^*, \mathsf{m}_b)$ which we expand as $c = (c_1, c_2)$ where

$$(c_1, k) \leftarrow \mathsf{Encap}(\mathsf{ID}^*), c_2 = \mathsf{m}_b \oplus k.$$

Game 1: We modify the challenge stage, so that the challenger uses the secret key $\mathsf{sk}_{\mathsf{ID}}$ to compute the ciphertext $c = (c_1, c_2)$ by: [4]

$$(c_1, k_1) \leftarrow \mathsf{Encap}(\mathsf{ID}^*), k_2 \leftarrow \mathsf{Decap}(c_1, \mathsf{sk}_{\mathsf{ID}^*}), c_2 = \mathsf{m}_b \oplus k_2$$

The difference between Game 0 and Game 1 is only the use of k_1 versus k_2. But, by the correctness of decapsulation, $k_1 = k_2$ with all but negligible probability so Games 0 and 1 are (statistically) indistinguishable.

Game 2: In Game 2, we modify the challenge stage still further by having the challenger use a *invalid* encapsulation procedure to compute the ciphertext $c = (c_1, c_2)$:

$$c_1 \leftarrow \mathsf{Encap}^*(\mathsf{ID}^*), k_2 \leftarrow \mathsf{Decap}(c_1, \mathsf{sk}_{\mathsf{ID}^*}), c_2 = \mathsf{m}_b \oplus k_2.$$

We claim that Games 1 and 2 are computationally indistinguishable by the valid/invalid ciphertext indistinguishability of IB-HPS. Notice that, although the valid/invalid ciphertext indistinguishability game does not have *leakage queries*, it allows the adversary to learn all secret-keys, including the secret key $\mathsf{sk}_{\mathsf{ID}^*}$ of the challenge identity ID^*. Therefore indistinguishability between Games 1,2 holds *even* if the adversary sees the *full* challenge identity secret-key $\mathsf{sk}_{\mathsf{ID}^*}$, and hence certainly given just some bounded leakage $f(\mathsf{sk}_{\mathsf{ID}^*})$.

Game 3: In Game 3, the challenge ciphertext $c = (c_1, c_2)$ is computed by:

$$c_1 \leftarrow \mathsf{Encap}^*(\mathsf{ID}^*), c_2 \leftarrow U_{\mathcal{K}}.$$

We claim that Games 2 and 3 are statistically indistinguishable by the ℓ-leakage-smoothness of IB-HPS. Indeed, for fixed values of $\mathsf{mpk}, \mathsf{msk}$ the only values in Game 2 which are correlated to $\mathsf{sk}_{\mathsf{ID}^*}$ are the outputs of the ℓ leakage-queries, and $k_2 \leftarrow \mathsf{Decap}(c_1, \mathsf{sk}_{\mathsf{ID}^*})$. But, by ℓ-leakage smoothness, (thinking of the ℓ leakage queries together as a single randomized function $f^*(\mathsf{sk}^*_{\mathsf{ID}})$), this is (statistically) indistinguishable from choosing a completely independent $k_2 \leftarrow U_{\mathcal{K}}$, which is equivalent to Game 3.

Therefore Game 0 and Game 3 are indistinguishable by a PPT adversary. Also, it is clear that the advantage of any adversary in Game 3 is exactly 0 (since Game 3 is independent of the bit b chosen by the challenger). Therefore the advantage of any PPT adversary in Game 0 is at most negligibly different from that of Game 3, and hence negligible in λ, as we wanted to show. □

[4] The value $\mathsf{sk}_{\mathsf{ID}^*}$ is either already defined if ID^* was part of a leakage/secret-key query, or chosen fresh from $\mathsf{KeyGen}(\mathsf{ID}^*, \mathsf{msk})$ and used to respond to future queries otherwise

5 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.

Figure 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 5.1 *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 is some setting of $t = O(v + \lambda)$ 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 for $\ell(\lambda) = (1 - \varepsilon)nm - v - \lambda$. The encapsulated-key-set of Π_2 is $\mathcal{K} = \{0,1\}^v$.*

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 3.1 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. We analyze the leakage smoothness of the construction in

Appendix E. First, we define a new notion called *approximately universal hashing* (Definition E.3), 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 (Lemma 2.2), called the *approximate leftover-hash lemma* (Theorem E.2) holds for approximate hashing. Lastly, in Appendix E.3, we show that the decapsulation procedure $\mathsf{Decap}_2(C, \mathsf{sk}_{\mathsf{ID}})$ is approximately universal, for appropriate parameters, when $C \leftarrow \mathsf{Encap}^*(\mathsf{ID})$.[5] Combining these results, we get the parameters of the theorem.

6 Public-Key Encryption and IBE in the BRM

A public-key encryption (PKE) scheme in the BRM consists of the algorithms $(\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$, 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 6.1 (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 6.2 (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:*

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

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

We can generalize the above definition to IBE schemes, and say that 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 6.1 (PKE and IBE in BRM) *Assume that we have an* (m, ρ)*-universal* $\mathsf{IB\text{-}HPS}$ *satisfying the conditions of Theorem 5.1 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* $\mathsf{IB\text{-}HPS}$.
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* $\mathsf{IB\text{-}HPS}$.
4. *Relative leakage is* $\alpha \ge \frac{m}{\hat{m}}(1 - \varepsilon)$*, 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* $\le \frac{\hat{m}}{m}(1 + \varepsilon)\ell$.

[5] For approximate universality, we think of $\mathsf{sk}_{\mathsf{ID}}$ as consisting of n symbols of an alphabet Σ, with one symbol for each component key $\mathsf{sk}_{\mathsf{ID}}[i]$. For the size $q = |\Sigma|$, we can consider an abstract (not necessarily efficient) representation of the keys $\mathsf{sk}_{\mathsf{ID}}[i]$, so $q \le 2^{m'}$.

Proof. Follows directly from leakage-amplification (Theorem 5.1). 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 4.1, 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. □

7 Extensions

CCA SECURITY. In Appendix F 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 EncapC, EcnapK (like Encap) are implicitly parameterized by the master public key mpk.

Definition 7.1 (Anonymous Encapsulation) *An* IB-HPS *has* anonymous encapsulation *if there exist efficient procedures* EncapC, EcnapK *as above, such that, for any fixed* mpk, msk, 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 our lattice-based 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.[6] 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*, or *hitter* (see Definition E.2) to 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.

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

In Appendix E.4, 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.

8 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 $v = O(\lambda)$.[7] 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 $t = O(\lambda)$. 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.[8] The value $\varepsilon > 0$ can be an arbitrary constant.

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

† = Random Oracle Model/Interactive Assumption

Table 1: Comparison of Our PKE in BRM Constructions

9 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

[ADR02] Yonatan Aumann, Yan Zong Ding, and Michael O. Rabin. Everlasting security in the bounded storage model. *IEEE Transactions on Information Theory*, 48(6):1668–1680, 2002.

[ADW09] Joël Alwen, Yevgeniy Dodis, and Daniel Wichs. Leakage-resilient public-key cryptography in the bounded-retrieval model. In *CRYPTO*, pages 36–54, 2009.

[AGV09] A. Akavia, S. Goldwasser, and V. Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In Omer Reingold, editor, *Theory of Cryptography — TCC 2009*, volume 5444 of *Lecture Notes in Computer Science*. Springer-Verlag, 2009.

[Ajt99] Miklós Ajtai. Generating hard instances of the short basis problem. In Jirí Wiedermann, Peter van Emde Boas, and Mogens Nielsen, editors, *ICALP*, volume 1644 of *Lecture Notes in Computer Science*, pages 1–9. Springer, 1999.

[AP09] Joël Alwen and Chris Peikert. Generating shorter bases for hard random lattices. In Susanne Albers and Jean-Yves Marion, editors, *STACS*, volume 09001 of *Dagstuhl Seminar Proceedings*, pages 75–86. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2009.

[AR99] Yonatan Aumann and Michael O. Rabin. Information theoretically secure communication in the limited storage space model. In Wiener [Wie99], pages 65–79.

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

[8] 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 $O(\lambda)$.

[BDL97] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the importance of checking cryptographic protocols for faults (extended abstract). In *EUROCRYPT*, pages 37–51, 1997.

[BGH07] Dan Boneh, Craig Gentry, and Michael Hamburg. Space-efficient identity based encryption without pairings. In *FOCS*, pages 647–657, 2007.

[BS97] Eli Biham and Adi Shamir. Differential fault analysis of secret key cryptosystems. In Burton S. Kaliski Jr., editor, *CRYPTO*, volume 1294 of *Lecture Notes in Computer Science*, pages 513–525. Springer, 1997.

[CDD+07] David Cash, Yan Zong Ding, Yevgeniy Dodis, Wenke Lee, Richard J. Lipton, and Shabsi Walfish. Intrusion-resilient key exchange in the bounded retrieval model. In Salil P. Vadhan, editor, *TCC*, volume 4392 of *Lecture Notes in Computer Science*, pages 479–498. Springer, 2007.

[CDH+00] Ran Canetti, Yevgeniy Dodis, Shai Halevi, Eyal Kushilevitz, and Amit Sahai. Exposure-resilient functions and all-or-nothing transforms. In *EUROCRYPT*, pages 453–469, 2000.

[CLW06] Giovanni Di Crescenzo, Richard J. Lipton, and Shabsi Walfish. Perfectly secure password protocols in the bounded retrieval model. In Halevi and Rabin [HR06], pages 225–244.

[CS02] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *EUROCRYPT*, volume 2332 of *Lecture Notes in Computer Science*, pages 45–64. Springer, 2002.

[DDN00] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.

[DKL09] Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett. On cryptography with auxiliary input. In *STOC*, 2009.

[DORS08] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008.

[DP07] Stefan Dziembowski and Krzysztof Pietrzak. Intrusion-resilient secret sharing. In *FOCS*, pages 227–237. IEEE Computer Society, 2007.

[DP08] Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *FOCS*, pages 293–302. IEEE Computer Society, 2008.

[DSS01] Yevgeniy Dodis, Amit Sahai, and Adam Smith. On perfect and adaptive security in exposure-resilient cryptography. In *EUROCRYPT*, pages 301–324, 2001.

[Dzi06] Stefan Dziembowski. Intrusion-resilience via the bounded-storage model. In Halevi and Rabin [HR06], pages 207–224.

[FKPR09] Sebastian Faust, Eike Kiltz, Krzysztof Pietrzak, and Guy Rothblum. Leakage-resilient signatures. Cryptology ePrint Archive, Report 2009/282, 2009. `http://eprint.iacr.org/`.

[Gen06] Craig Gentry. Practical identity-based encryption without random oracles. In Serge Vaudenay, editor, *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 445–464. Springer, 2006.

[GMO01] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic analysis: Concrete results. In Çetin Kaya Koç, David Naccache, and Christof Paar, editors, *CHES*, volume 2162 of *Lecture Notes in Computer Science*, pages 251–261. Springer, 2001.

[Gol97] Oded Goldreich. A sample of samplers - a computational perspective on sampling (survey). volume 4, 1997.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC '08: Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 197–206, New York, NY, USA, 2008. ACM.

[HR06] Shai Halevi and Tal Rabin, editors. *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, volume 3876 of *Lecture Notes in Computer Science*. Springer, 2006.

[HSH+08] J. Alex Halderman, Seth D. Schoen, Nadia Heninger, William Clarkson, William Paul, Joseph A. Calandrino, Ariel J. Feldman, Jacob Appelbaum, and Edward W. Felten. Lest we remember: Cold boot attacks on encryption keys. In Paul C. van Oorschot, editor, *USENIX Security Symposium*, pages 45–60. USENIX Association, 2008.

[ISW03] Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In *CRYPTO*, pages 463–481, 2003.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Wiener [Wie99], pages 388–397.

[Koc96] Paul C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In Neal Koblitz, editor, *CRYPTO*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, 1996.

[KPSY09] Eike Kiltz, Krzysztof Pietrzak, Martijn Stam, and Moti Yung. A new randomness extraction paradigm for hybrid encryption. In Antoine Joux, editor, *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 590–609. Springer, 2009.

[KV09] Jonathan Katz and Vinod Vaikuntanathan. Signature schemes with bounded leakage resilience, 2009. To Appear in Asiacrypt '09. `http://www.mit.edu/~vinodv/papers/asiacrypt09/KV-Sigs.pdf`.

[KZ03] Jesse Kamp and David Zuckerman. Deterministic extractors for bit-fixing sources and exposure-resilient cryptography. In *FOCS*, pages 92–101, 2003.

[Lin06] Yehuda Lindell. A simpler construction of CCA2-secure public-key encryption under general assumptions. *Journal of Cryptology*, 19(3):359–377, 2006.

[Lu02] Chi-Jen Lu. Hyper-encryption against space-bounded adversaries from on-line strong extractors. In Moti Yung, editor, *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 257–271. Springer, 2002.

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

[MR04a] Silvio Micali and Leonid Reyzin. Physically observable cryptography (extended abstract). In Moni Naor, editor, *TCC*, volume 2951 of *Lecture Notes in Computer Science*, pages 278–296. Springer, 2004.

[MR04b] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. In *FOCS '04: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 372–381, Washington, DC, USA, 2004. IEEE Computer Society.

[NS09] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In *CRYPTO*, pages 18–35, 2009.

[NY90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, pages 427–437, 1990.

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

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *STOC*, pages 333–342, 2009.

[Pie09] Krzysztof Pietrzak. A leakage-resilient mode of operation. In *Eurocrypt 2009, Cologne, Germany*, 2009.

[PR06] Chris Peikert and Alon Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In Halevi and Rabin [HR06], pages 145–166.

[QS01] Jean-Jacques Quisquater and David Samyde. Electromagnetic analysis (ema): Measures and counter-measures for smart cards. In Isabelle Attali and Thomas P. Jensen, editors, *E-smart*, volume 2140 of *Lecture Notes in Computer Science*, pages 200–210. Springer, 2001.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93, 2005.

[Sah99] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*, pages 543–553, 1999.

[Sho05] Victor Shoup. *A computational introduction to number theory and algebra*. Cambridge University Press, New York, NY, USA, 2005.

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

[Vad04] Salil P. Vadhan. Constructing locally computable extractors and cryptosystems in the bounded-storage model. *J. Cryptology*, 17(1):43–77, 2004.

[Vai09] Vinod Vaikuntanathan. Personal Communication, 2009.

[Wie99] Michael J. Wiener, editor. *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*. Springer, 1999.

A Counterexample to PKE in BRM Construction from General Leakage-Resilient IBE

We give a counterexample showing that a generic construction of PKE in the BRM from a leakage-resilient IBE, via parallel repetition, does not work. More specifically we construct an IBE with leakage ℓ, such that parallel-repetition does not amplify leakage-resilience beyond some small value, no matter how many "repetitions" n are taken.

As a start, assume that I is any IBE scheme. We first construct an IBE I' where the set of identities is exactly $[n]$ (which is only polynomial). The scheme is the same as I otherwise, except that the identity-secret-key generation procedure also gives a share S_i of the master secret key msk to each identity $i \in [n]$, along with its identity secret key. We assume the share is computed using an n-out-of-n secret sharing scheme. The resulting scheme I' is still and IBE (albeit with a small, polynomially size identity set) since, after observing the identity secret key of up to $n-1$ identities, the master secret key is still perfectly hidden. Moreover, the scheme is leakage-resilient, at least for some small (logarithmic in the security parameter) leakage ℓ (this must be the case for *any* IBE/PKE since a logarithmic number of bits of leakage can be efficiently guessed with polynomial probability). Nevertheless, the PKE construction that results from n-wise parallel-repetition is not leakage-resilient for any ℓ greater than the size of the master secret key of I, no matter how large n is. Indeed a valid leakage attack can look at all shares $S_1, \ldots, S_n$, and output the master secret key.

One objection to the counterexample, is that the IBE scheme I' only has a polynomial number of identities, and so it is not a legitimate IBE. We can get around this by defining a scheme I'' which runs one copy of I' for identities $\{1, \ldots, n\}$ and an independent copy of the original scheme I for all other (exponentially many) identities. Then I'' is a proper IBE, but leakage-amplification still fails.

B A Construction of IB-HPS Based on Bilinear Groups

B.1 Review of Bilinear Groups and Assumptions

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, on input 1^λ, 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 B.1 *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.*

B.2 Construction of IB-HPS based on Gentry's IBE [Gen06].

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. For completeness, we include the proof tailored to our presentation and terminology below.

Theorem B.1 *Under the* TABDHE *assumption, the above construction is an* IB-HPS *which is simultaneously* smooth *and* (m, ρ)-universal *for* $\rho = 0$*. More precisely, the valid/invalid ciphertext indistinguishability property holds under the* q-TABDHE *assumption for any adversary making at most* q *queries in test stages 1,2. Moreover:*

1. *The identity-key entropy is* $m = \log(p)$.
2. *The actual identity-key size is* $\hat{m} = 2\log(p) + O(1)$.[9]
3. *The effective-key size (logarithm of the number of values that any* $\mathsf{sk}_{\mathsf{ID}}$ *can take on) is* $m' = \log(p)$.
4. *The encapsulated-key size is* $v = \log(p)$.

Proof. Let us write $h = g^\beta$ for some $\beta \in \mathbb{Z}_p$, so that $h_{\mathsf{ID}} = g^{(\beta - r_{\mathsf{ID}})/(\alpha - \mathsf{ID})}$ for each $\mathsf{ID} \in \mathbb{Z}_p$.

I. For *correctness* we see that, for any $\mathsf{ID} \in \mathbb{Z}_p$, any correctly generated $\mathsf{mpk}, \mathsf{msk}, \mathsf{sk}_{\mathsf{ID}}$, if a pair $(c = (u, v), k)$ is generated by $\mathsf{Encap}(\mathsf{ID})$ then $u = g_1^s g^{-s\mathsf{ID}} = g^{s(\alpha - \mathsf{ID})}$, $v = e(g,g)^s$ for some $s \in \mathbb{Z}_p$ and $k = e(g,h)^s$. Correctness follows since:

$$\begin{aligned} \mathsf{Decap}(c, \mathsf{sk}_{\mathsf{ID}}) &= e(u, h_{\mathsf{ID}})v^{r_{\mathsf{ID}}} \\ &= e\left(g^{s(\alpha - \mathsf{ID})}, g^{(\beta - r_{\mathsf{ID}})/(\alpha - \mathsf{ID})}\right) e(g,g)^{sr_{\mathsf{ID}}} \\ &= e(g,g)^{s\beta} = e(g,h)^s = k. \end{aligned}$$

II. For *valid/invalid ciphertext indistinguishability*, we show how to use an adversary $\mathcal{A}$, which distinguishes valid and invalid ciphertexts using q queries, to create an adversary $\mathcal{B}$ which is a distinguisher for the q-TABDHE problem. In particular, the algorithm $\mathcal{B}$ receives as input $(g, g_1, \ldots, g_q, g', g'_{q+2}, Z)$ where $g_i = g^{(\alpha^i)}$, $g'_i = g'^{(\alpha^i)}$ for an unknown α, and Z is either $e(g_{q+1}, g')$ or a random element of $\mathbb{G}_T$. The algorithm $\mathcal{B}$ simulates the valid/invalid ciphertext distinguishability game for $\mathcal{A}$ as follows:

Setup: The algorithm $\mathcal{B}$ chooses a polynomial $f(x) \in \mathbb{Z}_p[x]$ of degree q uniformly at random, and computes $h = g^{f(\alpha)}$ using $g_1, \ldots, g_q$ (and without knowledge of α). The value $\mathsf{mpk} = (g, g_1 = g^\alpha, h = g^{f(\alpha)})$ is given to $\mathcal{A}$.

Test Stage 1: Whenever $\mathcal{A}$ makes a query for a new value of $\mathsf{ID} \in \mathbb{Z}_p$, the algorithm $\mathcal{B}$ computes $r_{\mathsf{ID}} = f(\mathsf{ID})$ and sets $h_{\mathsf{ID}} = g^{F_{\mathsf{ID}}(\alpha)}$ where $F_{\mathsf{ID}}(x) = (f(x) - f(\mathsf{ID}))/(x - \mathsf{ID})$ is a polynomial of degree $q-1$ and thus h_{ID} can be efficiently computed using $g_1, \ldots, g_{q-1}$ without knowing α. $\mathcal{B}$ gives $\mathsf{sk}_{\mathsf{ID}} = (r_{\mathsf{ID}}, h_{\mathsf{ID}})$ to the attacker $\mathcal{A}$.

[9] We use the fact that many elliptic-curves with pairings have source groups whose representation size is not much larger than its order.

Challenge Stage: The attacker $\mathcal{A}$ chooses an identity ID^*. Let $\mathsf{sk}_{\mathsf{ID}^*} = (r_{\mathsf{ID}^*}, h_{\mathsf{ID}^*})$ be computed as in the Test Stage. Let $H_{\mathsf{ID}}(x) = (x^{q+2} - \mathsf{ID}^{q+2})/(x - \mathsf{ID})$ be the polynomial of degree $q+1$, in which the x^{q+1} term has coefficient 1. Let $H^-_{\mathsf{ID}}(x) = H_{\mathsf{ID}}(x) - x^{q+1}$ be the polynomial of degree q. Then $\mathcal{B}$ sets:

$$u = \frac{g'_{q+2}}{g'^{((\mathsf{ID}^*)^{q+2})}} = g'^{\left(\alpha^{q+2} - (\mathsf{ID}^*)^{q+2}\right)}, v = Z \cdot \left(g', g^{H^-_{\mathsf{ID}^*}(\alpha)}\right)$$

where $g^{H^-_{\mathsf{ID}^*}(\alpha)}$ is computed using $g_1, \ldots, g_q$, and gives $c = (u, v)$ to $\mathcal{A}$.

Test Stage 2: The algorithm $\mathcal{B}$ responds to ID queries the same way as in Test Stage 1.

Output: If the adversary $\mathcal{A}$ outputs $b' = 0$ (indicating a valid ciphertext), the algorithm $\mathcal{B}$ outputs 0 (indicating that $Z = e(g_{q+1}, g')$) and if $\mathcal{A}$ outputs 1 then $\mathcal{B}$ also outputs 1.

We show that the view of $\mathcal{A}$ is statistically close to a run of the valid/invalid ciphertext indistinguishability game. Firstly, since f is random degree q polynomial,and is evaluated at at most $q+1$ values (the ID queries of $\mathcal{A}$ and at α) all of the outputs are mutually independent and uniform over $\mathbb{Z}_p$. In particular, letting $\beta = f(\alpha)$, this means that $\mathsf{mpk} = (g, g_1 = g^\alpha, h = g^\beta)$ and the identity secret keys $\mathsf{sk}_{\mathsf{ID}} = (r_{\mathsf{ID}}, h_{\mathsf{ID}} = g^{(\beta - r_{\mathsf{ID}})/(\alpha - \mathsf{ID})})$ seen during the Test Stages are (mutually) chosen from the same distribution as in the valid/invalid ciphertext indistinguishability game.

Now let us look at the challenge key-ciphertext. We write $g' = g^\gamma$ for a random (unknown) γ. When $Z = e(g_{q+1}, g')$ then the challenge ciphertext is $u = g^{s(\alpha - \mathsf{ID}^*)}$ for $s = \gamma H_{\mathsf{ID}^*}(\alpha)$, which is uniformly random for a random γ. Also,

$$v = e(g', g)^{H_{\mathsf{ID}^*}(\alpha)} = e(g', g)^{s/\gamma} = e(g, g)^s.$$

Therefore this (perfectly) corresponds to the distribution seen by $\mathcal{A}$ when the challenger chooses $b = 0$ (i.e. a valid key-ciphertext). On the other hand, when Z is uniformly random, then $v = e(g, g)^{s'}$ for a random s', independent of s so that (u, v) are uniform over $\mathbb{G} \times \mathbb{G}_T$. This is $1/p$ statistically close to the distribution of invalid ciphertexts output by $\mathsf{Encap}^*(\mathsf{ID}^*)$ and thus the case where the challenger chooses $b = 1$ (i.e. an invalid key-ciphertext). Therefore the advantage of $\mathcal{B}$ in the q-TABDHE game is negligibly close to the advantage of $\mathcal{A}$ in the valid/invalid indistinguishability game.

III. To show *smoothness* and *ρ-universality*, fix any $\mathsf{mpk}, \mathsf{msk}, \mathsf{ID}$. Let c be some output of $\mathsf{Encap}^*(\mathsf{ID})$, so that $c = (u, v)$ for $u = g^{s(\alpha - \mathsf{ID})}$ and $v = e(g, g)^{s'}$ where $s \neq s'$. Then, for any secret key $\mathsf{sk}_{\mathsf{ID}} = (r_{\mathsf{ID}}, h_{\mathsf{ID}} = g^{(\beta - r_{\mathsf{ID}})/(\alpha - \mathsf{ID})})$ we get:

$$\begin{aligned}
\mathsf{Decap}(c, \mathsf{sk}_{\mathsf{ID}}) &= e(u, h_{\mathsf{ID}}) v^{r_{\mathsf{ID}}} \\
&= e\left(g^{s(\alpha - \mathsf{ID})}, g^{(\beta - r_{\mathsf{ID}})/(\alpha - \mathsf{ID})})\right) v^{r_{\mathsf{ID}}} \\
&= e(g, g)^{s(\beta - r_{\mathsf{ID}})} e(g, g)^{s' r_{\mathsf{ID}}} \\
&= e(g, g)^{s\beta + (s' - s) r_{\mathsf{ID}}}
\end{aligned}$$

Therefore:

1. For any fixed c output by $\mathsf{Encap}^*(\mathsf{ID})$, the distribution of $\mathsf{Decap}(c, \mathsf{sk}_{\mathsf{ID}})$ (over a uniform $\mathsf{sk}_{\mathsf{ID}} \leftarrow \mathsf{KeyGen}(\mathsf{ID}, \mathsf{msk})$) is uniform over $\mathcal{K} = \mathbb{G}_T$. This implies *smoothness*.
2. If $\mathsf{Decap}(c, \mathsf{sk}_{\mathsf{ID}}) = \mathsf{Decap}(c, \mathsf{sk}'_{\mathsf{ID}})$ and $\mathsf{sk}_{\mathsf{ID}}, \mathsf{sk}'_{\mathsf{ID}}$ are outputs of $\mathsf{KeyGen}(\mathsf{ID}, \mathsf{msk})$ then $\mathsf{sk}_{\mathsf{ID}} = \mathsf{sk}'_{\mathsf{ID}}$. This implies 0-*universality*.

□

C A Construction of IB-HPS Based on Quadratic-Residuosity

C.1 Review of Terminology, the QR Assumption, Background

For a positive integer N, let $\mathcal{J}(N)$ denote the set $\mathcal{J}(N) \stackrel{\text{def}}{=} \{x \in \mathbb{Z}_N : \left(\frac{x}{N}\right) = 1\}$ where $\left(\frac{x}{N}\right)$ denotes the Jacobi symbol of x in $\mathbb{Z}_N$. Let $\mathrm{QR}(N) \subseteq \mathcal{J}(N)$ denote the set of quadratic residues modulo N.

Let $\mathsf{PrimeGen}(1^\lambda)$ be an algorithm which, given security parameter λ, outputs two primes (p, q) of length which is exponential in λ. We define the distributions:

DQR_λ **:** Choose $(p, q) \leftarrow \mathsf{PrimeGen}(1^\lambda)$, $N = pq$, $S \leftarrow \mathrm{QR}(N)$. Output (N, S).
DNQR_λ **:** Choose $(p, q) \leftarrow \mathsf{PrimeGen}(1^\lambda)$, $N = pq$, $S \leftarrow \mathcal{J}(N) \setminus \mathrm{QR}(N)$. Output (N, S).

The quadratic residuosity assumptions states that no PPT adversary can distinguish the distributions DQR_λ and DNQR_λ.

Definition C.1 (QR Assumption) *The quadratic residuosity assumptions states that for any PPT algorithm $\mathcal{A}$,*

$$|\Pr[\mathcal{A}(\mathrm{DQR}_\lambda) = 1] - \Pr[\mathcal{A}(\mathrm{DNQR}_\lambda) = 1]| \leq \mathrm{negl}(\lambda).$$

Recall the following elementary property of the Jacobi symbol.

Lemma C.1 *For any $x, y \in \mathbb{Z}_N$, $\left(\frac{x}{N}\right)\left(\frac{y}{N}\right) = \left(\frac{xy}{N}\right)$. Therefore, if $xy \in \mathcal{J}(N)$. Then $\left(\frac{x}{N}\right) = \left(\frac{y}{N}\right)$.*

We will also rely on the following two lemmas shown in [BGH07].

Lemma C.2 ([BGH07]) *Let N be a product of two primes, let $X \in \mathrm{QR}(N)$, and $S \in \mathcal{J}(N)$. Let $x_1, x_2 \in \mathbb{Z}_n$ be any two square roots of X. Let f be a polynomial such that $f(x)f(-x)S$ is a quadratic residue for all four square roots x of X. Then:*

- *When $S \notin \mathrm{QR}(N)$, the Jacobi symbols of x_1, x_2 are different $\left(\frac{f(x_1)}{N}\right) \neq \left(\frac{f(x_2)}{N}\right)$ iff $x_1 \neq -x_2$.*
- *When $S \in \mathrm{QR}(N)$, the Jacobi symbols of x_1, x_2 are always the same $\left(\frac{f(x_1)}{N}\right) = \left(\frac{f(x_2)}{N}\right)$.*

Lemma C.3 ([BGH07]) *There exists an* efficient and deterministic *algorithm $\mathcal{Q}$ which takes as input (N, u, R, S), where $N \in \mathbb{Z}^+$, $u, R, S \in \mathbb{Z}_N$, and outputs polynomials $f, \overline{f}, g, \tau \in \mathbb{Z}_N[x]$ satisfying the following conditions:*

1. *If $R, S \in \mathrm{QR}(N)$, then $f(r)g(s) \in \mathrm{QR}(N)$ for all square roots r of R and s of S.*
2. *If $uR, S \in \mathrm{QR}(N)$, then $\overline{f}(\overline{r})g(s)\tau(s) \in \mathrm{QR}(N)$ for all square roots $\overline{r}$ of uR and s of S.*
3. *If $R \in \mathrm{QR}(N)$, then $f(r)f(-r)S \in \mathrm{QR}(N)$ for all square roots r of R.*
4. *If $uR \in \mathrm{QR}(N)$, then $\overline{f}(\overline{r})\overline{f}(-\overline{r})S \in \mathrm{QR}(N)$ for all square roots $\overline{r}$ of uR.*
5. *If $S \in \mathrm{QR}(N)$, then $\tau(s)\tau(-s)u \in \mathrm{QR}(N)$ for all square roots s of S.*
6. *For any fixed values of N, u, S, the polynomial τ output by $\mathcal{Q}(N, u, R, S)$ is the same for all choices of R.*

Lastly, we review the *interactive QR (IQR)* assumption of [BGH07]. Let $\mathsf{PrimeGen}$ be as before, and let $H_N : \{0,1\}^* \to \mathcal{J}(N)$. We define two IQR oracles $\mathcal{O}_\lambda^{\mathrm{QR}}$ and $\mathcal{O}_\lambda^{\mathrm{NQR}}$ which work as follows:

- The oracle selects $(p, q) \leftarrow \mathsf{PrimeGen}(1^\lambda)$ and outputs $N = pq$.
- The oracle selects $u \leftarrow \mathcal{J}(N) \setminus \mathrm{QR}(N)$ and outputs u.
- The oracle $\mathcal{O}_\lambda^{\mathrm{QR}}$ selects $S \leftarrow \mathrm{QR}(N)$ while $\mathcal{O}_\lambda^{\mathrm{NQR}}$ selects $S \leftarrow \mathcal{J}(N)/\mathrm{QR}(N)$. The oracle outputs S.
- On each input $x \in \{0,1\}^*$, the oracle computes $R = H_N(x)$ and outputs a random square-root of either R or uR, depending on which one is a residue.

Definition C.2 (IQR Assumption of [BGH07]) *The IQR assumption for a pair $(\mathsf{PrimeGen}, H_N)$ states that no PPT adversary $\mathcal{A}$ can distinguish oracle access to $\mathcal{O}_\lambda^{\mathrm{QR}}$ from that to $\mathcal{O}_\lambda^{\mathrm{NQR}}$. That is,*

$$\left|\Pr\left[\mathcal{A}^{\mathcal{O}_\lambda^{\mathrm{QR}}}(1^\lambda) = 1\right] - \Pr\left[\mathcal{A}^{\mathcal{O}_\lambda^{\mathrm{NQR}}}(1^\lambda) = 1\right]\right| \leq \mathrm{negl}(\lambda).$$

It is easy to see (as was shown in [BGH07]) that the IQR assumption follows from the standard QR assumption when the hash-function H_N is modeled as a Random Oracle.

C.2 Construction of IB-HPS Based on the IBE of Boneh Gentry and Hamburg [BGH07]

We now present the construction of IB-HPS which is based on an IBE scheme of Boneh Gentry and Hamburg [Gen06], with some small modifications. In the description, $\mathcal{Q}$ is the algorithm defined by Lemma C.3 and PrimeGen is the prime-generation of Definition C.1.

$\mathsf{Setup}(1^\lambda)$ **:** Choose $(p,q) \leftarrow \mathsf{PrimeGen}(1^\lambda)$ and set $N = pq$. Sample $u \leftarrow \mathcal{J}(N) \setminus \mathrm{QR}(N)$. Let $H : \{0,1\}^* \to \mathcal{J}(N)$. Output $\mathsf{mpk} = (N, u, H)$ and $\mathsf{msk} = (p,q)$. The identity set is $\mathcal{ID} = \{0,1\}^*$. The encapsulate-key set is $\mathcal{K} = \{\pm 1\}$.

$\mathsf{KeyGen}(\mathsf{ID}, \mathsf{msk})$ **:** Let $R = H(\mathsf{ID})$. Let $a \in \{0,1\}$ be the unique choice for which $u^a R \in \mathrm{QR}(N)$. Let $\{r_1, r_2, r_3, r_4\}$ be a labeling of the four square-roots of $u^a R$ so that $r_1 < r_2 < r_3 < r_4$ (in $\mathbb{Z}$) and $r_1 = -r_4$, $r_2 = -r_3$. Choose $r \leftarrow \{r_1, r_2\}$, output $\mathsf{sk}_{\mathsf{ID}} = r$.

$\mathsf{Encap}(\mathsf{ID})$ **:** Choose $s \leftarrow \mathbb{Z}_N$ and set $S = s^2$. Run $\mathcal{Q}(N, u, 1, S)$ to obtain τ and compute $b = \left(\frac{\tau(s)}{N}\right)$. Set $c = (S, b)$. Let $R = H(\mathsf{ID})$. Run $\mathcal{Q}(N, u, R, s)$ to obtain a polynomial g, set $k = \left(\frac{g(s)}{N}\right)$. Output (c, k).

$\mathsf{Encap}^*(\mathsf{ID})$ **:** Choose $S \leftarrow \mathcal{J}(N) \setminus \mathrm{QR}(N)$, $b \leftarrow \{\pm 1\}$. Outputs $c = (S, b)$.

$\mathsf{Decap}(c, \mathsf{sk}_{\mathsf{ID}})$ **:** Let $r = \mathsf{sk}_{\mathsf{ID}}$, $R = H(\mathsf{ID})$. Parse $c = (S, b)$. If $r^2 = R$, output $k = \left(\frac{f(r)}{N}\right)$ else output $k = b\left(\frac{\overline{f}(r)}{N}\right)$, where $f, \overline{f}$ are the polynomials output by $\mathcal{Q}(N, u, R, S)$.

Theorem C.1 *Under the IQR assumption on the pair* $(\mathsf{PrimeGen}, H)$*, the above construction is* (m, ρ)*-Universal and smooth* IB-HPS *with* $\rho = 0$*. In particular, this is also the case under the QR assumption, when the function* H *is modeled as a Random Oracle. Moreover:*

1. *The identity-key entropy is* $m = 1$.
2. *The actual identity-key size is* $\hat{m} = \log(N)$.
3. *The effective-key size (logarithm of the number of values that any* $\mathsf{sk}_{\mathsf{ID}}$ *can take on) is* $m' = 1$.
4. *The encapsulated-key size is* $v = 1$.

Proof. We show correctness, valid/invalid ciphertext indistinguishability and universality separately.

I. CORRECTNESS Let $S \in \mathrm{QR}(N)$ and s be an arbitrary square-root of S. Let $R \in \mathcal{J}(N)$. Let $g, f, \overline{f}, \tau$ be outputs of $\mathcal{Q}(N, u, R, S)$, which results in the same τ as that output by $\mathcal{Q}(N, u, 1, S)$ by property 6 of $\mathcal{Q}$. To show correctness, it suffices to show that:

- If $R \in \mathrm{QR}(N)$ then $k = \left(\frac{g(s)}{N}\right) = \left(\frac{f(r)}{N}\right)$ for any square-root r of R.
- If $R \notin \mathrm{QR}(N)$ then $k = \left(\frac{g(s)}{N}\right) = \left(\frac{\tau(s)}{N}\right)\left(\frac{\overline{f}(\overline{r}}{N}\right)$ where $\overline{r}$ is any square-root of uR.

For the first bullet, we get $f(r)g(s) \in \mathrm{QR}(N)$ by property 1 of the algorithm $\mathcal{Q}$ (Lemma C.3) and so correctness follows by Lemma C.1. For the second bullet, we get $f(\overline{r})g(s)\tau(s) \in \mathrm{QR}(N)$ by property 2 of the algorithm $\mathcal{Q}$. Correctness then follows by Lemma C.1.

II. VALID/INVALID CIPHERTEXT INDISTINGUISHABILITY Assume that there is a PPT adversary $\mathcal{A}$ that distinguishes valid and invalid ciphertexts with non-negligible probability. We use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ for the IQR assumption. Essentially, $\mathcal{B}$ gets oracle access to some oracle $\mathcal{O}$ which is either $\mathcal{O}_\lambda^{\mathrm{QR}}$ or $\mathcal{O}_\lambda^{\mathrm{NQR}}$. As a first step, $\mathcal{B}$ receives (N, u, S) from its oracle. Then $\mathcal{B}$ simulates the valid/invalid ciphertext indistinguishability game for $\mathcal{A}$ as follows:

Key Setup: Give $\mathsf{mpk} = (N, u, H)$ to the adversary $\mathcal{A}$.
Test Stage 1: For each query to $\mathsf{ID} \in \{0,1\}^*$, the adversary $\mathcal{B}$ submits ID to its oracle $\mathcal{O}$ and receives an output r. It then outputs either r or $-r$ depending on the which one is smaller in $\mathbb{Z}$.
Challenge Stage: No matter what the challenge ID is, choose $b \leftarrow \{\pm 1\}$ and give (S, b) to $\mathcal{A}$.
Test Stage 2: Same as Test Stage 1.

At the end $\mathcal{B}$ outputs whatever $\mathcal{A}$ does. It is easy to see the the key setup, and the test stages 1,2, are simulated correctly. For the challenge phase:

- If the oracle $\mathcal{O}$ is $\mathcal{O}_\lambda^{\mathrm{NQR}}$ then the ciphertext $c = (S, b)$ is uniform over $(\mathcal{J}(N) \setminus \mathrm{QR}(N), \{\pm 1\})$, which is the same as when the challenger samples $c \leftarrow \mathsf{Encap}^*(\mathsf{ID})$.
- If the oracle $\mathcal{O}$ is $\mathcal{O}_\lambda^{\mathrm{QR}}$ then the ciphertext $c = (S, b)$ is uniform over $(\mathrm{QR}(N), \{\pm 1\})$. We claim this is the same as when the challenger samples $c \leftarrow \mathsf{Encap}(\mathsf{ID})$, where $S \leftarrow \mathrm{QR}(N)$ and $b = \left(\frac{\tau(s)}{N}\right)$ for a random square-root s of S. This follows by property 5 of $\mathcal{Q}$ in conjunction with Lemma C.2 which tells us that, for a fixed S, b is uniformly random over $\{\pm 1\}$ over a random square-root s of S.

Therefore the distinguishing advantage of $\mathcal{A}$ in the valid/invalid ciphertext indistinguishability game is the same as the advantage of $\mathcal{B}$ in the IQR game, which proved the theorem.

III. UNIVERSALITY For any fixed $\mathsf{mpk} = (N, u, H), \mathsf{msk} = (p, q), \mathsf{ID}$ there is a fixed $R = H(\mathsf{ID})$ and there are only two possibilities for $\mathsf{sk}_{\mathsf{ID}} = r \in \{r_1, r_2\}$. For any $c = (S, b)$ output by $\mathsf{Encap}^*(\mathsf{ID})$ we claim that, if $k_1 = \mathsf{Decap}(c, r_1), k_2 = \mathsf{Decap}(c, r_2)$ then $k_1 \neq k_2$. We do this in two cases:

Case $R \in \mathrm{QR}(N)$**:** Then $k_1 = \left(\frac{f(r_1)}{N}\right)$, $k_2 = \left(\frac{f(r_2)}{N}\right)$ where f is output by $\mathcal{Q}(N, u, R, S)$. By property 3 of $\mathcal{Q}$, we have $f(r)f(-r)S \in \mathrm{QR}(N)$ for all square roots r of R. By Lemma C.2, since $S \notin \mathrm{QR}(N)$, and $r_1 \neq -r_2$, we see that $k_1 \neq k_2$.

Case $uR \in \mathrm{QR}(N)$**:** Then $k_1 = b\left(\frac{\overline{f}(r_1)}{N}\right)$, $k_2 = b\left(\frac{\overline{f}(r_2)}{N}\right)$ where $\overline{f}$ is output by $\mathcal{Q}(N, u, R, S)$. By property 4 of $\mathcal{Q}$, we have $\overline{f}(\overline{r})\overline{f}(-\overline{r})S \in \mathrm{QR}(N)$ for all square roots $\overline{r}$ of uR. By Lemma C.2, since $S \notin \mathrm{QR}(N)$, and $r_1 \neq -r_2$, we see that $\left(\frac{\overline{f}(r_1)}{N}\right) \neq \left(\frac{\overline{f}(r_2)}{N}\right)$ and so $k_1 \neq k_2$.

Therefore, we get 0-*universality* and, since r chosen uniformly from $\{r_1, r_2\}$, we also get *smoothness*. Lastly, we see that the min-entropy of r is exactly 1 bit. □

D A Construction of IB-HPS Based on Lattices

As this section will use lattice and LWE based tools we keep to the standards common in these areas. In particular we use n to denote the security parameter. We will also use the following (slight abuse of) notation. For set S we write $x \leftarrow S$ to denote sampling variable x uniformly from S. Finally for random variable y and we write $y \sim D$ to denote that y is distributed according to D.

D.1 Learning with Errors

Following [GPV08] we briefly review some important definitions and facts concerning *learning with errors* (LWE). For fixed integers n and $q = q(n)$, vector $\mathbf{s} \in \mathbb{Z}_q^n$ and error distribution $\chi = \chi(n)$ over $\mathbb{Z}_q$ define the LWE oracle $A_{\mathbf{s},\chi}$ as follows. At each invocation $A_{\mathbf{s},\chi}$ samples $\mathbf{a} \leftarrow \mathbb{Z}_q^n$ and error term $x \leftarrow \chi$ and outputs $(\mathbf{a}, \mathbf{a}^T\mathbf{s} + x)$. For $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ the decision variant ($\mathrm{DLWE}_{q,n,\chi}$) is the problem of distinguishing between oracle access to $A_{s,\chi}$ and access to an oracle which simply samples $U_{\mathbb{Z}_q^n \times \mathbb{Z}_q}$. The $\mathrm{LWE}_{q,n,\chi}$ on the other hand is defined as the problem of finding $\mathbf{s}$ when given oracle access to $A_{q,\chi}$.

Gaussian Error Distribution. To base our construction on a concrete DLWE assumption we must specify a particular error distribution. We write Ψ_α to denote the Gaussian (normal) distribution with mean 0 and variance α^2. The error distribution we are interested in (which we denote with $\overline{\Psi}_\alpha$) is called the *one-dimensional discrete Gaussian over* Z_q for some positive integer q. It can be sampled by selecting $x \leftarrow \Psi_\alpha$ and outputting $\lceil q \cdot x \rfloor \mod q$. We will also need the standard tail inequality for the continuous Gaussian. That is if $x \sim \Psi_\alpha$ and $t > 1$ then

$$\Pr[|x| > t\alpha] \leq \frac{1}{t} e^{\frac{-t^2}{2}} \tag{4}$$

Reductions to Lattice Problems. The security of our scheme is based on the hardness of average-case DLWE. Ideally we would like use the results of [Reg05, Pei09] to further reduce to some worst case lattice problem such as the GapSVP or one of it's variants. While the reduction from average-case DLWE to worst-case apply directly, the reduction from worst-case DLWE to worst-case LWE runs in time polynomial in length of the prime factors of q (ref. [Pei09], Lemma 3.3). Yet for our construction we will require $q = 2^{\omega(\log n)}$ to be prime. Thus we must make an exponential hardness assumption for the worst-case LWE problem which in turn implies a similar assumption for GapSVP. Note that for $q \geq 2^{n/2}$ such a hardness assumption then suffices for a classic reduction to the standard GapSVP problem while for smaller q [Pei09] shows a classic probabilistic poly-time reduction to the ζ-to-γ-GapSVP variant.

D.2 Preimage Sampleable Functions

We will use the preimage sampleable functions of [GPV08]. That is the set of functions $f_{\mathbf{A}}$, indexed by matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, which map vector $\mathbf{x} \in \mathbb{Z}_q^m$ to $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \in \mathbb{Z}_q^n$. Further the trapdoor sampling algorithms of [Ajt99, AP09] efficiently generate an (almost) uniform $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ together with a trapdoor $\mathbf{T}$ which is used to invert $f_{\mathbf{A}}$.

Of particular interest to us is the preimage distribution $\mathbf{e} \sim D_{m,r}$ (over $\mathbb{Z}^m$) which the authors of [GPV08] describe in detail. For the purpose of our construction we will require a special case of a Lemma 4.4 of [MR04b] which bounds the length of $\mathbf{e} \leftarrow D_{m,r}$ for large enough r together with Lemma 5.3 of [GPV08] which describes a concrete bound on r for our choice of lattices (i.e. ones defined as the null space of left multiplication by a matrix $\mathbf{A}$ as defined above).

Lemma D.1 *Let $m \geq 2n \log q$, $r > \omega(\sqrt{\log m})$ and $\varepsilon > 0$. Then we have:*

$$\Pr_{\mathbf{e} \sim D_{m,r}} \left[\|\mathbf{e}\| > r\sqrt{m} \right] \leq \frac{1+\varepsilon}{1-\varepsilon} \cdot 2^{-m}$$

We will also need their efficient probabilistic algorithm $\mathsf{SampleISIS}$ which uses trapdoor $\mathbf{T}$ to sample from the preimage of $f_{\mathbf{A}}^{-1}(\mathbf{e})$ such that for random variables $\mathbf{e} \sim D_{m,r}$ and $\mathbf{y} \sim U_{\mathbb{Z}_q^n}$

$$\mathbf{SD}\left[(\mathbf{e}, f_{\mathbf{A}}(\mathbf{e})), (\mathsf{SampleISIS}(\mathbf{y}), \mathbf{y}) \right] \leq negl(n).$$

Finally we will need a lower bound on the min-entropy the output of $\mathsf{SampleISIS}$. For general lattices, Lemma 2.11 of [PR06] provides such a bound but we will use the refined version of [Vai09]. For our case it is summarized in the following lemma:

Lemma D.2 *For constant $c > 0$, fixed $\mathbf{A}$ and $\mathbf{T}$ as generated by* Setup *and fixed $\mathbf{u} \in \mathbb{Z}_q^n$ let $\mathbf{e} \leftarrow \mathsf{SampleISIS}(\mathbf{u})$. The $\mathbf{H}_\infty(\mathbf{e}) \geq m(\log(r) - \log(m^c))$.*

D.3 The Construction

We now describe a Universal Identity-Based Hash Proof System based on the DLWE which is a slight variant of the IBE scheme in [GPV08] which is in turn based on a variant of the encryption scheme of [Reg05].

Let m and n be positive integers, q be a prime. Further let χ be an LWE error distribution. Finally let $H : \{0,1\}^* \longrightarrow \mathbb{Z}_q^n$ be a hash function mapping identities to integer vectors.

$\mathsf{Setup}()$ **:** Run the trapdoor sampling algorithm of [Ajt99, AP09] to generate $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ together with a trapdoor $\mathbf{T}$. If the columns of $\mathbf{A}$ do not generate $\mathbb{Z}_q^n$ repeat. Otherwise set $\mathsf{mpk} = \mathbf{A}$ and $\mathsf{msk} = \mathbf{T}$.

$\mathsf{KeyGen}(\mathsf{ID}, \mathsf{msk})$ **:** Set $\mathbf{u} = H(\mathsf{ID})$. Use $\mathsf{msk} = \mathbf{T}$ to run $\mathsf{SampleISIS}$ sampling $\mathbf{e} \in f_{\mathbf{A}}^{-1}(\mathbf{u})$. Output $\mathsf{sk}_{\mathsf{ID}} = \mathbf{e}$.

$\mathsf{Encap}(\mathsf{ID})$ **:** Set $\mathbf{u} = H(\mathsf{ID})$. Sample $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, error vector $\mathbf{x} \leftarrow \chi^m$ and integer $v \leftarrow \mathbb{Z}_q$. Compute $\mathbf{p} = \mathbf{A}^T\mathbf{s} + \mathbf{x} \in \mathbb{Z}_q^m$. If $\left| v - \mathbf{u}^T\mathbf{s} \right| \leq \frac{q-1}{4}$ then set $k = 1$ else set $k = 0$. Output ciphertext $c = (\mathbf{p}, v)$ and key k.

$\mathsf{Encap}^*(\mathsf{ID})$ **:** Sample $\mathbf{p} \leftarrow \mathbb{Z}_q^m$ and $v \leftarrow \mathbb{Z}_q$. Output ciphertext $c = (\mathbf{p}, v)$.

$\mathsf{Decap}(c, \mathsf{sk}_{\mathsf{ID}})$ **:** Parse $c = (\mathbf{p}, v)$ and set $\mathbf{e} = \mathsf{sk}_{\mathsf{ID}}$. If $\left| v - \mathbf{e}^T\mathbf{p} \right| \leq \frac{q-1}{4}$ then output $k = 1$. Otherwise output $k = 0$.

The construction is parametrized by an $\varepsilon \in (0,1)$. The reason being that later we will use the IB-HPS to create a leakage resilient IBE which supports leakage of up to a $1-\varepsilon$ fraction of the secret key. Thus most of the settings of our parameters will depend on this ε.

The remainder of this section focuses on proving the following theorem.

Theorem D.1 *Let $\varepsilon \in (0,1)$ be a constant and n be the security parameter. Let prime $q = 2^{\omega(\log n)}$, $m \geq 2n \log q$ and let $r \geq m^{\frac{1}{\varepsilon}}$. Let $\alpha \leq \frac{1}{r\sqrt{2mn}}$. Then, under the DLWE$_{q,\overline{\Psi}_\alpha}$ assumption, the following holds in the Random Oracle model for the above construction:*

1. *It is an (m^*, ρ)-Universal IB-HPS for $m^* = \frac{1-\varepsilon^2}{\varepsilon} m \log(m)$ and for $\rho = \frac{1}{2} + \frac{1}{2q^2}$.*
2. *The actual key size (number of bits needed to represent $\mathsf{sk}_{\mathsf{ID}}$) is $\hat{m} = \frac{1+\varepsilon}{\varepsilon} m \log(m)$.*
3. *As a consequence $\frac{m^*}{\hat{m}} \geq 1 - \varepsilon$.*

Proof. The non-trivial part of the theorem is the first statement. It's proof has been broken up into three lemmata; one for each of the three following three properties of a Universal IB-HPS: correctness, indistinguishability and universality.

Lemma D.3 (Correctness) *For the choice of parameters above, the construction is* correct.

Proof. The crux of showing correctness is captured by the following claim. Intuitively it tells us that the distance between $\mathbf{u}^T\mathbf{s}$ which is used to compute the value of k during encapsulation is very close to the value of $\mathbf{e}^T\mathbf{p}$ which is used to guess k during decapsulation. Indeed, using this claim we can then argue that the error this discrepancy introduces is small enough to only have a negligible probability of causing decapsulation to fail at guessing k correctly.

Claim D.1 *For honestly generated parameters, secret key and encapsulation in the above scheme there exists a polynomial p such that $\left|\mathbf{e}^T\mathbf{p} - \mathbf{u}^T\mathbf{s}\right| \geq p(n)$ with at most negligible probability.*

Proof. Essentially we follow the proof of Lemma 8.2 in [GPV08] however for a different choice of parameters which results in a distance negligible in n. We first note that $\mathbf{e}^T\mathbf{p} - \mathbf{u}^T\mathbf{s} = \mathbf{e}^T(\mathbf{A}^T\mathbf{s} + \mathbf{x}) - \mathbf{u}^T\mathbf{s} = \mathbf{e}^T\mathbf{x}$. Thus we are interested in bounding the length of $\mathbf{e}^T\mathbf{x}$.

By definition $\chi = \overline{\Psi}_\alpha$ and so $x_i = \lceil q \cdot y_i \rfloor \mod q$ where $y_i \leftarrow \Psi_\alpha$ independently of all y_j for $j \neq i$. Thus $||\mathbf{x} - \mathbf{y}|| \leq \sqrt{m}/2$ and we get

$$\begin{aligned} |\mathbf{e}^T\mathbf{x}| &= |\langle \mathbf{e}, \mathbf{x} \rangle| \\ &= |\langle \mathbf{e}, \mathbf{y} + (\mathbf{x} - \mathbf{y}) \rangle| \\ &\leq |\langle \mathbf{e}, \mathbf{y} \rangle| + |\langle \mathbf{e}, \mathbf{x} - \mathbf{y} \rangle| \\ &\leq |\langle \mathbf{e}, \mathbf{y} \rangle| + \sqrt{m}/2 \cdot r\sqrt{m} \\ &= |\langle \mathbf{e}, \mathbf{y} \rangle| + (rm)/2 \end{aligned}$$

where the 4-th line follows from Cauchy-Schwarz inequality.

It remains to estimate the size $\mathbf{e}^T\mathbf{y}$. The components of $\mathbf{y}$ are independently normally distributed therefor $(\mathbf{y}^T\mathbf{e}) \sim \Psi_{||\mathbf{e}||\alpha}$. By Lemma D.1 we have $||\mathbf{e}||\alpha \leq r\sqrt{m} \cdot \alpha \leq \frac{1}{\sqrt{2n}}$ with overwhelming probability over the choice of $\mathbf{e}$ and for our constraint on α. Then the standard tail inequality (equation 4) with $t = \sqrt{2n}$ implies that $\Pr\left[||\mathbf{e}^T\mathbf{y}|| > 1\right] \leq \mathrm{negl}(n)$. □

For correctness it remains to show that with at most negligible probability Decap will output a bad guess for k. This happens if and only if exactly one of the values $\mathbf{e}^T\mathbf{p}$ and $\mathbf{u}^T\mathbf{s}$ is further then $\frac{q-1}{4}$ from v. Let $d = \left|\mathbf{e}^T\mathbf{p} - \mathbf{u}^T\mathbf{s}\right|$, then for fixed $\mathbf{e}, \mathbf{p}, \mathbf{u}$ and $\mathbf{s}$ there are $2d$ values of v such that Decap produces a bad guess for k. By Claim D.1 we have that d is a polynomial in n and so since $q = 2^{\omega(\log n)}$ the probability $2d/q$ that an independent $v \leftarrow \mathbb{Z}_q$ takes on one of those values in a negligible function in n. □

Lemma D.4 (Indistinguishability) *For our choice of parameters, the construction satisfies* valid/invalid ciphertext indistinguishability.

Proof. We need to show that for fixed $\mathbf{e}$ the ciphertexts output by Encap and Encap^* have indistinguishable distributions. We do this by reducing to the $\text{DLWE}_{q,n,\chi}$ assumption. That is we give a black-box construction of an efficient DLWE adversary $\mathcal{B}$ from any distinguishability adversary $\mathcal{A}$ such that $\mathcal{B}$'s advantage in the DLWE game is negligibly close to $\mathcal{A}$'s advantage at distinguishing outputs of Encap from those of Encap^*.

DLWE attacker $\mathcal{B}$ is given access to $\mathcal{A}$ (which expects to play the indistinguishability attack game) and to an oracle $\mathcal{O}$ which returns elements $(\mathbf{a}, b) \in \mathbb{Z}_q^m \times Z_q$. $\mathcal{B}$'s goal is to decide whether $\mathcal{O}$ is an LWE oracle or not. We now describe $\mathcal{B}$'s behavior during each of the steps of the indistinguishability attack game.

Setup: $\mathcal{B}$ makes m queries to $\mathcal{O}$ receiving $\{(\mathbf{a}_i, b_i)\}_{i\in[m]}$. It sets the i-th column of matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ to $\mathbf{a}_i$ and gives $\mathsf{mpk} = \mathbf{A}$ to $\mathcal{A}$.

Test Stage 1: $\mathcal{B}$ initializes an empty table of triples of the form $(\mathsf{ID}, \mathbf{u}, \mathbf{e}) \in \{0,1\}^* \times \mathbb{Z}_q^n \times \mathbb{Z}_q^m$. For each extraction query ID made by $\mathcal{A}$, $\mathcal{B}$ first checks whether a triple of the form $(\mathsf{ID}, \cdot, \cdot)$ is already in the table. If so then it returns $\mathsf{sk}_{\mathsf{ID}} = \mathbf{u}$ for the corresponding value of $\mathbf{u}$ in the triple. If not then $\mathcal{B}$ selects a fresh $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,r}$, stores $(\mathsf{ID}, \mathbf{Ae}, \mathbf{e})$ and returns $\mathsf{sk}_{\mathsf{ID}} = \mathbf{e}$ to $\mathcal{B}$.

Random Oracle Calls: Upon receiving call $H(\mathsf{ID})$, $\mathcal{B}$ checks if it has already stored a triple for ID. If so it returns the corresponding $\mathbf{u}$. Otherwise it samples a fresh $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,r}$, stores $(\mathsf{ID}, \mathbf{Ae}, \mathbf{e})$ and returns $\mathbf{Ae}$ as a response to $\mathcal{A}$.

Challenge Stage: $\mathcal{B}$ returns the string $c = b_1 b_2 \cdots b_m$ as a challenge ciphertext.

Test Stage 2: $\mathcal{B}$ answers key extraction and random oracle queries as in Test Stage 1.

Output: $\mathcal{B}$ receives b' from $\mathcal{A}$ and forwards it the the DLWE challenger.

It remains to argue the correctness of $\mathcal{B}$. First we point out that the view of $\mathcal{A}$ is identical to that of the real indistinguishability attack game. This follows from the fact that the distribution of $\mathbf{A}$ in the real game is statistically close to uniform while both types of oracles $\mathcal{O}$ output truly uniform $\mathbf{A}$. Further the joint distributions of $(H, \mathsf{ID}, \mathbf{e})$ are indistinguishable by the correctness of the $\mathsf{SampleISIS}$ algorithm[10].

Next we analyze the distribution of challenge ciphertext c in the game run by $\mathcal{B}$. When $\mathcal{O} = A_{q,\chi}$ then c is distributed exactly like the output of $\mathsf{Encap}(\mathsf{ID}^*)$. On the other hand, when $\mathcal{O}$ samples $U_{\mathbb{Z}_q^n \times \mathbb{Z}_q}$, then c is distributed exactly like the output of $\mathsf{Encap}^*(\mathsf{ID}^*)$. Therefor the game run by $\mathcal{B}$ essentially imitates the game $\mathcal{A}$ expects to play and the bit b' output by $\mathcal{B}$ depends on which type of oracle $\mathcal{A}$ has access to. □

Lemma D.5 (Universality) *For the choice of parameters above the construction is (m^*, ρ)-universal for $m^* = \frac{1-\varepsilon^2}{\varepsilon} m \log(m)$ and $\rho = \frac{1}{2} + \frac{1}{2q^2}$.*

Proof. To show (m^*, ρ)-universality we need two properties. First, for fixed $\mathbf{A}$, H[11] and $\mathbf{u} = H(\mathsf{ID})$ we need to show that the min-entropy of $\mathbf{e}$ as output by $\mathsf{KeyGen}(\mathsf{ID}, \mathbf{A})$ is at least $m^* \geq \frac{1-\varepsilon^2}{\varepsilon} m \log(m)$. Set $c = \varepsilon$ and $r \geq m^{\frac{1}{\varepsilon}}$ then the result follows directly from Lemma D.2.

To show the second property fix ID and $\mathbf{e} \neq \mathbf{e}'$ with $H(\mathsf{ID}) = \mathbf{Ae} = \mathbf{Ae}'$. We wish to compute

$$\rho = \Pr_{c \leftarrow \mathsf{Encap}^*(\mathsf{ID})} \left[\mathsf{Decap}(c, \mathbf{e}) = \mathsf{Decap}(c, \mathbf{e}') \right]$$

This is done in two steps. Recall that $c = (\mathbf{p}, v)$. For random variable $\mathbf{p}$ define variable $D = \left| \mathbf{e}^T \mathbf{p} - \mathbf{e}'^T \mathbf{p} \right|$. First we show that $D \sim U_{\mathbb{Z}_q}$. Then we use that fact to explicitly calculate the collision probability ρ of Decap.

Let $i \in [m]$ be the index of a component where $\mathbf{e}$ and $\mathbf{e}'$ differ. Fix all p_j with $j \neq i$ and let

$$a = \left| \sum_{j \neq i} (e_j - e'_j) p_j \right|$$

[10] In particular the upper-bound on $||\mathbf{T}||$ from [AP09] implies that our choice of r is enough to satisfy the conditions of $\mathsf{SampleISIS}$.

[11] Note that for the proof of Universality we do not need to model H as a Random Oracle.

Now as q is prime addition (of a) in $\mathbb{Z}_q$ is a bijection and multiplication by the non-zero value $(e_i - e'_i)$ is also a bijection in $\mathbb{Z}_q$. Thus there is a bijection between values taken by $p_i \sim U_{\mathbb{Z}_q}$ and D implying $D \sim U_{\mathbb{Z}_q}$.

We are now ready to calculate the value of ρ. A collision occurs if and only if exactly one of the two quantities $\mathbf{e}^T\mathbf{p}$ and $\mathbf{e}'^T\mathbf{p}$ is more then $\frac{q-1}{4}$ from v. Let ρ_d be the collision probability for a given distance $D = d$, then for $d \in [(q-1)/2]$ we have $\rho_d = 1 - (2d)/q$ because $v \leftarrow \mathbb{Z}_q$ is independent of $\mathbf{e}, \mathbf{e}'$ and $\mathbf{p}$. Then we can have:

$$\begin{aligned}
\rho &= \sum_{d=0}^{q-1} \rho_d \cdot \Pr[D = d] = \frac{1}{q} + 2\sum_{d=1}^{(q-1)/2} \rho_d \cdot \Pr[D = d] = \frac{1}{q} + \frac{2}{q}\sum_{d=1}^{(q-1)/2} \rho_d \\
&= \frac{1}{q} + \frac{2}{q}\sum_{d=1}^{(q-1)/2} 1 - \frac{2d}{q} = \frac{1}{q} + \frac{2}{q}\left(\frac{q-1}{2} - \frac{2}{q}\sum_{d=1}^{(q-1)/2} d\right) \\
&= \frac{1}{2} + \frac{1}{2q^2}
\end{aligned}$$

□

Taken together the previous three lemmata conclude the proof of the first statement of Theorem D.1.

The second statement follows directly form Lemma D.1. If the norm of $\mathbf{e}$ is bounded by $r\sqrt{m}$ (with overwhelming probability) then so too of course, are it's components. Thus $\log(r) + \log(m)$ bits suffice to represent each component of which there are m. In particular for $r \geq m^{\frac{1}{\varepsilon}}$ no more then $\hat{m} = \frac{1+\varepsilon}{\varepsilon} m \log(m)$ bits are needed.

Finally the third statement of the theorem follows from the calculation:

$$\frac{m^*}{\hat{m}} = \frac{\frac{1-\varepsilon^2}{\varepsilon} m \log(m)}{\frac{1+\varepsilon}{\varepsilon} m \log(m)} = \frac{(1-\varepsilon)(1+\varepsilon)}{1+\varepsilon} = 1 - \varepsilon$$

□

E Approximate Hashing and Approximate Leftover-Hash Lemma

E.1 Background.

First, we review several standard notions which we will need in the remainder of the section.

Definition E.1 *The q-ary Shannon entropy function is defined as $H_q(x) \stackrel{def}{=} x\log_q(q-1) - x\log_q(x) - (1-x)\log_q(1-x)$.*

Lemma E.1 (Volume of Hamming Ball) *Let $x \in \Sigma^n$ be an arbitrary value, where Σ is some alphabet of size $|\Sigma| = q$, and let δ be an arbitrary value in the range $1/n < \delta \leq 1 - 1/q$. Define $V_q(n, \delta, x) \stackrel{def}{=} |\{x' \in \Sigma^n : d_H(x, x') \leq \delta n\}|$ to be the volume of the hamming ball of radius δn centered at x. Then there is a function $V_q(n, \delta)$ such that $V_q(n, \delta) = V_q(n, \delta, x)$ for all $x \in \Sigma^n$, and $V_q(n, \delta) \leq q^{H_q(\delta)n}$.*

See e.g. [Sho05] for the following Lemma.

Lemma E.2 (Collision Probability and Statistical Distance) *Suppose X is a random variable that takes values from a set Π of size $|\Pi| = p$. We define the* collision probability *$\beta \stackrel{def}{=} \Pr[x = x']$ where x, x' are independently sampled according to X. Then $\mathbf{SD}(X, U_\Pi) \leq \frac{1}{2}\sqrt{p\beta - 1}$.*

Definition E.2 (Hitter) *Let $\mathsf{Hit} : \{0,1\}^w \to [n]^t$ be a function and interpret the output $\mathsf{Hit}(e)$ as a sample of t elements in $[n]$. We say that $\mathsf{Hit}(e)$ hits $S \subseteq [n]$ if it includes at least one member of S. A function Hit is a (δ, ψ)-*hitter *if for every subset $S \subseteq [n]$ of size $|S| \geq \delta n$, $\Pr_{e \leftarrow U_w}[\mathsf{Hit}(e)$ hits $S] \geq (1 - \psi)$.*

A simple hitter construction involves choosing t uniformly random and independent elements of $[n]$. This results in a (δ, ψ)-hitter with $\psi = (\delta)^t$ for any $0 < \delta < 1$. Alternatively, for any $0 < \delta < 1, 0 < \psi$ we get a (δ, ψ)-hitter with *sample complexity* $t = \mathcal{O}(\log(1/\psi)/\delta)$ and *randomness complexity* $w = t\log(n)$. Interestingly, the randomness complexity can be reduced significantly by using a more clever construction. Indeed, the survey of Goldreich [Gol97] shows how to achieve the following parameters using a construction based on expander graphs.

Theorem E.1 ([Gol97]) *There exists an efficient ensemble of hitters* $\mathsf{Hit} : \{0,1\}^w \to [n]^t$ *such that, for any integer* n *and any* δ, ψ *with* $0 < \delta < 1$, $0 < \psi$*, we get sample complexity* $t = \mathcal{O}(\log(1/\psi)/\delta)$ *and randomness complexity* $w = \log(n) + 3\log(1/\psi)$.

E.2 Definition and Results.

We define a new notion of universal hashing, which relaxes ρ-universality, by *only* insisting that values which are far from each other (over the Hamming metric) are unlikely to collide.

Definition E.3 (Approximately Universal Hashing) *A function-family* $\mathcal{H}$*, consisting of functions* $h : \Sigma^n \to \Pi$*, is called* (δ, τ)-approximately *universal if for all* $x, x' \in \Sigma^n$ *with* $d_H(x, x') \geq \delta n$ *we have* $\Pr_{h \leftarrow \mathcal{H}}[h(x) = h(x')] \leq \tau$.

Now we are ready to prove a generalized version of the leftover-hash lemma for approximate universal hashing.

Theorem E.2 (Approximate Leftover-Hash Lemma) *Assume that a family* $\mathcal{F}$ *of functions* $f : \Sigma^n \to \Pi$ *is* (δ, τ)-*approximately universal. Define* $q \stackrel{def}{=} |\Sigma|$, $v \stackrel{def}{=} \log(|\Pi|)$. *Let* δ *be in the range* $1/n \leq \delta \leq (1 - \frac{1}{q})$. *Let* X, Z *be arbitrary random variables where* X *is distributed over* Σ^n *and* $m \stackrel{def}{=} \widetilde{\mathbf{H}}_\infty(X|Z)$. *Let* F *be uniform over* $\mathcal{F}$. *Then*

$$\mathbf{SD}(\ (F, Z, F(X))\ ,\ (F, Z, U_\Pi)\ ) \leq \frac{1}{2}\sqrt{2^{H_q(\delta) n \log(q) + v - m} + \tau 2^v - 1}$$

where H_q *is* q*-ary Shannon entropy function. In particular, the statistical distance above is at most* ε *as long as:*

$$m \geq H_q(\delta) n \log(q) + v + 2\log(1/\varepsilon) - 1 \quad \textit{and} \quad \tau \leq \frac{1}{2^v}(1 + \varepsilon^2).$$

Proof. For each value z in the support of Z, define the random variable $X_z = (X \mid Z = z)$. We start by computing the collision probability $\beta \stackrel{\text{def}}{=} \Pr[(f, f(x)) = (f', f'(x'))]$ where f, f' are independently sampled from F and x, x' are independently sampled according to X_z. Then

$$\begin{aligned}
\beta &= \Pr[(f, f(x)) = (f', f'(x'))] = \Pr[f = f'] \Pr[f(x) = f(x')] \\
&\leq \frac{1}{|\mathcal{H}|}\left(\Pr[d_H(x, x') < \delta n] + \tau\right) \\
&\leq \frac{1}{|\mathcal{H}|}\left(\frac{V_q(n, \delta)}{2^{\mathbf{H}_\infty(X_z)}} + \tau\right) \leq \frac{1}{|\mathcal{H}|}\left(\frac{q^{H_q(\delta) n}}{2^{\mathbf{H}_\infty(X_z)}} + \tau\right) \qquad (5) \\
&\leq \frac{1}{|\mathcal{H}| 2^v}\left(2^{H_q(\delta) n \log(q) + v - \mathbf{H}_\infty(X_z)} + \tau 2^v\right)
\end{aligned}$$

where (5) follows by Lemma E.1. We now apply Lemma E.2 to the random variable $(F, F(X_z))$, which gives us:

$$\mathbf{SD}(\ (F, F(X_z))\ ,\ (F, U_\Pi)\ ) \leq \frac{1}{2}\sqrt{2^{H_q(\delta) n \log(q) + v - \mathbf{H}_\infty(X_z)} + \tau 2^v - 1}$$

Now, by averaging over $z \leftarrow Z$, we get:

$$\begin{aligned}
\mathbf{SD}(\ (F, Z, F(X))\ ,\ (F, Z, U_\Pi)\ ) &= \mathbb{E}_z[\ \mathbf{SD}(\ (F, F(X_z))\ ,\ (F, U_\Pi)\ )] \\
&\leq \mathbb{E}_z\left[\frac{1}{2}\sqrt{2^{H_q(\delta) n \log(q) + v - \mathbf{H}_\infty(X_z)} + \tau 2^v - 1}\right] \\
&\leq \frac{1}{2}\sqrt{\mathbb{E}_z\left[2^{H_q(\delta) n \log(q) + v - \mathbf{H}_\infty(X_z)}\right] + \tau 2^v - 1} \\
&= \frac{1}{2}\sqrt{2^{H_q(\delta) n \log(q) + v - \widetilde{\mathbf{H}}_\infty(X|Z)} + \tau 2^v - 1} \\
&= \frac{1}{2}\sqrt{2^{H_q(\delta) n \log(q) + v - m} + \tau 2^v - 1}
\end{aligned}$$

which proves the first part of theorem. For the second part of the theorem, $\mathbf{SD}(\ (F,Z,F(X))\ ,\ (F,Z,U_{\Pi})\ ) \leq \varepsilon$ if

$$\varepsilon \ \geq \ \frac{1}{2}\sqrt{2\max\left(2^{H_q(\delta)n\log(q)+v-m}\ ,\ \tau 2^v-1\right)}$$

which is satisfied by the conditions of the second part of the theorem. □

E.3 Analysis of a Concrete Approximately Universal Function.

We now explore a concrete example of an *approximately universal hash function* with locality, which will be used in our construction PKE in the BRM. Let Σ, Ψ be some alphabets, and let $\mathcal{F}$ be a family of ρ-universal hash functions $f\ :\ \Sigma \to \Psi$. For integers $n,t,v>0$ we define the family $\mathcal{H}_{(n,t,v)}$ of hash function $h\ :\ \Sigma^n \to \{0,1\}^v$ as follows:

- Each h in $\mathcal{H}_{(n,t,v)}$ is uniquely described by:
  1. A vector $\overline{r}=(r_1,\ldots,r_t)$ of (not necessarily distinct) indices $r_i \in [n]$.
  2. A vector $\overline{f}$ of functions $(f_1,\ldots,f_t)$ where $f_i \in \mathcal{F}$.
  3. A function $g \in \mathcal{G}$, where $\mathcal{G}$ is some $(1/2^v)$-universal-hash function family of functions $g\ :\ \Psi^t \to \{0,1\}^v$.

  In particular, a random h from $\mathcal{H}$ consists of a uniformly random choice of $(\overline{r},\overline{f},g) \in_R ([n]^t, \mathcal{F}^t, \mathcal{G})$.
- For $x \in \Sigma^n$, we define $h(x) = g(\ (\ f_1(x[r_1]),\ldots,f_t(x[r_t])\ )\ )$.

The family $\mathcal{H}_{(n,t,v)}$ will be useful as it will form the backbone of our construction of PKE in the BRM. Note that, although the above definition of $\mathcal{H}$ might appear overly complicated and unnatural, it arises from our need to work with an existing IB-HPS (which, in turn, is delicately designed based on some underlying computational assumptions) and thus we do not, in general, have the freedom to choose all the components of $\mathcal{H}$. In particular, the alphabets Σ, Ψ and the function family $\mathcal{F}$ will be a part of the underlying IB-HPS and thus not in our control, while we will have the freedom to choose n,t,v.

Lemma E.3 *Let Σ, Ψ be alphabets, and let $\mathcal{F}$ be a family of ρ-universal hash functions $f\ :\ \Sigma \to \Psi$. For integers $n,t,v>0$ let $\mathcal{H}_{(n,t,v)}$ be the family of functions $h\ :\ \Sigma^n \to \{0,1\}^v$ as defined above. Then the family $\mathcal{H}$ is (δ,τ)-approximately universal for any $\delta>0$ with $\tau \leq (1-\delta(1-\rho))^t + 1/2^v$.*

Proof. For any $x,x' \in \Sigma^n$ where $d_H(x,x') \geq \delta n$, we have

$$\begin{aligned}
\Pr_{h\leftarrow\mathcal{H}}[h(x)=h(x')] &\leq \Pr_{h\leftarrow\mathcal{H}}[\ (\ f_1(x[r_1]),\ldots,f_t(x[r_t]\ )\ =\ (\ f_1(x'[r_1]),\ldots,f_t(x'[r_t]\ )\ ] + 1/2^v \\
&\leq \sum_{i=0}^{t}\left(\Pr[d_H(\ (x[r_1],\ldots,x[r_t])\ ,\ (x'[r_1],\ldots,x'[r't])\ )=i]\rho^i\right) + 1/2^v \\
&\leq \sum_{i=0}^{t}\left[\binom{t}{i}\delta^i(1-\delta)^{t-i}\rho^i\right] + 1/2^v \\
&\leq (1-\delta(1-\rho))^t + 1/2^v
\end{aligned}$$

□

Corollary E.1 *Let Σ, Ψ be alphabets where $|\Sigma|=q$ and $\mathcal{F}$ be a ρ-universal family of hash functions. Let $\lambda,n,t,v>0$ be integers, and $1/n<\delta<1/2$. Let H be uniform over $\mathcal{H}_{(n,t,v)}$ and X,Z be arbitrary correlated random variables where X is distributed over Σ^n. Then $SD(\ (X,H,Z,H(X))\ ,\ (X,H,Z,U_v)\ ) \leq 2^{-\lambda}$ as long as:*

$$t \geq (v+2\lambda)/(\delta(1-\rho)) \quad , \quad \widetilde{\mathbf{H}}_{\infty}(X|Z) \geq H_q(\delta)n\log(q) + v + 2\lambda - 1.$$

In particular, for any constants $\varepsilon>0$ and $\rho<1$, there exists some constant $c \geq 0$, such that for any $q \geq 2$, $v \geq 1$, $t \geq c(v+\lambda)$, $n \geq 0$ the family $\mathcal{H}_{(n,t,v)}$ has the following property:

$$\textit{If} \quad \widetilde{\mathbf{H}}_{\infty}(X|Z) \geq \varepsilon n\log(q) + v + 2\lambda \quad \textit{then} \quad \mathbf{SD}(\ (X,H,Z,H(X))\ ,\ (X,H,Z,U_v)\ ) \leq 2^{-\lambda}$$

where H is uniform over $\mathcal{H}_{(n,t,v)}$.

E.4 An Alternative Approximately Universal Function (For Anonymous Encapsulation Scheme)

We also explore another example, which is used by our "short ciphertext" scheme based on "anonymous encapsulation". Let Σ, Ψ be some alphabets, and let $\mathcal{F}$ be a family of ρ-universal hash functions $f : \Sigma \to \Psi$. Let $\mathsf{Hit} : \{0,1\}^w \leftarrow [n]^t$ be a (δ, τ)-hitter and $\mathcal{G}$ be a family of γ-universal hash functions $g : \Psi^t \to \{0,1\}^v$. For integers $n, v > 0$ we define the family $\mathcal{H}^*_{(n,v)}$ of hash function $h : \Sigma^n \to \{0,1\}^v$ as follows:

- Each h in $\mathcal{H}^*_{(n,v)}$ is uniquely described by:
  1. A seed $e \in \{0,1\}^w$ for the hitter.
  2. A function $f \in \mathcal{F}$.
  3. A function $g \in \mathcal{G}$.

  In particular, a random h from $\mathcal{H}^*_{(n,v)}$ consists of a uniformly random choice of $(e, f, g) \in (\{0,1\}^w, \mathcal{F}, \mathcal{G})$.
- For $x \in \Sigma^n$, we define $h(x) = g(\ (\ f(x[r_1]), \ldots, f(x[r_t])\ )\ )$ where $(r_1, \ldots, r_t) = \mathsf{Hit}(e)$.

Lemma E.4 *Let Σ, Ψ be alphabets, and let $\mathcal{F}$ be a family of ρ-universal hash functions $f : \Sigma \to \Psi$. Let Hit be a (δ, ψ) hitter and $\mathcal{G}$ be a γ-universal hash family. For integers $n, v > 0$ let $\mathcal{H}^*_{(n,v)}$ be the family of functions $h : \Sigma^n \to \{0,1\}^v$ as defined above. Then the family $\mathcal{H}^*_{(n,v)}$ is (δ, τ)-approximately universal for any $\delta > 0$ with $\tau \leq \psi + \rho + \gamma$.*

Proof. For any $x, x' \in \Sigma^n$ where $d_H(x, x') \geq \delta n$, we have

$$\begin{aligned}
\Pr_{h \leftarrow \mathcal{H}}[h(x) = h(x')] &\leq \Pr_{h \leftarrow \mathcal{H}}[\ (\ f(x[r_1]), \ldots, f(x[r_t]\ )\ =\ (\ f(x'[r_1]), \ldots, f(x'[r_t]\ )\ ] + \gamma \\
&\leq \Pr_{h \leftarrow \mathcal{H}}[\ (\ x[r_1], \ldots, x[r_t]\ )\ =\ (\ x'[r_1], \ldots, x'[r_t]\ )\ ] + \rho + \gamma \\
&\leq \psi + \rho + \gamma
\end{aligned}$$

where $(r_1, \ldots, r_t) = \mathsf{Hit}(e)$ and $h = (e, g, f)$. □

Corollary E.2 *For any family $\mathcal{F}$ of 0-universal functions, any constant $\varepsilon > 0$, any $q \geq 2$, any polynomials $v(\lambda)$, $n(\lambda)$, there exists some some instantiation of the hitter Hit and the family $\mathcal{G}$ so that the following holds about the resulting family $\mathcal{H}^*_{(n,v)}$. For any X, Z*

$$\textit{If} \quad \widetilde{\mathbf{H}}_\infty(X|Z) \geq \varepsilon n \log(q) + v + \Omega(\lambda) \quad \textit{then} \quad \mathbf{SD}(\ (X, H, Z, H(X))\ ,\ (X, H, Z, U_v)\ ) \leq 2^{-\Omega(\lambda)}$$

where H is uniform over $\mathcal{H}^*_{(n,v)}$. Moreover, the description size of $h \in \mathcal{H}^*_{n,v}$ is $O(v+\lambda)+|f|$ where $|f|$ is the description size of $f \in \mathcal{F}$. Lastly, the locality of $h \in \mathcal{H}^*_{n,v}$ (the number of x_i accessed) is $t = O(v + \lambda)$.

F Chosen-Ciphertext Security

F.1 A Leakage-Resilient CCA-Secure IBE

Following the approach presented in Appendix Appendix B, we show that the CCA-secure variant of Gentry's IBE scheme [Gen06] can used for constructing an IBE scheme that is resilient to any leakage of length roughly $\ell = s/6$ bits, where s is the length of the secret key of each identity. We begin by providing a formal definition of a leakage-resilient IBE, and then present our construction.

F.1.1 Definition

The following definition is a natural generalization of the definition presented in Section 4. Given a security parameter λ and a leakage parameter ℓ, we define the following game between an adversary $\mathcal{A}$ and a challenger:

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

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 $\mathcal{A}$ can adaptively ask the challenger for 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 $\mathsf{ID} \in \mathcal{ID}$ and a PPT function $f : \{0,1\}^* \to \{0,1\}$, the challenger replies with $f(\mathsf{sk}_{\mathsf{ID}})$.
Decryption Queries: On input $\mathsf{ID} \in \mathcal{ID}$ and a ciphertext c, the challenger replies with $\mathsf{Decrypt}(c, \mathsf{sk}_{\mathsf{ID}})$.

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, computes $c^* \leftarrow \mathsf{Encrypt}(\mathsf{ID}^*, \mathsf{m}_b)$, and gives c^* to the adversary $\mathcal{A}$.

Test Stage 2: The adversary can adaptively submit secret-key queries with any $\mathsf{ID} \neq \mathsf{ID}^*$, and decryption queries with any $(\mathsf{ID}, c) \neq (\mathsf{ID}^*, c^*)$.

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 and 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, leakage, or decryption query) and responds to all future queries on the same ID with the same $\mathsf{sk}_{\mathsf{ID}}$.

For any adversary $\mathcal{A}$, its advantage in the above game with an identity-based encryption scheme IBE is defined as $\mathsf{Adv}^{\mathsf{IBE\text{-}SS\text{-}CCA}}_{\mathsf{IBE},\mathcal{A}}(\lambda, \ell) \stackrel{\text{def}}{=} \left|\Pr[\mathcal{A} \text{ wins }] - \frac{1}{2}\right|$. We say that an identity-based encryption scheme IBE is *ℓ-leakage-resilient under a chosen-ciphertext attack* if for any PPT adversary $\mathcal{A}$ it holds that $\mathsf{Adv}^{\mathsf{IBE\text{-}SS\text{-}CCA}}_{\mathsf{IBE},\mathcal{A}}(\lambda, \ell)$ is negligible in λ.

F.1.2 The Construction

Let $\mathbb{G}$ and $\mathbb{G}_T$ be cyclic groups of prime order p, and let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map. Let $\mathsf{Ext} : \mathbb{G}_T \times \{0,1\}^d \to \{0,1\}^m$ be an average-case $(\log p - \ell, \varepsilon)$-strong extractor for some negligible $\varepsilon = \varepsilon(\lambda)$, where $\ell \leq \log p - \omega(\log \lambda) - m$, and let $\mathcal{H} = \{H : \mathbb{G} \times \mathbb{G}_T \times \{0,1\}^d \times \{0,1\}^m \to \mathbb{Z}_p\}$ be a collection of universal one-way hash functions. The following describes an identity-based encryption scheme $\mathsf{IBE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$:

Setup: Choose random generators $g, h_1, h_2, h_3 \in \mathbb{G}$, a random $\alpha \in \mathbb{Z}_p$, and a sample a function $H \in \mathcal{H}$. Let $g_1 = g^\alpha$ and output

$$\mathsf{mpk} = (g, g_1, h_1, h_2, h_3, H), \quad \mathsf{msk} = \alpha \ .$$

Key generation: On input an identity $\mathsf{ID} \in \mathbb{Z}_p \setminus \{\alpha\}$ sample $r_{\mathsf{ID},i} \in \mathbb{Z}_p$ uniformly at random for $i \in \{1,2,3\}$, and output the secret key $\mathsf{sk}_{\mathsf{ID}} = \{(r_{\mathsf{ID},i}, h_{\mathsf{ID},i})\}_{i=1}^3$ where

$$h_{\mathsf{ID},i} = \left(h_i g^{-r_{\mathsf{ID},i}}\right)^{1/(\alpha - \mathsf{ID})} \ .$$

If $\mathsf{ID} = \alpha$ then the algorithm aborts without producing a secret key.

Encryption: On input a message $\mathsf{m} \in \{0,1\}^m$ and an identity $\mathsf{ID} \in \mathbb{Z}_p$, choose $r \in \mathbb{Z}_p$ and $s \in \{0,1\}^d$ independently and uniformly at random, and output the ciphertext $c = (u, v, s, w, y)$, where:

$$u = g_1^r g^{-r \cdot \mathsf{ID}},\ v = e(g,g)^r,\ w = \mathsf{Ext}\left(e(g,h_1)^r, s\right) \oplus \mathsf{m},\ y = e(g,h_2)^r e(g,h_3)^{r\beta} \ ,$$

and $\beta = H(u, v, s, w)$.

Decryption: On input a ciphertext (u, v, s, w, y) and a secret key $\mathsf{sk}_{\mathsf{ID}}$, if $y = e(u, h_{\mathsf{ID},2} {h_{\mathsf{ID},3}}^\beta) v^{r_{\mathsf{ID},2} + r_{\mathsf{ID},3}\beta}$ where $\beta = H(u, v, s, w)$, then output $w \oplus \mathsf{Ext}\left(e(u, h_{\mathsf{ID},1}) v^{r_{\mathsf{ID},1}}, s\right)$, and otherwise output $\perp$.

Theorem F.1 *Fix any polynomials q_{ID}, q_L, q_C, and let $q = q_{\mathsf{ID}} + q_L + 3$. Assuming the hardness of the q-TABDHE problem, $\mathsf{Adv}^{\mathsf{IBE\text{-}SS\text{-}CCA}}_{\mathsf{IBE},\mathcal{A}}(\lambda, \ell)$ is negligible for any PPT adversary $\mathcal{A}$ submitting at most q_{ID} secret-key queries, q_L leakage queries, and q_C decryption queries.*

Proof. We show that any efficient adversary $\mathcal{A}$ for which $\mathsf{Adv}^{\mathsf{IBE\text{-}SS\text{-}CCA}}_{\mathsf{IBE},\mathcal{A}}(\lambda, \ell)$ is noticeable can be used to either solve the q-TABDHE problem with a noticeable advantage or the break the security of the collection $\mathcal{H}$ of universal one-way

hash functions. Let $\mathcal{D}$ be a distinguisher for the q-TABDHE problem that receives as input a challenge of the form $\left(g', g'_{q+2}, g, g_1, \ldots, g_q, Z\right)$ (recall that $g_i = g^{(\alpha^i)}$, $g'_{q+2} = g'^{(\alpha^{q+2})}$, and that Z is either $e(g_{q+1}, g')$ or a random element of $\mathbb{G}_T$), and simulates the game $\mathsf{IBE\text{-}SS\text{-}CCA}(\lambda, \ell)$ to the adversary $\mathcal{A}$ as follows:

- **Setup:** For $i \in \{1,2,3\}$ the distinguisher $\mathcal{D}$ generates a random polynomial $f_i(x) \in \mathbb{Z}_p[x]$ of degree q, and sets $h_i = g^{f_i(\alpha)}$ (note that h_i is efficiently computable from $g, g_1, \ldots, g_q$). In addition, $\mathcal{D}$ samples $H \in \mathcal{H}$, and outputs $\mathsf{mpk} = (g, g_1, h_1, h_2, h_3, H)$.

- **Secret-key queries:** If $\mathcal{A}$ ever submits a secret-key query with $\mathsf{ID} = \alpha$ then $\mathcal{D}$ solves the q-TABDHE problem. Otherwise, whenever $\mathcal{A}$ submits a secret-key query with $\mathsf{ID} \neq \alpha$, for every $i \in \{1,2,3\}$ let $F_{\mathsf{ID},i}(x)$ denote the $(q-1)$-degree polynomial $(f_i(x) - f_i(\mathsf{ID}))/(x - \mathsf{ID})$, and $\mathcal{D}$ outputs $\mathsf{sk}_{\mathsf{ID}} = \{(r_{\mathsf{ID},i}, h_{\mathsf{ID},i})\}_{i=1}^3$ where

$$r_{\mathsf{ID},i} = f_i(\mathsf{ID}), \quad h_{\mathsf{ID},i} = g^{F_{\mathsf{ID},i}(\alpha)} \ .$$

- **Leakage queries:** If $\mathcal{A}$ ever submits a leakage query with $\mathsf{ID} = \alpha$ then $\mathcal{D}$ solves the q-TABDHE problem. Otherwise, whenever $\mathcal{A}$ submits a leakage query with $\mathsf{ID} \neq \alpha$ then $\mathcal{D}$ computes $\mathsf{sk}_{\mathsf{ID}}$ as in the simulation of secret-key queries, and outputs the value of the given leakage function when applied to $\mathsf{sk}_{\mathsf{ID}}$.

- **Decryption queries:** If $\mathcal{A}$ ever submits a decryption query (ID, c) with $\mathsf{ID} = \alpha$ then $\mathcal{D}$ solves the q-TABDHE problem. Otherwise, whenever $\mathcal{A}$ submits a decryption query with $\mathsf{ID} \neq \alpha$ then $\mathcal{D}$ computes $\mathsf{sk}_{\mathsf{ID}}$ as in the simulation of secret-key queries, and outputs the result of the decryption algorithm applied to $\mathsf{sk}_{\mathsf{ID}}$ and c.

- **Challenge stage:** If $\mathcal{A}$ submits $(\mathsf{ID}, \mathsf{m}_0, \mathsf{m}_1)$ such that $\mathsf{ID} = \alpha$ then $\mathcal{D}$ solves the q-TABDHE problem. Otherwise, $\mathcal{D}$ chooses $b \in \{0,1\}$ uniformly at random, and computes $\mathsf{sk}_{\mathsf{ID}}$ as in the simulation of secret-key queries. Let $\bar{f}(x) = x^{q+2}$, $\bar{F}_{\mathsf{ID}}(x) = (\bar{f}(x) - \bar{f}(\mathsf{ID}))/(x - \mathsf{ID})$. Then, $\mathcal{D}$ chooses $s \in \{0,1\}^d$ uniformly at random, and computes

$$\begin{aligned} u &= g'^{\bar{f}(\alpha) - \bar{f}(\mathsf{ID})} \\ v &= Z \cdot e\left(g', \prod_{i=0}^{q} g^{\bar{F}_{\mathsf{ID},i}\alpha^i}\right) \\ w &= M_b \oplus \mathsf{Ext}\left(e(u, h_{\mathsf{ID},1})v^{r_{\mathsf{ID},1}}, s\right) \\ y &= e(u, {h_{\mathsf{ID},2}h_{\mathsf{ID},3}}^{\beta})v^{r_{\mathsf{ID},2} + r_{\mathsf{ID},3}\beta} \ , \end{aligned}$$

  where $\bar{F}_{\mathsf{ID},i}$ is the coefficient of x^i in $\bar{F}_{\mathsf{ID}}(x)$, and $\beta = H(u, v, s, w)$. Finally, $\mathcal{D}$ outputs the challenge ciphertext (u, v, s, w, y).

- **Output:** If $\mathcal{A}$ outputs b' such that $b' = b$ then $\mathcal{D}$ outputs 1, and otherwise $\mathcal{D}$ outputs 0.

In the remainder of the proof we say that a ciphertext (u, v, s, w, y) is *well-formed* for identity ID if it holds that $y = e(u, {h_{\mathsf{ID},2}h_{\mathsf{ID},3}}^{\beta})v^{r_{\mathsf{ID},2} + r_{\mathsf{ID},3}\beta}$, where $\beta = H(u, v, s, w)$. Note that by the definition of the decryption algorithm, it accepts a ciphertext if and only if it is a well-formed ciphertext. In addition, we say that a ciphertext (u, v, s, w, y) is *valid* for identity ID if it holds that $v = e(u, g)^{1/(\alpha - \mathsf{ID})}$, and otherwise we say that it is *invalid*.

Without loss of generality we assume that if the adversary submits more than ℓ leakage queries with the same identity, then in query $\ell + 1$ he is given the secret key of this identity. That is, we replace query $\ell + 1$ with a secret-key query. This assumption is valid since such an identity cannot be chosen as the challenge identity, and therefore the adversary can might as well ask for the corresponding secret key. In addition, we assume that the adversary never submits a decryption query with an identity to which he already knows the secret key. This assumption is valid since the challenger in the $\mathsf{IBE\text{-}SS\text{-}CCA}(\lambda, \ell)$ game simply invokes the decryption algorithm, and this can be simulated internally by the adversary.

The proof consists of two main arguments. First, we prove that if $Z = e(g_{q+1}, g')$ then $\mathcal{A}$'s view in the simulated attack (i.e., in the interaction with $\mathcal{D}$) is statistically-close to $\mathcal{A}$'s view in the actual attack (i.e., in the $\mathsf{IBE\text{-}SS\text{-}CCA}(\lambda, \ell)$ game). Then, we prove that if Z is random then $\mathcal{A}$ has only a negligible advantage in outputting the bit b. These two arguments are proved in Lemmata F.1 and F.2, respectively, and conclude the proof of the theorem.

Lemma F.1 *If $Z = e(g_{q+1}, g')$ then $\mathcal{A}$'s view in the simulated attack is statistically-close to $\mathcal{A}$'s view in the actual attack.*

Proof. Assuming that in both the simulated attack and the actual attack all decryption queries with invalid ciphertexts are rejected, the views of the adversary are identical in both cases. This follows from the fact that the adversary learns the value of f_1, f_2, and f_3 on at most $q - 1 = q_{\mathsf{ID}} + q_L + 1$ points (these include q_{ID} secret-key queries, q_L leakage queries, the point α, and the challenge identity), and that decryption queries with valid ciphertext reveal no information on the secret key that is used to decrypt the ciphertext. Therefore, the fact that the polynomials f_1, f_2, and f_3 are of degree q implies that the above $q - 1$ values are independent and uniformly distributed from the adversary's point of view (note that here it in fact suffices that these polynomials are of degree $q-2$, but we will need them to be of degree q to argue that invalid ciphertexts are rejected). In the following claim we argue that in the simulation all invalid ciphertext are rejected with overwhelming probability. A similar and much simpler claim holds for the actual attack as well (see [Gen06]).

Claim F.1 *If $Z = e(g_{q+1}, g')$ then the decryption algorithm rejects all invalid ciphertexts, except with a negligible probability.*

Proof. We bound the probability that the adversary submits a decryption query with an invalid ciphertext and this query is accepted by the decryption algorithm (i.e., the ciphertext is well-formed). We analyze this probably by considering the joint distribution of the coefficients of the polynomials f_2 and f_3 from the adversary's point of view. Denote by $(\mathsf{ID}, (u, v, s, w, y))$ the first decryption query submitted by $\mathcal{A}$ with an invalid ciphertext. Denote by $sk_{\mathsf{ID}} = \{(r_{\mathsf{ID},i}, h_{\mathsf{ID},i})\}_{i=1}^{3}$ the secret key for ID as computed by $\mathcal{D}$ when answering this decryption query. In order for the ciphertext (u, v, s, w, y) to be accepted by the decryption algorithm it must hold that $y = e(u, h_{\mathsf{ID},2} {h_{\mathsf{ID},3}}^{\beta}) v^{r_{\mathsf{ID},2} + r_{\mathsf{ID},3}\beta}$, where $\beta = H(u, v, s, w)$. By letting $a_u = \log_g u$, $a_v = \log_{e(g,g)} v$, and $a_y = \log_{e(g,g)} y$, this condition can be written as

$$a_y = a_u \left(\log_g h_{\mathsf{ID},2} + \beta \log_g h_{\mathsf{ID},3} \right) + a_v \left(r_{\mathsf{ID},2} + \beta r_{\mathsf{ID},3} \right) \quad . \tag{6}$$

In addition, from the public parameters we obtain the following equations:

$$\log_g h_1 = (\alpha - \mathsf{ID}) \log_g h_{\mathsf{ID},1} + r_{\mathsf{ID},1} \tag{7}$$

$$\log_g h_2 = (\alpha - \mathsf{ID}) \log_g h_{\mathsf{ID},2} + r_{\mathsf{ID},2} \tag{8}$$

$$\log_g h_3 = (\alpha - \mathsf{ID}) \log_g h_{\mathsf{ID},3} + r_{\mathsf{ID},3} \quad . \tag{9}$$

Combining Equations (6), (8), and (9), in order for the ciphertext to be accepted the adversary $\mathcal{A}$ has to compute y such that

$$a_y = \frac{a_u}{\alpha - \mathsf{ID}} \cdot \left(\log_g h_2 + \beta \log_g h_3 \right) + \left(a_v - \frac{a_u}{\alpha - \mathsf{ID}} \right) \cdot \left(r_{\mathsf{ID},2} + \beta r_{\mathsf{ID},3} \right) \quad . \tag{10}$$

Up to this point the view of the adversary contains the public parameters (which we already took into consideration in Equations (7), (8), and (9)), the result of at most q_{ID} secret-key queries and q_L key-leakage queries, the result of decryption queries with valid ciphertexts (these do not reveal any more information on f_2 and f_3), and possibly also the challenge ciphertext. For the sake of this proof we can even assume that the adversary actually obtains all the secret keys for which it requested leakage information, the secret key of the challenge identity, and an additional ℓ bits of leakage on the secret key of ID. Ignoring these ℓ bits of leakage for now, this means that the adversary knows the values of f_2 and f_3 at the point α (this is from h_2 and h_3), and at no more than $q_{\mathsf{ID}} + q_L + 1 = q - 2$ distinct identities that we denote by $x_1, \ldots, x_{q-2}$. Letting $f_i(x) = \sum_{j=0}^{q} f_{i,j} x^j$ for $i \in \{2, 3\}$, and $x_{q-1} = \alpha$, the knowledge of the adversary

can be represented by the following product:

$$\begin{pmatrix} f_{2,0} & \cdots & f_{2,q} & f_{3,0} & \cdots & f_{3,q} \end{pmatrix} \begin{pmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 \\ x_1 & \cdots & x_{q-1} & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x_1^q & \cdots & x_{q-1}^q & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1 & \cdots & 1 \\ 0 & \cdots & 0 & x_1 & \cdots & x_{q-1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & x_1^q & \cdots & x_{q-1}^q \end{pmatrix} \tag{11}$$

Let $\boldsymbol{f}$ denote the vector on the left, and let V denote the matrix on the right. Note that V contains two $(q+1) \times (q-1)$ Vandermonde matrices and its columns are linearly independent. Therefore, from $\mathcal{A}$'s view, since V has four more rows than columns, the solution space for $\boldsymbol{f}$ is four-dimensional.

Let γ_{ID} denote the vector $(1, \mathsf{ID}, \ldots, \mathsf{ID}^q)$, then Equation (10) can be re-written as follows:

$$a_y = \frac{a_u}{\alpha - \mathsf{ID}} \cdot \left(\log_g h_2 + \beta \log_g h_3\right) + \left(a_v - \frac{a_u}{\alpha - \mathsf{ID}}\right) \cdot \langle \boldsymbol{f}, \gamma_{\mathsf{ID}} || \beta\gamma_{\mathsf{ID}} \rangle \ , \tag{12}$$

where $\langle \cdot, \cdot \rangle$ denotes inner-product in $\mathbb{Z}_p$, and $\gamma_{\mathsf{ID}}||\beta\gamma_{\mathsf{ID}}$ is the vector of length $2(q+1)$ that consists of the concatenation of γ_{ID} and $\beta\gamma_{\mathsf{ID}}$. Note that the vector $\gamma_{\mathsf{ID}}||\beta\gamma_{\mathsf{ID}}$ is not contained in the linear span of the columns of the matrix V, and therefore even given all the above knowledge the value $\langle \boldsymbol{f}, \gamma_{\mathsf{ID}}||\beta\gamma_{\mathsf{ID}} \rangle$ is still uniformly distributed in $\mathbb{Z}_p$. In addition, the assumption that the ciphertext is invalid is equivalent to $a_v - a_u/(\alpha - \mathsf{ID}) \neq 0$, and therefore the value a_y is uniformly distributed as well. Now, assuming that the adversary obtains at most ℓ bits of leakage, then from the adversary's view the value a_y has average min-entropy at least $\log p - \ell$, and this implies that the probability that this invalid ciphertext is accepted (i.e., the probability that the adversary computes y that passes the validity test) is at most $2^\ell/p$.

An almost identical argument holds for all the subsequent invalid decryption queries. The only difference is that each time the decryption oracle rejects an invalid ciphertext the adversary can rule out one more value of $\boldsymbol{f}$. This shows that the decryption algorithm accepts the i-th invalid ciphertext with probability at most $2^\ell/(p-i+1)$. The claim now follows from the fact that the number q_C of decryption queries is polynomial, and from the restriction $\ell \leq \log p - \omega(\log n)$. □

□

Lemma F.2 *If Z is random then $\mathcal{A}$ has only a negligible advantage in outputting the bit b.*

Proof. We denote by $(u^*, v^*, s^*, w^*, y^*)$ and ID^* the challenge ciphertext and challenge identity, respectively, and denote by Collision the event in which for one of $\mathcal{A}$'s decryption queries (u, v, s, w, y) it holds that $(u, v, s, w) \neq (u^*, v^*, s^*, w^*)$ and $H(u, v, s, w) = H(u^*, v^*, s^*, w^*)$. We prove Lemma F.2 in a sequence of three claims. First, we prove that assuming that the event Collision does not occur, the decryption algorithm rejects all invalid ciphertexts except with a negligible probability. Then, we show that if the decryption algorithm rejects all invalid ciphertexts, then $\mathcal{A}$ has only a negligible advantage in outputting the bit b (we note that this is essentially the only part in the proof of this lemma that differs from [Gen06], given our analysis from the proof of Lemma F.1). Finally, we prove that the event Collision occurs with only a negligible probability.

Claim F.2 *If Z is random and the event* Collision *does not occur, then the decryption algorithm rejects all invalid ciphertexts except with a negligible probability.*

Proof. Suppose that $\mathcal{A}$ submits a decryption query $(\mathsf{ID}, (u, v, s, w, y))$ with an invalid ciphertext. Let $\beta = H(u, v, s, w)$ and $\beta^* = H(u^*, v^*, s^*, w^*)$. For any such query that is submitted prior to the challenge phase the analysis of Claim F.1 still applies, since the view of the adversary up to this point is independent of whether Z is $e(g_{q+1}, g')$ or random. For any such query that is submitted after the challenge phase it holds that $(\mathsf{ID}, (u, v, s, w, y)) \neq (\mathsf{ID}^*, (u^*, v^*, s^*, w^*, y^*))$, and therefore there are three cases to consider:

Case 1: $(u, v, s, w) = (u^*, v^*, s^*, w^*)$. If $\mathsf{ID} = \mathsf{ID}^*$ then $y \neq y^*$ and therefore the ciphertext (u, v, s, w, y) is not well-formed for ID and will be rejected. If $\mathsf{ID} \neq \mathsf{ID}^*$, then the adversary has to compute the y that satisfies Equation (12) in order for the ciphertext to be well-formed. However, we claim that the vector $\gamma_{\mathsf{ID}}||\beta\gamma_{\mathsf{ID}}$ (from Equation (12)) is linearly independent of the vector $\gamma_{\mathsf{ID}^*}||\beta\gamma_{\mathsf{ID}^*}$ (from the challenge ciphertext) and the columns of the matrix V, and therefore (as in the proof of Claim F.1) $\mathcal{A}$ cannot generate such a y except with probability $2^\ell/(p-i+1)$, where (u, v, s, w, y) is the i-th invalid ciphertext.

To see that these vectors are indeed linearly independent, denote by $V_1, \ldots, V_{2q-2}$ the columns of the matrix V, and suppose that there exist integers $a_1, \ldots, a_{2q}$, not all zero, such that $a_1V_1 + \cdots + a_{2q-2}V_{2q-2} + a_{2q-1}(\gamma_{\mathsf{ID}}||\beta\gamma_{\mathsf{ID}}) + a_{2q}(\gamma_{\mathsf{ID}^*}||\beta\gamma_{\mathsf{ID}^*})$ is the zero vector in $\mathbb{Z}_p^{2(q+1)}$. Then, either $a_1, \ldots, a_{q-1}, a_{2q-1}, a_{2q}$ or $(a_q, \ldots, a_{2q-2}, a_{2q-1}, a_{2q})$ is not all zeros. In the first case, note that the first $q+1$ coordinates of the vectors $V_1, \ldots, V_{q-1}, \mathsf{ID}, \mathsf{ID}^*$ form a invertible matrix, but the first $q+1$ coordinates of $a_1V_1 + \cdots + a_{q-1}V_{q-1} + a_{2q-1}(\gamma_{\mathsf{ID}}||\beta\gamma_{\mathsf{ID}}) + a_{2q}(\gamma_{\mathsf{ID}^*}||\beta\gamma_{\mathsf{ID}^*})$ is the zero vector in $\mathbb{Z}_p^{q+1}$ and this is not possible. The second case is similarly analyzed.

Case 2: $(u, v, s, w) \neq (u^*, v^*, s^*, w^*)$ and $\beta = \beta^*$. This case is not possible since we assume that the event Collision does not occur.

Case 3: $(u, v, s, w) \neq (u^*, v^*, s^*, w^*)$ and $\beta \neq \beta^*$. In this case the adversary has to compute the y that satisfies Equation (12) in order for the ciphertext to be well-formed. If $\mathsf{ID} \neq \mathsf{ID}^*$ then the same analysis as in case 1 shows that the adversary has only a negligible probability in computing such y. If $\mathsf{ID} = \mathsf{ID}^*$, then the vectors $V_1, \ldots, V_{2q-2}, (\gamma_{\mathsf{ID}}||\beta\gamma_{\mathsf{ID}}), (\gamma_{\mathsf{ID}^*}||\beta^*\gamma_{\mathsf{ID}^*})$ are linearly independent and the same analysis applies.

□

Claim F.3 *If Z is random and the decryption algorithm rejects all invalid ciphertexts, then $\mathcal{A}$ has only a negligible advantage in outputting the bit b.*

Proof. We prove the claim by analyzing the distribution of $e(u^*, h_{\mathsf{ID}^*,1})v^{*r_{\mathsf{ID}^*,1}}$ from the adversary's point of view. Ignoring any leakage information from the secret key sk_{ID^*} of the challenge identity for now, we argue that $r_{\mathsf{ID},1}$ is uniformly distributed and independent from the adversary's view: the adversary's view contains the values of f_1 on at most $q_{\mathsf{ID}} + q_C$ identities, and therefore secret-key queries and leakage queries on any $\mathsf{ID} \neq \mathsf{ID}^*$ do not restrict $r_{\mathsf{ID},1}$ due to the degree of the polynomial f_1, decryptions of valid ciphertexts do not reveal any additional information, and all invalid ciphertexts are assumed to be rejected. The adversary may obtain at most ℓ bits of leakage on sk_{ID^*}, and therefore from the adversary's point of view prior to the challenge phase (and, in particular, before the seed s^* is chosen) it holds that $r_{\mathsf{ID}^*,1}$ has average min-entropy at least $\log p - \ell$ (note that after the challenge phase the adversary obtains no information on $r_{\mathsf{ID}^*,1}$).

In addition, observe that

$$\begin{aligned} e\left(u^*, h_{\mathsf{ID}^*,1}\right) v^{*r_{\mathsf{ID}^*,1}} &= e\left(u^*, \left(h_1 g^{-r_{\mathsf{ID}^*,1}}\right)^{1/(\alpha-\mathsf{ID}^*)}\right) v^{*r_{\mathsf{ID}^*,1}} \\ &= e\left(u^*, h_1\right)^{\alpha-\mathsf{ID}^*} \left(\frac{v^*}{e(u^*, g)^{1/(\alpha-\mathsf{ID}^*)}}\right)^{r_{\mathsf{ID}^*,1}} , \end{aligned}$$

and since Z is completely random and independent of all other parameters then with probability $1 - 1/p$ it holds that $v^* \neq e(u^*, g)^{1/(\alpha-\mathsf{ID}^*)}$. Therefore, with probability $1 - 1/p$ also the value $e(u^*, h_{\mathsf{ID}^*,1})v^{*r_{\mathsf{ID}^*,1}}$ has average min-entropy at least $\log p - \ell$ conditioned on the adversary's view. Thus, the average-case strong extractor guarantees that the challenge message is masked statistically. □

Claim F.4 *The event* Collision *occurs with only a negligible probability.*

Proof. Given an adversary $\mathcal{A}$ for which the event Collision occurs with a noticeable probability, we construct an algorithm $\mathcal{A}'$ that breaks the security of the collection $\mathcal{H}$ of universal one-way hash functions:

1. A' chooses $u \in \mathbb{G}$, $v \in \mathbb{G}_T$, $s \in \{0,1\}^d$ and $w \in \{0,1\}^m$ uniformly at random, and announces (u, v, s, w).

2. $\mathcal{A}'$ is given a randomly chosen function $H \in \mathcal{H}$.

3. A' chooses random generators $g, h_1, h_2, h_3 \in \mathbb{G}$ and a random $\alpha \in \mathbb{Z}_p$. Then, $\mathcal{A}'$ lets $g_1 = g^\alpha$, $\mathsf{mpk} = (g, g_1, h_1, h_2, h_3, H)$, $\mathsf{msk} = \alpha$, and sends mpk to $\mathcal{A}$.

4. $\mathcal{A}'$ simulates the secret-key, leakage, and decryption queries to $\mathcal{A}$ using msk.

5. In the challenge phase $\mathcal{A}'$ ignores the two messages $\mathsf{m}_0, \mathsf{m}_1 \in \{0,1\}^m$, computes

$$\beta = H(u, v, s, w),\ y = e(u, h_{\mathsf{ID}^*,2} h_{\mathsf{ID}^*,3}{}^{\beta}) v^{r_{\mathsf{ID}^*,2} + r_{\mathsf{ID}^*,3}\beta} \ ,$$

   and sends $\mathcal{A}$ the challenge ciphertext (u, v, s, w, y).

6. If at some point $\mathcal{A}$ submits a decryption query with (u', v', s', w') such that $(u', v', s', w') \neq (u, v, s, w)$ and $H(u', v', s', w') = H(u, v, s, w)$ then $\mathcal{A}'$ outputs (u', v', s', w'). Otherwise $\mathcal{A}'$ outputs $\perp$.

Claims F.2 and F.3 guarantee that as long as the event $\mathsf{Collision}$ does not occur, then $\mathcal{A}$ cannot distinguish between $\mathcal{A}'$ and $\mathcal{D}$. Specifically, both in the interaction with $\mathcal{A}'$ and in the interaction with $\mathcal{D}$ the component in the challenge ciphertext that depends on the bit b is ε-close to uniform given the adversary's view (for some negligible ε). Therefore, with a non-negligible probability $\mathcal{A}$ submits a decryption query with (u', v', s', w') such that $(u', v', s', w') \neq (u, v, s, w)$ and $H(u', v', s', w') = H(u, v, s, w)$, and in this case $\mathcal{A}'$ finds a collision. □

□

□

F.2 A Generic Transformation in the BRM

In the setting of relative leakage Naor and Segev [NS09] proved that the Naor-Yung "double encryption" paradigm [DDN00, NY90, Lin06, Sah99] can be used to construct a CCA-secure public-key encryption scheme from any CPA-secure one using non-interactive zero-knowledge proofs. The key property of the transformation is that the size of the secret key in the resulting CCA-secure scheme is exactly the same as in the underlying CPA-secure scheme, and this in turns enables to preserve the relative amount of leakage to which the scheme is resilient. Moreover, we point out that since the resulting scheme also preserves the efficiency of the underlying scheme (when ignoring computations that are independent of the amount of leakage), this implies that the same transformation extends to the BRM as well. For completeness we provide here the description of the transformation, and refer the reader to [NS09] for the proof of security.

Let $\Pi_0 = (\mathsf{KeyGen}_0, \mathsf{Encrypt}_0, \mathsf{Decrypt}_0)$ be a public-key encryption scheme that is semantically secure in the BRM against chosen-plaintext attacks with leakage ℓ, and let $\Pi_1 = (\mathsf{KeyGen}_1, \mathsf{Encrypt}_1, \mathsf{Decrypt}_1)$ be any public-key encryption scheme that is semantically secure against chosen-plaintext attacks (note that Π_1 is not required to be resilient to leakage). Let $(\mathcal{P}, \mathcal{V})$ be a one-time simulation-sound adaptive NIZK proof system for the following NP-language[12]:

$$L = \{(c_0, c_1, \mathsf{pk}_0, \mathsf{pk}_1) \mid \exists \mathsf{m}, r_0, r_1 \text{ s.t. } c_0 = \mathsf{Encrypt}_0(\mathsf{m}, \mathsf{pk}_0; r_0) \text{ and } c_1 = \mathsf{Encrypt}_1(\mathsf{m}, \mathsf{pk}_1; r_1)\} \ .$$

The following scheme is semantically secure against chosen-ciphertext attacks in the BRM with leakage ℓ:

Key generation: Sample $(\mathsf{sk}_0, \mathsf{pk}_0) \leftarrow \mathsf{KeyGen}_0(1^\lambda)$ and $(\mathsf{sk}_1, \mathsf{pk}_1) \leftarrow \mathsf{KeyGen}_1(1^\lambda)$, and a reference string σ for the NIZK proof system. Output $\mathsf{sk} = \mathsf{sk}_0$ and $\mathsf{pk} = (\mathsf{pk}_0, \mathsf{pk}_1, \sigma)$.

Encryption: On input a message m choose $r_0, r_1 \in \{0,1\}^*$, and compute $c_0 = \mathsf{Encrypt}_0(\mathsf{m}, \mathsf{pk}_0; r_0)$ and $c_1 = \mathsf{Encrypt}_1(\mathsf{m}, \mathsf{pk}_1; r_1)$. Then, invoke the NIZK prover $\mathcal{P}$ to obtain a proof π for the statement $(c_0, c_1, \mathsf{pk}_0, \mathsf{pk}_1) \in L$ with respect to the reference string σ. Output the ciphertext (c_0, c_1, π).

Decryption: On input a ciphertext (c_0, c_1, π), invoke the NIZK verifier $\mathcal{V}$ to verify that π is an accepting proof with respect to the reference string σ. If $\mathcal{V}$ accepts then output $\mathsf{Decrypt}_0(c_0, \mathsf{sk}_0)$, and otherwise output $\perp$.

[12] We refer the reader to [Lin06, Sah99] for the definition of a one-time simulation-sound adaptive NIZK proof system.