Simultaneous Hardcore Bits and Cryptography Against Freezing Attacks

Adi Akavia* Shafi Goldwasser† Vinod Vaikuntanathan‡

Abstract

This paper considers two questions in cryptography.

1. **Simultaneous Hardcore Bits.** Let f be a one-way function. We say that a block of bits of x are *simultaneously hard-core* for $f(x)$ if given $f(x)$, they cannot be distinguished from a random string of the same length. Although there are many examples of (candidate) one-way functions with one hardcore bit (or even $O(\log n)$ simultaneously hardcore bits), there are very few examples of one-way functions (and even fewer examples of trapdoor one-way functions) for which a linear number of the input bits are simultaneously hardcore.

   We show that for the lattice-based (injective) trapdoor function recently proposed by Gentry, Peikert and Vaikuntanathan (STOC 2008), which is in turn based on the one-way function of Regev (STOC 2005), an $n - o(n)$ number of input bits are simultaneously hardcore (where n is the total number of input bits).

2. **Cryptography Against Memory-Freezing Attacks.** The absolute privacy of the secret keys associated with cryptographic algorithms has been the corner-stone of modern cryptography. Still, it has been clear that in practice keys do get compromised at times, by various means. In a particularly devastating side-channel attack, termed the "freezing attack" which was proposed recently, a significant fraction of the bits of the secret key can be measured if the secret key is *ever stored* in the part of memory which can be accessed (even after power has been turned off for a short amount of time). Such an attack has been shown to completely compromise the security of various cryptosystems, including the RSA cryptosystem and variants.

   We show that the public-key encryption scheme of Regev (STOC 2005), and the identity-based encryption scheme of Gentry, Peikert and Vaikuntanathan (STOC 2008) are remarkably robust against freezing attacks where the adversary can measure a large fraction of the bits of the secret-key. This is done without increasing the size of the secret key, or by introducing any complication of the natural encryption and decryption routines.

Although seemingly completely different, these two problems turn out to be very similar: in particular, our results demonstrate that *the proof techniques* that can be used to solve both these problems are intimately related.

*Institute of Advanced Study, Princeton, NJ and DIMACS, Rutgers. EMAIL: akavia@ias.edu

†MIT and Weizmann Institute. Supported in part by NSF grants CCF-0514167, CCF-0635297, NSF-0729011 and the Israel Science Foundation 700/08. EMAIL: shafi@csail.mit.edu

‡MIT and IBM Research. Supported in part by NSF grants CCF-0635297 and Israel Science Foundation 700/08. EMAIL: vinodv@alum.mit.edu

1 Introduction

This paper considers two questions in cryptography. The first is the ability to prove that many input bits are simultaneously hardcore for efficient trapdoor one-way functions f. The second is to construct a public-key encryption scheme and an identity-based encryption scheme that withstand a strong kind of side-channel attack that was recently proposed in the literature, called "memory-freezing attacks"[18]. Although seemingly completely different, we show that these two problems are in fact related. In particular, our results demonstrate that *the techniques* that can be used to solve both problems are very closely related.

We go on to elaborate on each of these problems, and our contributions in some detail.

1.1 Simultaneous Hard-Core Bits

The notion of hard-core bits for one-way functions was introduced very early in the developement of the theory of cryptography [17, 4, 38]. Indeed, the existence of hard-core bits for particular proposals of one-way functions (see, for example [4, 1, 19, 22]) and later for any one-way function [14], has been central to the constructions of secure public (and private) key encryption schemes, and strong pseudo-random bit generators, the cornerstones of cryptography.

The main questions which remain open in this area concern the generalized notion of "simultaneous hard-core bit security" loosely defined as follows. Let f be a one-way function and h an easy to compute function. We say that h is a *simultaneously hard-core function* for f if given $f(x)$, $h(x)$ is computationally indistinguishable from random. In particular, we say that a block of bits of x are simultaneously hard-core for $f(x)$ if given $f(x)$, they cannot be distinguished from a random string of the same length (this corresponds to a function h that outputs a subset of its input bits).

The question of how many bits of x can be proved simultaneously hard-core has been studied for general one-way functions as well as for particular candidates in [37, 1, 26, 20, 15, 14], but the results obtained are far from satisfactory. For a general one-way function (modified in a similar manner as in their hard-core result) [14] has shown the existence of h that outputs $\log k$ bits (where k is the security parameter) which is a simultaneous hard-core function for f. For particular candidate one-way functions such as the the exponentiation function (modulo a prime p), the RSA function and the Rabin function [37, 26] have pointed to particular blocks of $O(\log k)$ input bits which are simulateously hard-core given $f(x)$ (where k is the security parameter).

The only known examples of one-way functions that have more than $O(\log k)$ simultaneous hardcore bits are the modular exponentiation function $f(x) = g^x \bmod N$ [20, 15], where N is an RSA composite, and the Pallier function[31]. [20, 15] show that for the modular exponentiation function (modulo an RSA composite N), *half* the bits of x (resp, any constant fraction of the bits of x) are simulatenously hard-core, given $g^x \bmod N$, under the factoring assumption (resp. a stronger variant of the discrete logarithm assumption [32]). In the case of the Paillier function, [6] show that any constant fraction of the bits of the input are hardcore, under a strong variant of Paillier's assumption (or, the composite residuosity assumption). In particular, the Paillier function is the only known *trapdoor function* where a linear fraction of the input bits are simultaneously hardcore. [6] raised the question of whether it is possible to construct other natural and efficient trapdoor functions with many simultaneous hardcore bits.

In this paper, we show for the lattice-based (injective) trapdoor function recently proposed by Gentry, Peikert and Vaikuntanathan [13] (based on the one-way function of Regev [35]), an $n - o(n)$ number of input bits are simultaneously hardcore. The one-wayness of the function is based on the hardness of the learning with error (LWE) problem with dimension (security parameter) n which is defined as follows:

given polynomially many pairs of the form $(\mathbf{a}_i, \langle \mathbf{a}_i, \mathbf{s} \rangle + x_i)$ where $\mathbf{s} \in \mathbb{Z}_q^n$ and $\mathbf{a}_i \in \mathbb{Z}_q^n$ (for some prime $q = \mathsf{poly}(n)$) are uniformly random and independent and the x_i are chosen from some "error distribution" (in particular, think of x_i's as being small in magnitude), find $\mathbf{s}$.

In particular, we show:

Informal Theorem 1. *There exists an injective trapdoor function for which $n - k$ bits are simultaneously hardcore (for any k), assuming that the hardness of the learning with error (LWE) assumption with dimension k against polynomial-time adversaries. Here, n is the input-length of the trapdoor function.*

Regev [35] showed that the complexity of LWE is intimately connected to the *worst-case* complexity of many lattice problems. In particular, he showed that any algorithm that solves the LWE problem (for appropriate parameters m and q and an error distribution χ) can be used to solve many lattice-problems *in the worst-case* using a quantum algorithm. Thus, the one-wayness of this function is based on the worst-case quantum hardness of lattice problems as well.

Our proof is simple, and general: one of the consequences of the proof is that the related one-way function based on learning parity with noise (in $GF(2)$) [2] also has $n - o(n)$ simultaneous hardcore bits (See Sections 2.1 and 4).

1.2 Security against Memory-Freezing Side-Channel Attacks

The absolute privacy of the secret-keys associated with cryptographic algorithms has been the corner-stone of modern cryptography. Still, in practice keys do get compromised at times for a variety or reasons. A particularly disturbing loss of secrecy is as a result of side channel attacks. One may distinguish, as we do here, between two types of side-channel attacks on secret-keys: computational and memory-freezing.

Informally, a computational side-channel attack is the leakage of information about the secret key which occurs as a result of preforming a computation on the secret-key (by some cryptographic algorithm which is a function of the secret-key). Some well-known examples of computational side-channel attacks are timing attacks [23], power attacks [24] and cache attacks [30] (see [27] for a glossary of various side-channel attacks). A basic defining feature of a computational side-channel attack, as put forth by Micali and Reyzin [29] in their work on "Physically Observable Cryptography" is that in this case *computation and only computation leaks information*. Portions of memory which are not involved in computation do not leak during that computation. There has been a growing amount of interest in designing cryptographic algorithms robust against computational side-channel attacks, as evidenced by the many recent works in this direction [29, 21, 34, 16, 11]. A major approach in designing cryptographic algorithms against computational side-channel attacks is to somehow limit the portions of the secret key which are involved in each step of the computation [21, 34, 16, 11].

A different type of attack entirely which has recently received much attention, is the memory-freezing attack intrduced by Felton et al. [18]. In this attack, a significant fraction of the bits of the secret key can be measured if the secret key is *ever stored* in the part of memory which can be accessed (even after power has been turned off for a short amount of time), and even if it has not been touched by computation. This attack violates the basic assumption of [29] that *only computation* leaks information. Obviously, if the attack uncovers the entire secret key, there is no hope for any cryptography. However, it seems that such an attack usually only recovers some fraction of the secret key.

The question that emerges is whether cryptosystems can sustain their security in presence of such an attack. There are two natural directions to take in addressing this question.

The first is to look for redundant representations of secret-keys which will enable battling memory freezing attacks. The works of [5, 21] can be construed in this light. Naturally, this entails expansion of the storage required for secret keys and data.

The second approach would be to examine natural and existing cryptosystems, and see how vulnerable they are to memory-freezing attacks which uncovers a fixed fraction of bits of the secret key. Indeed, [18] shows that uncovering half of the bits of the secret key stored in the natural way completely compromises the security of cryptosystems, such as the RSA and Rabin cryptosystems. This follows from the work of Rivest and Shamir, and Coppersmith [36, 7], and has been demonstrated in practice by [18]: their experiments described successfuly recovered RSA and AES keys.

In this paper, we take the second approach: we prove that the public-key encryption scheme of Regev[35] and the identity-based encryption scheme of Gentry, Peikert and Vaikuntanathan [13] are remarkably robust against the memory-freezing attack.

In particular, we differentiate between two flavors of this attack [1]. The first is *non-adaptive α-freezing attacks*. Intuitively, in this case, a function h with output-length α is chosen by the adversary first, and the adversary is given $(\text{PK}, h(\text{SK}))$, where (PK, SK) is a random key-pair produced by the key-generation algorithm. The key point to note is that the function h is fixed in advance, independent of the parameters of the system and in particular PK. We remark that even though seems like a weak attack, it is the attack specified in [18] as it corresponds to the fact that the bits measured are a function of the hardware or rather the storage medium used, and do not depend on the choice of the public key (See the definition in Section 2.3 and the discussion that follows).

In this case we show:

Informal Theorem 2. *(Under variants of the LWE assumption) there exists a public-key encryption scheme and an identity-based encryption scheme that are secure against a non-adaptive $(n - o(n))$-freezing attack, where n is the size of the secret-key.*

The second, stronger flavor is *adaptive memory freezing attacks*. In this case, the key generation algorithm is run first to output a pair (PK, SK), and then the adversary on input PK chooses functions h_i adaptively (depending on the PK and the outputs of $h_j(SK)$, for $j < i$) and receives $h_i(SK)$. In this case, we show:

Informal Theorem 3. *(Under variants of the LWE assumption) there exists a public-key encryption scheme and an identity-based encryption scheme that are secure against an adaptive $\frac{n}{\mathsf{polylog}(n)}$-freezing attack, where n is the size of the secret-key.*

We find it extremely interesting to construct encryption schemes which are secure against α-freezing attacks, where α is an arbitrary polynomial in the size of the secret-key. Of course, if the secret-key is kept static, this is not achievable (since the adversary can measure the entire secret-key, as soon as α is larger than the length of the secret-key). Thus, it seems that to achieve this goal, some off-line (randomized) refreshing of the secret key must be done periodically. We do not deal with these further issues in this paper. (However, for more on this issue, see the discussion in Section 2.3).

2 Preliminaries and Definitions

We will let bold capitals such as $\mathbf{A}$ denote matrices, and bold small letters such as $\mathbf{a}$ denote vectors. If $\mathbf{A}$ is an $m \times n$ matrix and $S \subseteq [n]$ represents a subset of the columns of $\mathbf{A}$, we let $\mathbf{A}_S$ denote the restriction of

[1] In this paper, we are concerned with designing public-key encryption and identity-based encryption schemes. Thus, our description will be tailored to the case of encryption schemes.

$\mathbf{A}$ to the columns in S, namely the $m \times |S|$ matrix consisting of the columns with indices in S. In this case, we will write $\mathbf{A}$ as $[\mathbf{A}_S, \mathbf{A}_{\overline{S}}]$.

2.1 Cryptographic Assumptions

The cryptographic assumptions we make are related to the hardness of learning-type problems. In particular, we will consider the hardness of learning parity over $GF(2)$ with noise (equivalently, the hardness of decoding random linear codes over $GF(2)$) and the hardness of learning with error. The latter problem was introduced by Regev [35] where he showed a relation between the hardness of this problem, and the worst-case hardness of certain problems on lattices.

Learning With Error (LWE). Learning with Error, defined by Regev[35], is a variant of learning parity with noise. The interesting feature of this problem is the relation between its average-case hardness and the (quantum) worst-case hardness of standard lattice-problems.

Our notation here follows [35, 33]. Before we define the problem, we will define a normal distribution over $\mathbb{R}$ and its discretization. The *normal distribution* with mean 0 and variance σ^2 (or standard deviation σ) is the distribution on $\mathbb{R}$ having density function $\frac{1}{\sigma \cdot \sqrt{2\pi}} \exp(-x^2/2\sigma^2)$. It is possible to efficiently sample from a normal variable to any desired level of accuracy.

For $\alpha \in \mathbb{R}^+$ we define Ψ_α to be the distribution on $[0, 1)$ of a normal variable with mean 0 and standard deviation $\alpha/\sqrt{2\pi}$, reduced modulo 1 [2]. For any probability distribution $\phi : \mathbb{T} \to \mathbb{R}^+$ and an integer $q \in \mathbb{Z}^+$ (often implicit) we define its *discretization* $\bar{\phi} : \mathbb{Z}_q \to \mathbb{R}^+$ to be the discrete distribution over $\mathbb{Z}_q$ of the random variable $\lfloor q \cdot X_\phi \rceil \bmod q$, where X_ϕ has distribution ϕ [3].

Consider the family of functions $\mathcal{F}_{\mathsf{LWE}}$, parametrized by numbers $m(n) \in \mathbb{N}$ and $q(n) \in \mathbb{N}$ and a probability distribution $\chi(n) : \mathbb{Z}_q \to \mathbb{R}$, defined the following way: Let n be a security parameter. Each function $f_{\mathbf{A}}$ is indexed by a matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$. The input of $f_{\mathbf{A}}$ is $(\mathbf{s}, \mathbf{x})$ where $\mathbf{s}$ is chosen uniformly at random from $\mathbb{Z}_q^n$ and $\mathbf{x} = (x_1, \ldots, x_m)$ is chosen such that the x_i's are independent and each $x_i \leftarrow \chi$. The output is $f_{\mathbf{A}}(\mathbf{s}, \mathbf{x}) = \mathbf{A}\mathbf{s} + \mathbf{x}$, where all operations are performed in $\mathbb{Z}_q$.

The hardness of LWE is parametrized chiefly by the dimension n. Therefore, we let all other parameters (m, q and χ) be functions of n, sometimes omitting the explicit dependence for notational clarity.

We say that the $(m(n), q(n), \chi(n))$-LWE problem is $t(n)$-hard if for every family of circuits Adv of size at most $t(n)$,

$$\Pr[\mathsf{Adv}(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{x}) = \mathbf{s}] \leq \frac{1}{t(n)}$$

where the probability is over the choice of a random $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, random $\mathbf{s} \in \mathbb{Z}_q^n$ and a vector $\mathbf{x} = (x_1, \ldots, x_m)$ is chosen such that each x_i is chosen independently from the distribution χ.

In other words, the assumption says that $f_{\mathbf{A}}$ (for a randomly chosen $\mathbf{A}$) is a one-way function against adversaries of size $t(n)$. Regev[35] showed that if $f_{\mathbf{A}}$ is a one-way function, then it is a pseudorandom generator as well (where the distinguishing probability is worse by a factor of $m(n)$, the length of the output).

Regev [35] demonstrated a connection between the LWE problem for certain moduli q and error distributions χ, and worst-case lattice problems. In particuar, he showed that $\mathsf{LWE}_{q,\chi}$ is as hard as solving several standard *worst-case* lattice problems *using a quantum algorithm*. We state a version of his result here.

[2] For $x \in \mathbb{R}$, $x \bmod 1$ is simply the fractional part of x.

[3] For a real x, $\lfloor x \rceil$ is the result of rounding x to the nearest integer.

Proposition 1 ([35]). *Let $\alpha = \alpha(n) \in (0,1)$ and let $q = q(n)$ be a* prime *such that $\alpha \cdot q > 2\sqrt{n}$. If there exists an efficient (possibly quantum) algorithm that solves $\mathsf{LWE}_{q,\overline{\Psi}_\alpha}$, then there exists an efficient* quantum *algorithm for solving the worst-case lattice problems SIVP and GapSVP in the ℓ_2 norm.*

We stress that our cryptosystems will be defined purely in relation to the LWE problem, without explicitly taking into account the connection to lattices (or their parameter restrictions). The connection to lattices for appropriate choices of the parameters will then follow by invoking Proposition 1, which will ensure security assuming the (quantum) hardness of lattice problems.

Learning Parity With Noise (LPN). See Appendix A.

2.2 Cryptographic Definitions

The notion of a meaningful/meaningless public-key encryption scheme was first proposed by Kol and Naor [25] [4]. Such encryption schemes have two types of public-keys: meaningful public-keys, which retain full information about the encrypted message (which can be recovered using a matching secret-key) and meaningless public-keys, which lose all information about the message. Moreover, meaningful and meaningless public-keys are computationally indistinguishable. A formal definition follows.

Definition 1. ([25]) *A triple of algorithms* $\text{PKE} = (\text{GEN}, \text{ENC}, \text{DEC})$ *is called a meaningful/meaningless encryption scheme if it has the following three properties.*

- **Meaningful Keys:** *With high probability over* $(\text{PK}, \text{SK}) \leftarrow \text{GEN}(1^n)$*, for every message* m *and ciphertext* $c \leftarrow \text{ENC}(\text{PK}, m)$, $\text{DEC}_{\text{SK}}(c) = m$.
- **Meaningless Keys:** *There is an efficient algorithm* BADGEN *such that with high probability over* $\text{PK} \leftarrow \text{BADGEN}(1^n)$*, for every two messages* m_0 *and* m_1, $\text{ENC}_{\text{PK}}(m_0) \approx_s \text{ENC}_{\text{PK}}(m_1)$.
- **Indistinguishability of Meaningful and Meaningless Keys:** *The following two distributions are computationally indistinguishable.*

$$\{\text{PK} : (\text{PK}, \text{SK}) \leftarrow \text{GEN}(1^n)\} \approx_c \{\text{PK} : \text{PK} \leftarrow \text{BADGEN}(1^n)\}$$

Semantic security for meaningful meaningless encryption schemes follow from these three properties: to see this, observe that given a meaningless public-key, no (even unbounded) algorithm can distinguish between encryption of m_0 and encryption of m_1 under the public-key. Thus, if an adversary manages to distinguish between the encryptions of m_0 and m_1 using the meaningful public-key, it must mean that the adversary is also an efficient distinguisher between meaningful and meaningless public keys. Since these two kinds of keys are indistinguishable, semantic security follows. (For a stronger statement and proof, see Lemma 4).

2.3 Defining Memory-Freezing Attacks

In this section, we define the security of cryptographic primitives against freezing attacks. In particular, we define the semantic security of public-key encryption schemes, and identity-based encryption schemes

[4] A definition of a similar flavor has been around in the context of commitment schemes even earlier. See the mixed commitment primitive of [8].

against freezing attacks. The definitions in this section can be extended to other cryptographic primitives as well, but we omit these extensions.

Generally speaking, we follow similar definitions that appeared in the literature before (most notably, the definitional framework of Micali and Reyzin [29]) except for a few important differences: on the one hand, whereas the definition of Micali and Reyzin captures *computational side-channel attacks*, our goal is to capture the stronger notion of *memory-freezing attacks*. In particular, [29] make the key axiomatic assumption (used in many of the later works, for instance [16, 12]) that "only computation leaks information". This assumption, although reasonable in the context of computational side-channel attacks, is simply false when considering memory-freezing attacks: for that reason, our definition does not make such an assumption.

On the other hand, one of the consequences of our definitions is that they are achievable only against side-channel attacks in which the total number of bits measured during the attack is upper-bounded by the length of the original secret-key (for more discussion on this issue, see Section 2.3.1).

We proceed to define two flavors of semantic security against freezing attacks, security against non-adaptive freezing attacks, and against adaptive freezing attacks.

Semantic Security Against Non-Adaptive Freezing Attacks. Non-adaptive freezing attacks capture the scenario in which the measurement function h is fixed in advance (possibly as a function of the encryption scheme, and the underlying hardware), but *independent of the parameters of the system*, for example the public-key of the encryption scheme. The definition is parametrized by a function $\alpha(n)$, and requires that for any h whose output-length is bounded by $\alpha(s(n))$ (where $s(n)$ is the length of the secret-key output by the key-generation algorithm) the scheme remains semantically secure (just as in Goldwasser and Micali [17]), *even if the adversary is also given* $h(\mathrm{SK})$. The formal definition follows.

Definition 2. *Let* $\alpha : \mathbb{N} \to \mathbb{N}$ *be a function, and let the size of the secret-key output by* $\mathrm{GEN}(1^n)$ *be* $s(n)$*. A public-key encryption scheme* $\mathrm{PKE} = (\mathrm{GEN}, \mathrm{ENC}, \mathrm{DEC})$ *is* semantically secure against non-adaptive $\alpha(n)$-freezing attacks *if for any function* $h : \{0,1\}^{s(n)} \to \{0,1\}^{\alpha(s(n))}$*, and for any PPT adversary* $A = (A_1, A_2)$*, the probability that* A *wins in the following experiment differs from* $\frac{1}{2}$ *by a negligible function in* n*:*

$(\mathrm{PK}, \mathrm{SK}) \leftarrow \mathrm{GEN}(1^n)$
$(m_0, m_1, \mathsf{state}) \leftarrow A_1(\mathrm{PK}, h(\mathrm{SK}))$ *s.t.* $|m_0| = |m_1|$
$y \leftarrow \mathrm{ENC}_{\mathrm{PK}}(m_b)$ *where* $b \in \{0,1\}$ *is a random bit*
$b' \leftarrow A_2(y, \mathsf{state})$

The adversary A *wins the experiment if* $b' = b$*.*

Semantic Security Against Adaptive Freezing Attacks. An adaptive freezing attack is a strong form of side-channel attack, where the adversary can request for functions h of the secret-key SK (adaptively, and depending on the public-key and the results of the previous measurements). The definition is parametrized by a function $\alpha(n)$, and requires that as long as the total number of bits that the adversary gets as a result of all his measurements is at most $\alpha(s(n))$ (where $s(n)$ is the length of the secret-key), he cannot break the semantic security of the encryption scheme. We stress that the adversary is allowed to choose his measurement function h at a certain point depending on all the information he has so far, including the public-key and the results of his previous measurements.

In the formal definition, the adversary gets access to an oracle $\mathcal{S}_{\mathrm{SK}}$ (parametrized by the secret-key SK) which takes as input a polynomial-size circuit h and outputs $h(\mathrm{SK})$.

Definition 3. *Let $\alpha : \mathbb{N} \to \mathbb{N}$ be a function, and let the size of the secret-key output by* $\text{GEN}(1^n)$ *be* $s(n)$. *Let* $\mathcal{S}_{\text{SK}}$ *(parametrized by the secret-key* SK*) denote the oracle that gets as input a polynomial-size circuit* h *and outputs* $h(\text{SK})$. *A public-key encryption scheme* $\text{PKE} = (\text{GEN}, \text{ENC}, \text{DEC})$ *is* semantically secure against adaptive $\alpha(n)$-freezing attacks *if for any PPT adversary* $A = (A_1, A_2)$*, the probability that* A *wins in the following experiment differs from* $\frac{1}{2}$ *by a negligible function in* n.

$(\text{PK}, \text{SK}) \leftarrow \text{GEN}(1^n)$
$(m_0, m_1, \mathsf{state}) \leftarrow A_1^{\mathcal{S}_{\text{SK}}}(\text{PK})$ *s.t.* $|m_0| = |m_1|$
$y \leftarrow \text{ENC}_{\text{PK}}(m_b)$ *where* $b \in \{0,1\}$ *is a random bit*
$b' \leftarrow A_2(y, \mathsf{state})$

The adversary A *wins the experiment if both (a)* $b' = b$*, and (b) the total numbers of bits that* A *receives as part of the answers for the queries to the* $\mathcal{S}_{\text{SK}}$ *oracle is at most* $\alpha(s(n))$.

The definitions of (chosen-identity) semantic security against freezing attacks for identity-based encryption schemes is similar in spirit, and is given in Appendix D.

2.3.1 Definitional Issues and Remarks about the Definition

The goal of this subsection is to address several questions that arise about the definitions. We address these questions one by one.

Why is the non-adaptive definition interesting? In the non-adaptive definition, the measurement function h is (adversarially) chosen, independent of the parameters of the system, for example the public-key. This captures the case when the information that leaks from the hardware is a characteristic of the hardware only.

One might, in this case, wonder if we can design the decryption algorithm that is tailor-made for the particular h function. However, this means designing a new software (for example, the decryption algorithm) for every possible piece of hardware (for example, a smart-card implementing the decryption algorithm). This is highly impractical. Moreover, it seems that such a solution will involve artificially expanding the secret-key, which we wish to avoid. We stress that our goal is to show that a *natural and efficient* encryption scheme is secure against freezing attacks.

Why does the adversary get to measure only the secret-key, but not the "secret-memory"? The secret-memory refers to the entire configuration of the decryption machine that is intended to be private. This includes the secret-key, as well as the results of intermediate computations. Potentially, measuring these intermediate values might give more information than measuring just the secret-key.

We have two answers to this issue: first of all, in the case of our adaptive definition, we do not lose any generality by restricting the adversary to measure just the secret-key. This is because the computation of the decryption machine is deterministic and is a function of only the secret and the public keys (and the inputs that it receives). This whole computation can be captured using an h-function query to the $\mathcal{S}_{\text{SK}}$ oracle.

In the non-adaptive case, it turns out that even though the definition may not generalize, the constructions are secure even under a stronger definition which allows measurement of the secret-memory. Roughly speaking, the reason is that in the schemes we present, the decryption algorithm can be implemented using a small amount of space. This means that most of the memory is occupied by the secret-key, at any point of time. We omit further consideration of these issues, for the sake of clarity.

Which functions $\alpha(n)$ (in the definition of $\alpha(n)$-freezing attacks) are achievable? Clearly $\alpha(n)$ can be

at most n since otherwise, the adversary can read out the entire secret-key. In our case, we achieve $\alpha(n)$ up to $n - \omega(\log^2 n)$ in the case of the non-adaptive definition and upto $\frac{n}{\log^2 n}$ in the case of the adaptive definition.

We find it extremely interesting to generalize our definition to a repeated (arbitrary) polynomial number of meaurements. Obviously, in this case, some off-line refreshing of the secret key must be done periodically (by the discussion in the above paragraph). Our definition currently does not capture refreshing the secret-key, and we do not deal with these further issues in this paper.

Why does the adversary A_2 (in the adaptive definition) not get access to the oracle $\mathcal{S}_{\mathrm{SK}}$? It is easy to see that if A_2 (which gets as input a challenge ciphertext) gets to ask *even one query* to $\mathcal{S}_{\mathrm{SK}}$, it can break the semantic security of the encryption scheme. Intuitively, this is because A_2 can use the oracle $\mathcal{S}_{\mathrm{SK}}$ to decrypt the challenge ciphertext. This issue is similar to the one that arises in the definition of CCA2-security of encryption schemes, where one has to prohibit the adversary from querying the decryption oracle on the challenge ciphertext. Unfortunately, whereas the solution to this issue in the CCA2-secure encryption case is straightforward (namely, explicity disallow querying the decryption oracle on the challenge ciphertext), it seems far less clear in our case (for example, the adversary could construct a circuit that asks for the decryption of the challenge ciphertext in a number of ways, and it is unclear how we can explicitly rule out all these ways). Extending our definition to handle this is an interesting open question.

3 Public-key Encryption Secure Against Freezing Attacks

In this section, we construct public-key encryption schemes that are secure against memory-freezing attacks.

In Section 3.1, we show that a minor modification of the lattice-based public-key encryption scheme of Regev [35] is semantically secure against a *non-adaptive $\alpha(n)$-freezing attack*, where $\alpha(n) \leq n - \omega(\log^2 n)$, where n is the size of the secret-key [5]. We show the semantic security of this encryption scheme against $\alpha(n)$-freezing attacks, under the assumption that the LWE problem is hard (for polynomial-time algorithms) with security parameter (or dimension) $n - \alpha(n)$. The best known polynomial-time (in n) algorithm solves LWE upto security parameter $O(\log n \log\log n)$[3], and the best conjectured bound is $O(\log^2 n)$. Thus, with $\alpha(n) = n - \omega(\log^2 n)$, the scheme is secure against $\alpha(n)$-freezing attacks under the (plausible) assumption that LWE is hard with security parameter $\omega(\log^2 n)$.

The modified encryption scheme is different from Regev's encryption scheme only in the way the public-key is generated, and in particular, retains all the efficiency parameters of the [35] system.

In Section 3.2, we show that the Regev encryption scheme (without any modifications) is secure against *adaptive $\alpha(n)$-freezing attacks*, for any $\alpha(n) = o(\frac{n}{\log n})$. The semantic security against the $k(n)$-freezing attack is under the assumption that LWE is hard for algorithms that run in time $2^{k(n)}$. We show the semantic security of this encryption scheme against $\alpha(n)$-freezing attacks, under the assumption that the LWE problem with security parameter n is $2^{\alpha(n)+\omega(\log n)}$-hard, namely hard for algorithms that run in time $2^{\alpha(n)+\omega(\log n)}$. The best known algorithm to solve n-dimensional LWE runs in time $2^{n/\log n}$ [3]. Thus, with $\alpha(n) = o(\frac{n}{\log n})$, the scheme is secure against $\alpha(n)$-freezing attacks under the (plausible) assumption that LWE is $2^{o(n/\log n)}$-hard for security parameter n.

We stress, however, that both the results are fully parametrized and works for the whole range of $\alpha(n)$, under the LWE assumption with related security parameters, and hardness thresholds.

[5] Here, the size of the secret-key is measured as the number of elements in $\mathbb{Z}_q$ that the secret-key is made up of (for some number q). Thus, the bit-length of the secret-key is $n \log q$ and the security holds against freezing attacks that measure $\alpha(n) \log q$ bits. For simplicity, we will omit the $\log q$ factor, when it is clear from the context.

The Regev Encryption Scheme. First, we describe the public-key encryption scheme of Regev, which we denote RPKE, as well as our modification which we call RPKE$'$.

The encryption scheme RPKE $=$ (RGEN, RENC, RDEC) works as follows. Let n be the security parameter and let $q(n), m(n) \in \mathbb{N}$ and the probability distribution $\chi(n)$ over $\mathbb{Z}_q$ be parameters of the system.

- RGEN(1^n) picks a random matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, a random vector $\mathbf{s} \in \mathbb{Z}_q^n$ and a vector $\mathbf{x} = (x_1, \ldots, x_m)$ where each x_i is chosen independently from the probability distribution χ. Output PK $= (\mathbf{A}, \mathbf{As}+\mathbf{x})$ and SK $= \mathbf{s}$.
- RENC(PK, b), where b is a bit, works as follows. First, pick a vector $\mathbf{r}$ at random from $\{0,1\}^m$. Output $(\mathbf{rA}, \mathbf{r}(\mathbf{As}+\mathbf{x}) + b\lfloor \frac{q}{2} \rceil)$.
- RDEC(SK, c) first parses $c = (\mathbf{c}_0, c_1)$, computes $b' = c_1 - \mathbf{c}_0^T\mathbf{s}$ and outputs 0 if b' is closer to 0 than to $\frac{q}{2}$, and 1 otherwise.

Our modification to the Regev encryption, which we denote RPKE$' =$ (RGEN$'$, RENC$'$, RDEC$'$) is different from RPKE only in the key-generation algorithm. Let $k(n) \in \mathbb{N}$ be an additional parameter of the system. RGEN$'$ works as follows.

- RGEN$'(1^n)$ picks two random matrices $\mathbf{B} \in \mathbb{Z}_q^{m \times k}$ and $\mathbf{C} \in \mathbb{Z}_q^{k \times n}$, a random vector $\mathbf{s} \in \mathbb{Z}_q^n$ and a vector $\mathbf{x} = (x_1, \ldots, x_m)$ where each x_i is chosen independently from the probability distribution χ. Let $\mathbf{A} = \mathbf{BC}$. Output PK $= (\mathbf{A}, \mathbf{As}+\mathbf{x})$ and SK $= \mathbf{s}$.

Regev's proof of semantic-security of RPKE in fact shows that it is a meaningful/meaningless encryption scheme, to use terminology that was developed later (and therefore, a fortiori, that it is also semantically secure). The distribution of meaningless keys in RPKE is the uniform distribution of $(\mathbf{A}, \mathbf{y})$ over $\mathbb{Z}_q^{m \times (n+1)}$. It is easy to show that RPKE$'$ is a meaningful/meaningless encryption scheme as well, under the LWE assumption with security parameter $k(n)$. In RPKE$'$, the distribution of meaningless public-keys is $(\mathbf{A}, \mathbf{y})$ where $\mathbf{A} = \mathbf{BC}$ is a product of a random $m(n) \times k(n)$ matrix $\mathbf{B}$, and a a random $k(n) \times n$ matrix $\mathbf{C}$ (both with entries in $\mathbb{Z}_q$), and $\mathbf{y}$ is a random vector in $\mathbb{Z}_q^n$ (we omit the proofs of these claims). Let RBADGEN$'$ denote an algorithm the samples a meaningless public-key.

3.1 Security Against Non-Adaptive Freezing Attacks

Theorem 1. *The public-key encryption scheme* RPKE$'$ *is secure against a non-adaptive $\alpha(n)$-freezing attack, assuming that the LWE problem with security paramter (dimension) $(n - \alpha(n) - \omega(\log n))$ is hard for polynomial-time algorithms. (where the implicit parameters $m(n)$ and $q(n)$ in the LWE definition are both* polynomial in n*)*

Proof. First, we show that the meaningful and meaningless public-keys of RPKE$'$ are computationally indistinguishable, even given some information about the (real) secret-key. In particular, we show that for any function $h : \mathbb{Z}_q^n \to \mathbb{Z}_q^{\alpha(n)}$,

$$\{(\text{PK}, h(\text{SK})) : (\text{PK}, \text{SK}) \leftarrow \text{RGEN}'(1^n)\} \approx_c \{(\text{PK}', h(\text{SK})) : \text{PK}' \leftarrow \text{RBADGEN}'(1^n), (\text{PK}, \text{SK}) \leftarrow \text{RGEN}'(1^n)\}$$

Secondly, we show that if the meaningful and meaningless keys are computationally indistinguishable given $h(\text{SK})$ for any $h : \mathbb{Z}_q^n \to \mathbb{Z}_q^{\alpha(n)}$, then the encryption scheme is semantically secure against non-adaptive $\alpha(n)$-freezing attacks as well. The proof of this statement essentially follows from similar lemmas that appear in [25, 33]. The proof of the theorem follows from these two claims. For the full proof, see Appendix C. □

3.2 Security Against Adaptive Freezing Attacks

In this section, we show that the Regev encryption scheme [35] RPKE (without any modifications) is secure against $\alpha(n)$-freezing attacks, assuming that the LWE problem with dimension n is $2^{\alpha(n)+\omega(\log n)}$-hard, (for polynomial $m(n)$ and $q(n)$).

Theorem 2. *The public-key encryption scheme* RPKE *is secure against an adaptive $\alpha(n)$-freezing attack, assuming that the LWE problem with security parameter (dimension) n is $2^{\alpha(n)+\omega(\log n)}$-hard (for polynomial $m(n)$ and $q(n)$).*

Proof. (Sketch.) Our proof proceeds in three steps. First, we show that without loss of generality, it suffices to consider a simpler adversary in the definition of semantic security against adaptive freezing attacks. Recall that the adversary A, on input a public-key PK, adaptively queries an oracle on many polynomial-size circuits h_i and gets $h_i(\text{SK})$ (the choice of h is adaptive, and can depend on the PK as well as the answers $h_j(\text{SK})$ for $j < i$). The adversary then tries to break the (regular) semantic security of the encryption scheme.

We show that for every adversary A that gets a total of $\alpha(n)$ bits from the oracle, there is a circuit $h_A : \{0,1\}^{p(n)+s(n)}$ (where $p(n)$ and $s(n)$ are the sizes of the public-key and secret-key, respectively) such that A and the following adversary A' are equivalent. A' gets as input $(\text{PK}, h_A(\text{PK}, \text{SK}))$, and tries to break the regular semantic security of the encryption scheme. In particular, A' does not get oracle access to $\mathcal{S}_{\text{SK}}$. Roughly speaking, h_A "simulates" the computation of A on input PK and oracle access to $\mathcal{S}_{\text{SK}}$: h_A can do this because it has SK as one of the inputs. For the rest of the proof, we will consider an adversary A' of this form.

In the next two steps, we will establish that RPKE is a meaningful/ meaningless encryption scheme *even in the presence of the auxiliary information h_A*. In particular, we show that for the encryption scheme RPKE,

1. for every circuit $h : \{0,1\}^{s(n)} \to \{0,1\}^{\alpha(n)}$, there is a distribution $\mathcal{D}_h$ such that (a) $\mathcal{D}_h$ is computationally indistinguishable from a public-key PK generated by $\text{RGEN}(1^n)$, *even given* $h(\text{PK}, \text{SK})$ assuming that the LWE problem is $2^{\alpha(n)+\omega(\log n)}$-hard, and (b) $\mathcal{D}_h$ has min-entropy at least $p(n) - (\alpha(n) + \omega(\log n))$, where $p(n)$ is the size of the public-key (See Claim 1).

2. For any distribution $\mathcal{D}$ with min-entropy at least $p(n) - (n+1)\log q + \omega(\log n)$, with high probability, a public-key sampled from $\mathcal{D}$ is meaningless. (See Claim 2).

Together, these two steps mean that RPKE is a meaningful/meaningless encryption scheme even in the presence of auxiliary information h_A, as long as $p(n) - (\alpha(n) + \omega(\log n)) \geq p(n) - (n+1)\log q + \omega(\log n)$, which holds if $\alpha(n) \leq (n+1)\log q - \omega(\log n)$ (which is trivially true). The proof of semantic security against adaptive attacks now follows by an argument similar to the one in Lemma 4, and is omitted.

□

Claim 1. *Assuming that the LWE problem is $2^{\alpha(n)+\omega(\log n)}$-hard, for every circuit $h : \{0,1\}^{s(n)} \to \{0,1\}^{\alpha(n)}$, there is a distribution $\mathcal{D}_h$ such that (a) $\mathcal{D}_h$ is computationally indistinguishable from a public-key* PK *generated by* $\text{RGEN}(1^n)$, even given $h(\text{PK}, \text{SK})$ *and (b) $\mathcal{D}_h$ has min-entropy at least $p(n) - (\alpha(n) + \omega(\log n))$, where $p(n)$ is the size of the public-key.*

Proof. (Sketch.) A public-key PK generated by $\text{RGEN}(1^n)$ is of the form $(\mathbf{A}, \mathbf{As} + \mathbf{x})$ where $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ is a random matrix, $\mathbf{s} \in \mathbb{Z}_q^n$ is a random vector and $\mathbf{x} \in \chi^m$ is a vector each of whose entries is chosen independently at random from the error-distribution χ (think of the parameters m, q and χ as fixed). The

LWE assumption implies that the distribution of $(\mathbf{A}, \mathbf{As}+\mathbf{x})$ is computationally indistinguishable from uniformly random in $\mathbb{Z}_q^{mn+n}$, and in particular that the function $G(\mathbf{A}, \mathbf{s}, \mathbf{x}) = (\mathbf{A}, \mathbf{As}+\mathbf{x})$ is a pseudorandom generator.

We now use a lemma of Dziembowski and Pietrzak, which informally states that the output of any (sufficiently secure) PRG has high HILL-entropy, even given the result of a function h (with bounded output length) applied on the seed of the PRG. A formal statement of (a variant of) [11, Lemma 3], specialized to the case of RPKE, is given in Lemma 1.

The claim follows as a direct consequence of this lemma, applied to the PRG G and the function $h(\text{PK}, \text{SK}) = h(\underbrace{\mathbf{A}, \mathbf{As}+\mathbf{x}}_{=\text{PK}}, \underbrace{\mathbf{s}}_{=\text{SK}}) = \widetilde{h}(\mathbf{A}, \mathbf{s}, \mathbf{x})$ (for some related $\widetilde{h}$). □

Lemma 1. ([11, Lemma 3], simplified) *Let $G : \{0,1\}^n \to \{0,1\}^m$ be any $2^{\alpha(n)+\omega(\log n)}$-secure PRG. Then, for every function $h : \{0,1\}^n \to \{0,1\}^{\alpha(n)}$, there exists a distribution $\mathcal{D}_h$ with min-entropy at least $m - (\alpha(n) + \omega(\log n))$ such that*

$$(G(U_n), h(U_n)) \approx_c (y \leftarrow \mathcal{D}_h, h(U_n))$$

Claim 2. *For any distribution $\mathcal{D}$ with min-entropy at least $p(n) - (n+1)\log q + \omega(\log n)$, with high probability, a public-key sampled from $\mathcal{D}$ is meaningless for the encryption scheme* RPKE.

Proof. (Sketch.) Peikert, Vaikuntanathan and Waters [33, Lemma 7.4] prove that the number of meaningless public-keys in RPKE is very large. A quantitative version of their lemma is given as Lemma 2. In particular, Lemma 2 implies that the number of meaningless public-keys is at least $2^{p(n)}(1 - \frac{1}{q^{n+1}})$. Thus, if the min-entropy of $\mathcal{D}$ is at least $p(n) - (n+1)\log q + \omega(\log n)$, then a sample from $\mathcal{D}$ will be a meaningless public-key $1 - \mathsf{negl}(n)$ fraction of the time. □

Lemma 2. ([33, Lemma 7.4], simplified) *Let $m > 3(n+1)\log q$. Then we have*

$$\Pr_{\text{PK}}[\mathsf{SD}(\text{PK}, U_{p(n)}) > q^{-(n+1)/2}] \leq 1/q^{n+1}$$

The probability is taken over a uniformly random choice of PK *from $2^{p(n)}$, where $p(n)$ is the bit-length of* PK.

4 Simultaneous Hardcore Bits

In this section, we show that for the trapdoor (injective) one-way function proposed recently by Gentry, Peikert and Vaikuntanathan [13] (based on the one-way function $\mathcal{F}_{\mathsf{LWE}}$ of Regev [35]), $n - o(n)$ bits of the input are simultaneously hardcore (where n is the length of the input). The one-wayness of the function is based on the hardness of the learning with error (LWE) problem (in this paper, we are not concerned with exactly how the trapdoor inversion works; therefore, we refrain from describing it).

We remark that the exact same proof can also be used to show that $n - o(n)$ bits are simultaneously hardcore for the one-way function $\mathcal{F}_{\mathsf{LPN}}$ based on the hardness of learning parity with noise (in $GF(2)$). We do not discuss this extension further in this paper.

The rest of this section is devoted to proving the following theorem.

Theorem 3. *Let $f_{\mathbf{A}}(\mathbf{s}, \mathbf{x}) = \mathbf{Ax} + \psi$, where $\mathbf{A}$ is a random matrix in $\mathbb{Z}_q^{m(n)\times n}$, $\mathbf{s}$ is a random vector in $\mathbb{Z}_q^n$ and $\mathbf{x}$ is a vector where each component x_i is chosen independently from the error-distribution $\chi(n)$. Then, for every subset $S \subseteq [n]$, $\mathbf{s}|_S$ is simultaneously hardcore for $f_{\mathbf{A}}$, assuming that the LWE problem with dimension $n - |S|$ is hard for polynomial-time algorithms. That is, $(\mathbf{A}, \mathbf{As}+\mathbf{x}, \mathbf{s}|_S) \approx_c (\mathbf{A}, \mathbf{As}+\mathbf{x}, U_{\mathbb{Z}_q^{|S|}})$.*

Proof. We show this by a hybrid argument. **Hybrid** H_0 denotes the distribution $(\mathbf{A}, \mathbf{Ax} + \psi, \mathbf{x}|_S)$. This is the distribution on the left in the statement of the theorem. **Hybrid** H_1 denotes the distribution $(\mathbf{A}, U_{\mathbb{Z}_q^m}, U_{\mathbb{Z}_q^{|S|}})$. Note that all the components of H_2 are uniformly random (and independent) in their respective domains. **Hybrid** H_2 denotes the distribution $(\mathbf{A}, \mathbf{Ax}+\psi, U_{\mathbb{Z}_q^{|S|}})$. This is the distribution on the right in the statement of the theorem.

We will show that $H_0 \approx_c H_1$ (Claim 3) and (2) $H_1 \approx_c H_2$ (Claim 4), which proves the theorem. □

Claim 3. $H_0 \approx_s H_1$.

Proof. We want to show that $(\mathbf{A}, \mathbf{As} + \mathbf{x}, \mathbf{s}|_S) \approx_c (\mathbf{A}, U_{\mathbb{Z}_q^m}, U_{\mathbb{Z}_q^{|S|}})$. We will show this by contradiction: suppose a PPT algorithm D distinguishes between the two distributions. Then, we construct a PPT algorithm E that breaks the LWE assumption with security parameter $n - |S|$.

E gets as input $(\mathbf{A}', \mathbf{y}')$ where $\mathbf{A}'$ is uniformly random in $\mathbb{Z}_q^{m\times(n-|S|)}$ and works as follows: E first sets $\mathbf{A}_{\bar{S}} = \mathbf{A}'$, picks $\mathbf{A}_S$ at random from $\mathbb{Z}_q^{m\times|S|}$ and sets $\mathbf{A} = [\mathbf{A}_S, \mathbf{A}_{\bar{S}}]$. E also picks $\mathbf{s}_S \leftarrow \mathbb{Z}_q^{|S|}$ uniformly at random and computes $\mathbf{y} = \mathbf{y}' + \mathbf{A}_S\mathbf{s}_S$. E then runs D with input $(\mathbf{A}, \mathbf{y}, \mathbf{s}_S)$, and outputs whatever D outputs.

We show that E distinguishes between the case where $y' = \mathbf{A}'\mathbf{s}'+\mathbf{x}$ and where y' is a uniformly random string of the same length. If $(\mathbf{A}', \mathbf{y}')$ is distributed according to LWE (that is, it is of the form $(\mathbf{A}', \mathbf{A}'\mathbf{s}'+\mathbf{x})$) then $(\mathbf{A}, \mathbf{y} = \mathbf{y}' + \mathbf{A}_S\mathbf{x}_S)$ is distributed identical $(\mathbf{A}, \mathbf{As} + \mathbf{x})$, that is an LWE distribution with dimension n. In this case, the input to D is distributed identical to H_0.

On the other hand, if $(\mathbf{A}', \mathbf{y}')$ is uniformly random, then $(\mathbf{A}, \mathbf{y} = \mathbf{y}' + \mathbf{A}_S\mathbf{x}_S, \mathbf{x}_S)$ consists of uniformly random and independent entries (which is exactly H_1). Thus, E distinguishes between LWE with security parameter $n - |S|$ and uniform, at least as often as D distinguishes between H_0 and H_1.

Since distinguishing between the LWE distribution and uniform is equivalent to breaking the LWE assumption [35], we are done. □

Claim 4. $H_1 \approx_s H_2$.

Proof. We want to show that $(\mathbf{A}, \mathbf{As} + \mathbf{x}, U_{\mathbb{Z}_q^{|S|}}) \approx_c (\mathbf{A}, U_{\mathbb{Z}_q^m}, U_{\mathbb{Z}_q^{|S|}})$. It is easy to see that this is equivalent to distinguishing between the LWE distribution and uniform, which by [35] is equivalent to solving LWE with security parameter n. □

Open Questions. In this paper, we design public-key and identity-based encryption schemes that are secure against freezing attacks. The first question that arises from our work is whether it is possible to (define and) construct other cryptographic primitives such as signature schemes, identification schemes and even protocol tasks that are secure against freezing attacks. The second question is whether it is possible to protect against freezing attacks that measure an arbitrary polynomial number of bits. Clearly, this requires some form of (randomized) refreshing of the secret-key, and it would be interesting to construct such a mechanism. Finally, it would be interesting to improve the parameters of our construction, as well as the complexity assumptions, and also to design encryption schemes against freezing attacks under other cryptographic assumptions.

Acknowledgments. The third author would like to gratefully acknowledge delightful discussions with Rafael Pass about the simultaneous hardcore bits problem in the initial stages of this work.

References

[1] Werner Alexi, Benny Chor, Oded Goldreich, and Claus-Peter Schnorr. Rsa and rabin functions: Certain parts are as hard as the whole. *SIAM J. Comput.*, 17(2):194–209, 1988. 1

[2] Avrim Blum, Merrick L. Furst, Michael J. Kearns, and Richard J. Lipton. Cryptographic primitives based on hard learning problems. In *CRYPTO*, pages 278–291, 1993. 2, 15

[3] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, 2003. 8, 15

[4] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM J. Comput.*, 13(4):850–864, 1984. 1

[5] Ran Canetti, Dror Eiger, Shafi Goldwasser, and Dah-Yoh Lim. How to protect yourself without perfect shredding. In *ICALP (2)*, pages 511–523, 2008. 3

[6] Dario Catalano, Rosario Gennaro, and Nick Howgrave-Graham. Paillier's trapdoor function hides up to () bits. *J. Cryptology*, 15(4):251–269, 2002. 1

[7] Don Coppersmith. Small solutions to polynomial equations, and low exponent rsa vulnerabilities. *J. Cryptology*, 10(4):233–260, 1997. 3

[8] Ivan Damgård and Jesper Buus Nielsen. Perfect hiding and perfect binding universally composable commitment schemes with constant expansion factor. In *CRYPTO*, pages 581–596, 2002. 5

[9] 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. 16

[10] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *EUROCRYPT*, pages 523–540, 2004. 16

[11] Stefan Dziembowski and Krysztof Pietrzak. Leakage-resilient stream ciphers. In *To Appear in the IEEE Foundations of Computer Science*, 2008. 2, 11

[12] Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography in the standard model. In *FOCS*, 2008. 6

[13] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206, 2008. 1, 3, 11, 18

[14] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *STOC*, pages 25–32, 1989. 1

[15] Oded Goldreich and Vered Rosen. On the security of modular exponentiation with application to the construction of pseudorandom generators. *Journal of Cryptology*, 16:2003, 2000. 1

[16] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. One-time programs. In *CRYPTO*, pages 39–56, 2008. 2, 6

[17] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984. 1, 6

[18] Alex Halderman, Seth Schoen, Nadia Heninger, William Clarkson, William Paul, Joseph Calandrino, Ariel Feldman, Jacob Appelbaum, and Edward Felten. Lest we remember: Cold boot attacks on encryption keys. In *Usenix Security Symposium*, 2008. 1, 2, 3

[19] Johan Håstad and Mats Näslund. The security of individual rsa bits. In *FOCS*, pages 510–521, 1998. 1

[20] Johan Håstad, A. W. Schrift, and Adi Shamir. The discrete logarithm modulo a composite hides o(n) bits. *J. Comput. Syst. Sci.*, 47(3):376–404, 1993. 1

[21] Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In *CRYPTO*, pages 463–481, 2003. 2, 3

[22] Burton S. Kaliski. A pseudo-random bit generator based on elliptic logarithms. In *CRYPTO*, pages 84–103, 1986. 1

[23] Paul C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *CRYPTO*, pages 104–113, 1996. 2

[24] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *CRYPTO*, pages 388–397, 1999. 2

[25] Gillat Kol and Moni Naor. Cryptography and game theory: Designing protocols for exchanging information. In *TCC*, pages 320–339, 2008. 5, 9, 17, 18

[26] Douglas L. Long and Avi Wigderson. The discrete logarithm hides o(log n) bits. *SIAM J. Comput.*, 17(2):363–372, 1988. 1

[27] Side-Channel Cryptanalysis Lounge, 2008. `http://www.crypto.rub.de/en_sclounge.html`. 2

[28] Vadim Lyubashevsky. The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem. In *APPROX-RANDOM*, pages 378–389, 2005. 15

[29] Silvio Micali and Leonid Reyzin. Physically observable cryptography (extended abstract). In *TCC*, pages 278–296, 2004. 2, 6

[30] Dag Arne Osvik, Adi Shamir, and Eran Tromer. Cache attacks and countermeasures: The case of aes. In *CT-RSA*, pages 1–20, 2006. 2

[31] Pascal Paillier. A trapdoor permutation equivalent to factoring. In *Public Key Cryptography*, pages 219–222, 1999. 1

[32] Sarvar Patel and Ganapathy S. Sundaram. An efficient discrete log pseudo random generator. In *CRYPTO*, pages 304–317, 1998. 1

[33] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. Efficient and composable oblivious transfer. 2008. 4, 9, 11, 17, 18

[34] Christophe Petit, François-Xavier Standaert, Olivier Pereira, Tal Malkin, and Moti Yung. A block cipher based pseudo random number generator secure against side-channel key recovery. In *ASIACCS*, pages 56–65, 2008. 2

[35] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93, 2005. 1, 2, 3, 4, 5, 8, 10, 11, 12

[36] Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. Cryptology ePrint Archive, Report 2008/116, 2008. 3

[37] Umesh V. Vazirani and Vijay V. Vazirani. Efficient and secure pseudo-random number generation. In *CRYPTO*, pages 193–202, 1984. 1

[38] Andrew C. Yao. Theory and application of trapdoor functions. *Symposium on Foundations of Computer Science*, 0:80–91, 1982. 1

A Some More Definitions

Learning Parity With Noise (LPN). The learning parity with noise assumption was first formulated and used for cryptographic purposes in the work of Blum, Furst, Kearns and Lipton [2].

Consider the family of functions $\mathcal{F}_{\mathsf{LPN}}$, parametrized by numbers $m(n) \in \mathbb{N}$ and $p(n) \in [0, \frac{1}{2})$, defined the following way: Let n be a security parameter. Each function $f_{\mathbf{A}}$ is indexed by a matrix $\mathbf{A} \in \mathbb{Z}_2^{m \times n}$. The input of $f_{\mathbf{A}}$ is $(\mathbf{s}, \mathbf{x})$ where $\mathbf{s}$ is chosen uniformly at random from $\mathbb{Z}_2^n$ and $\mathbf{x} = (x_1, \ldots, x_m)$ is chosen such that each x_i is 1 with probability $p = p(n)$ and 0 otherwise. The output is $f_{\mathbf{A}}(\mathbf{s}, \mathbf{x}) = \mathbf{A}\mathbf{s} + \mathbf{x}$.

The hardness of LPN is parametrized chiefly by the dimension n. Therefore, we let all other parameters (m and p) be functions of n, often omitting the explicit dependence for notational clarity. For our purposes, we will be concerned with $m(n)$ being a polynomial in n, and $p(n)$ being the inverse of a polynomial in n.

We say that the $(m(n), p(n))$-LPN problem is $t(n)$-hard if for every family of circuits Adv of size at most $t(n)$,

$$\Pr[\mathsf{Adv}(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{x}) = \mathbf{s}] \leq \frac{1}{t(n)}$$

where the probability is over the choice of a random $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, random $\mathbf{s} \in \mathbb{Z}_q^n$ and a vector $\mathbf{x} = (x_1, \ldots, x_m)$ is chosen such that each x_i is 1 with probability $p = p(n)$ and 0 otherwise.

In other words, the assumption says that $f_{\mathbf{A}}$ (for a randomly chosen $\mathbf{A}$) is a one-way function against adversaries of size $t(n)$. Furthermore, [2] showed that if $f_{\mathbf{A}}$ is a one-way function, then it is a pseudorandom generator as well (for $m(n) > n$).

The best algorithm to solve LPN runs in time $O(2^{O(n/\log n)})$ [3, 28].

B Techniques: Min-Entropy and Randomness Extraction

If X is a random variable, we will also denote by X the probability distribution on the range of the variable. We let U_m denote the uniform distribution on $\mathbb{Z}_p^m$ (note that the support of this distribution is $\mathbb{Z}_p^m$, rather than the usual $\{0,1\}^m$). If a random variable occurs twice in the same expression, it means that the same value is used in both, rather than two independent samples.

A standard measure of the (worst-case) entropy of a random variable X is its min-entropy $\mathsf{H}_\infty(X)$, defined as $\mathsf{H}_\infty(X) = -\log(\max_{d \in D} \Pr[X = d])$. In addition, we would like to define a notion of "conditional min-entropy" of a random-variable X given a possibly correlated random variable Y. The notion that is most appropriate for our purposes is called average min-entropy $\tilde{\mathsf{H}}_\infty(X|Y)$ defined by Dodis, Reyzin and

Smith [10]:

$$\tilde{\mathsf{H}}_\infty(X|Y) = -\log \mathbb{E}_{d\in D} \max_x \Pr[X = x|Y = d] = -\log \mathbb{E}_{d\leftarrow D} 2^{-\mathsf{H}_\infty(X|Y=y)}$$

Average min-entropy satisfies the following weak chain-rule [9]: if the support of Y is of size at most 2^λ, then $\mathsf{H}_\infty(X|Y) \geq \mathsf{H}_\infty(X) - \lambda$. Roughly, this can be interpreted as saying that revealing any λ bits of information about X causes its min-entropy to go down by at most λ. More generally,

Proposition 2. *For any random variables X and Y, $\mathsf{H}_\infty(X|Y) \geq \mathsf{H}_\infty(X) - \log(|\mathsf{Supp}(Y)|)$, where $\mathsf{Supp}(Y)$ is the support of the random variable Y.*

If X and Y are distributions with a common support D, then let $\mathsf{SD}(X, Y)$ denote the statistical distance between distributions X and Y. That is,

$$SD(X, Y) = \frac{1}{2} \sum_{d\in D} |\Pr[X = d] - \Pr[Y = d]|$$

We need the following standard fact about how statistical distance changes when a function is applied to a random variable.

Proposition 3. *For any two random variables X and Y and any (possibly randomized) function f, $\mathsf{SD}(f(X), f(Y)) \leq \mathsf{SD}(X, Y)$.*

Matrix Multiplication and Randomness Extraction. Consider the family of functions $\mathcal{H} = \{h_C : \mathbb{Z}_p^n \to \mathbb{Z}_p^\ell\}_{C\in\mathbb{Z}_p^{n\times\ell}}$ defined by $h_C(\mathbf{x}) = C \cdot \mathbf{x}$. We show that this family of functions defines a good randomness extractor. Note that this is not a universal (or even almost-universal) family of hash functions, and therefore, the leftover hash lemma does not apply directly.

Proposition 4. *Let $\mathcal{H}$ be the family of pairwise independent hash functions from $\mathbb{Z}_p^n \to \mathbb{Z}_p^\ell$. Let X and Y be random variables such that $X \in \mathbb{Z}_p^n$ and $\tilde{\mathsf{H}}_\infty(X|Y) \geq k$. Then, for $h_C \leftarrow \mathcal{H}$, we have*

$$\mathsf{SD}((Y, C, h_C(X)), (Y, C, U_\ell)) \leq \epsilon$$

as long as $\ell \leq k - \log q - 2\log(1/\epsilon)$.

The proof of this lemma is essentially the same as in [10], and we omit it: the only difference in the statement of the lemma is the extra $\log q$ factor, which comes in because we do not have a pairwise independent hash function at hand.

C Security Against Non-Adaptive Freezing Attacks

Here, we give the full proof of Theorem 1.

Theorem 4. *The public-key encryption scheme* RPKE′ *is secure against a non-adaptive $\alpha(n)$-freezing attack, assuming that the LWE problem with security paramter (dimension) $(n - \alpha(n) - \omega(\log n))$ is hard for polynomial-time algorithms. (where the implicit parameters $m(n)$ and $q(n)$ in the LWE definition are both* polynomial in n*)*

Proof. First, in Lemma 3, we show that the meaningful and meaningless public-keys of RPKE′ are computationally indistinguishable, even given some information about the (real) secret-key. In particular, we show that for any function $h : \mathbb{Z}_q^n \to \mathbb{Z}_q^{\alpha(n)}$,

$$\{(\text{PK}, h(\text{SK})) : (\text{PK}, \text{SK}) \leftarrow \text{RGEN}'(1^n)\} \approx_c \{(\text{PK}', h(\text{SK})) : \text{PK}' \leftarrow \text{RBADGEN}'(1^n), (\text{PK}, \text{SK}) \leftarrow \text{RGEN}'(1^n)\}$$

Secondly, in Lemma 4 we show that if the meaningful and meaningless keys are computationally indistinguishable given $h(\text{SK})$ for any $h : \mathbb{Z}_q^n \to \mathbb{Z}_q^{\alpha(n)}$, then the encryption scheme is semantically secure against non-adaptive $\alpha(n)$-freezing attacks as well. The proof of Lemma 4 essentially follows from similar lemmas that appear in [25, 33].

The proof of the theorem follows from Lemmas 3 and 4. □

Lemma 3. *Let $k(n) \in \mathbb{N}$ and let $h : \mathbb{Z}_q^n \to \mathbb{Z}_q^{\alpha(n)}$ be any function. Assuming that LWE with security parameter $(n - \alpha(n) - \omega(\log n))$ is hard for polynomial time algorithms, a random meaningful public-key of* RPKE′ *is computationally indistinguishable from a random meaningless public-key, even given $h(\text{SK})$, where* SK *is the real secret-key. In particular,*

$$\{(\text{PK}, h(\text{SK})) : (\text{PK}, \text{SK}) \leftarrow \text{RGEN}'(1^n)\} \approx_c \{(\text{PK}', h(\text{SK})) : \text{PK}' \leftarrow \text{RBADGEN}'(1^n), (\text{PK}, \text{SK}) \leftarrow \text{RGEN}'(1^n)\}$$

Proof. What we need to prove is the following: for any function $h : \mathbb{Z}_q^n \to \mathbb{Z}_q^{\alpha(n)}$, given a random matrix $\mathbf{A}$, $\mathbf{As} + \mathbf{x}$ is pseudorandom, given $h(\mathbf{s})$. That is,

$$(\mathbf{A}, \mathbf{As} + \mathbf{x}, h(\mathbf{s})) \approx_c (\mathbf{A}, U_{\mathbb{Z}_q^m}, h(\mathbf{s}))$$

First, consider the distribution of $\mathbf{s}$, conditioned on $h(\mathbf{s})$. We claim that this distribution has (average) min-entropy $(n - \alpha(n)) \log q$. From Proposition 2,

$$\mathsf{H}_\infty(\mathbf{s} \mid h(\mathbf{s})) \geq \mathsf{H}_\infty(\mathbf{s}) - \log |\mathbb{Z}_q^{\alpha(n)}| \geq (n - \alpha(n)) \log q$$

Now, by the leftover hash lemma (using matrix multiplication as the hash function) we get (by Proposition 4) that

$$(\mathbf{C}, \mathbf{Cs}, h(\mathbf{s})) \approx_s (\mathbf{C}, U_{\mathbb{Z}_q^{k(n)}}, h(\mathbf{s}))$$

for a random $\mathbf{C} \in \mathbb{Z}_q^{k(n) \times n}$ and a random $\mathbf{s} \in \mathbb{Z}_q^n$ as long as $k(n) \leq n - \alpha(n) - \omega(\log n)$.

By Proposition 3, $(\mathbf{B}, \mathbf{C}, \mathbf{BCs} + \mathbf{x}, h(\mathbf{s})) \approx_s (\mathbf{B}, \mathbf{C}, \mathbf{Bt} + \mathbf{x}, h(\mathbf{s}))$ where $\mathbf{t}$ is uniformly random and independent of all other components. Using the fact that $\mathbf{A} = \mathbf{BC}$, this means that

$$(\mathbf{A}, \mathbf{As} + \mathbf{x}, h(\mathbf{s})) \approx_s (\mathbf{A}, \mathbf{Bt} + \mathbf{x}, h(\mathbf{s})) \approx_c (\mathbf{A}, U_{\mathbb{Z}_q^m}, h(\mathbf{s}))$$

where the second indistinguishability follows, using the LWE assumption with security parameter $k(n)$, and the fact that $(\mathbf{A}, \mathbf{At} + \mathbf{x})$ is statistically independent of $\mathbf{s}$.

This proves the lemma. □

Lemma 4. *Let* PKE $= (\text{GEN}, \text{ENC}, \text{DEC})$ *be any meaningful/meaningless public-key encryption scheme.* PKE *is secure against non-adaptive $\alpha(n)$-freezing attacks if meaningful and meaningless public-keys are computationally indistinguishable given $h(\text{SK})$, for every $h : \{0,1\}^{s(n)} \to \{0,1\}^{\alpha(s(n))}$ (where $s(n)$ is the size of the secret-key* SK*). That is, if*

$$\{(\text{PK}, h(\text{SK})) : (\text{PK}, \text{SK}) \leftarrow \text{GEN}(1^n)\} \approx_c \{(\text{PK}', h(\text{SK})) : \text{PK}' \leftarrow \text{BADGEN}(1^n), (\text{PK}, \text{SK}) \leftarrow \text{GEN}(1^n)\}$$

Proof. (Sketch.) The proof follows the ideas of [25, 33]. We sketch the proof here. By the definition of meaningless public-keys, even an unbounded adversary cannot distinguish between the encryptions of any two messages m_0 and m_1 under a meaningless public-key PK' (and this is true even given $h(\text{SK})$, since SK a random secret-key, sampled independently of PK'). Now, if the adversary could distinguish between the encryptions of m_0 and m_1 under a meaningful public-key PK (when given $h(\text{SK})$ for the matching secret-key SK) then it must mean that the adversary distinguishes between meaningful and meaningless public-keys, given $h(\text{SK})$. Thus, the encryption scheme is secure against non-adaptive freezing attacks, if the two distributions in the statement of the lemma are computationally indistinguishable. □

D Identity-Based Encryption Secure Against Freezing Attack

In this section, we construct identity-based encryption schemes that are secure against memory-freezing attacks. We show that the identity-based encryption scheme of Gentry, Peikert and Vaikuntanathan [13] is semantically secure against both a non-adaptive and an adaptive $\alpha(m)$-freezing attack, where $\alpha(m) \leq m - o(m)$, where m is the length of the secret-key. We show the semantic security of this encryption scheme against $\alpha(m)$-freezing attacks, under the standard LWE assumption, namely that the LWE problem with security parameter n is hard for polynomial-time algorithms.

First, we describe the [13] IBE.

The GPV ID-based Encryption Scheme. Let n be the security parameter and let $q(n), m(n), k(n) \in \mathbb{N}$ and the probability distribution $\chi(n)$ over $\mathbb{Z}_q$ be parameters of the system.

The master-public key in the encryption scheme is $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, the public-key PK corresponding to an identity id is $\mathbf{y} = H(\mathsf{id}) \in \mathbb{Z}_q^n$ (where H is a random oracle) and the secret-key SK is a vector $\mathbf{r} \in \{0,1\}^m$ such that $\mathbf{y} = \mathbf{r}\mathbf{A}$. [6]

- $\text{GPVKEYDER}_{MSK}(\mathsf{id})$ lets $\mathbf{y} = H(\mathsf{id})$ and outputs SK_{id} to be a vector $\mathbf{r} \in \{0,1\}^m$ such that $\mathbf{r}\mathbf{A} = \mathbf{y}$.
- $\text{GPVENC}(\mathsf{id}, b)$, where b is a bit, works as follows. Let $\text{PK} = h(\mathsf{id})$. Pick a random vector $\mathbf{s} \in \mathbb{Z}_q^n$ and let the vector $\mathbf{x} = (x_1, \ldots, x_m)$ and the number η consist of values chosen independently from the probability distribution χ. Output the ciphertext $(\mathbf{A}\mathbf{s} + \mathbf{x}, \mathbf{y}^T\mathbf{s} + \eta + b\lfloor \frac{q}{2} \rfloor)$.
- $\text{GPVDEC}(\text{SK}, c)$ first parses $c = (\mathbf{c}_0, c_1)$, computes $b' = c_1 - \mathbf{r}^T\mathbf{c}_0$ and outputs 0 if b' is closer to 0 than to $\frac{q}{2}$, and 1 otherwise.

D.1 Security Against Non-adaptive Freezing Attacks

Theorem 5. *The identity-based encryption scheme* GPVIBE *is secure against a non-adaptive and adaptive $\alpha(m)$-freezing attack (where m is the bit-length of the secret-key) for $\alpha(m) \leq m - n\log q - \omega(\log n)$, assuming that* LWE *with security parameter n is hard for polynomial time algorithms.*

Proof. (Sketch.) We will prove this in two steps. First, we show that security against non-adaptive freezing attacks follows from the fact that the distribution of $H(\mathsf{id})$ for any id is statistically close to random, even

[6] The actual GPV secret-key derivation algorithm does not produce secret keys $\mathbf{r} \in \{0,1\}^m$ but only $\mathbf{r}$'s such that the length $||r||_2$ is small. This complicates the precise quantitative statements of our theorems. For the sake of the present paper, we assume that $\mathbf{r} \in \{0,1\}^m$, to allow for simpler to state theorems.

given $n - o(n)$ bits of the secret-key (this is similar to the proof of 4) and is omitted). Secondly, we show that this is the case.

The proof of the latter claim follows from leftover hash lemma and the fact that matrix multiplication is a good extractor. If the adversary measures $\alpha(m)$ bits of $\mathbf{r}$, then we are still left with $m - \alpha(m)$ bits of min-entropy. That is, at least $m - (m - n \log q - \omega(\log n)) \geq n \log q + \omega(m))$ bits of min-entropy. Now, since the output of $\mathbf{rA}$ is $n \log q$ bits, it follows from the left-over hash lemma that $(\mathbf{A}, \mathbf{rA})$ is statistically close to uniform even given $h(\mathbf{r})$.

To handle an adaptive attack, note that if the h function measures both PK (i.e, $(\mathbf{A}, \mathbf{rA})$) and SK (i.e, $\mathbf{r}$) then the distribution of PK together with $h(\text{PK}, \text{SK})$ is no longer uniform but has large min-entropy. Nevertheless, we can show that the encryption scheme is secure even with such a public-key, assuming exponential hardness of LWE.

We omit the details. □