Leakage Resilient Fully Homomorphic Encryption

Alexandra Berkoff* Feng-Hao Liu†

Abstract

We construct the first leakage resilient variants of fully homomorphic encryption (FHE) schemes. Our leakage model is bounded adaptive leakage resilience. We first construct a leakage-resilient leveled FHE scheme, meaning the scheme is both leakage resilient and homomorphic for all circuits of depth less than some pre-established maximum set at the time of key generation. We do so by applying ideas from recent works analyzing the leakage resilience of public key encryption schemes based on the decision learning with errors (*DLWE*) assumption to the Gentry, Sahai and Waters ([17]) leveled FHE scheme. We then move beyond simply leveled FHE, removing the need for an a priori maximum circuit depth, by presenting a novel way to combine schemes. We show that by combining leakage resilient leveled FHE with multi-key FHE, it is possible to create a leakage resilient scheme capable of homomorphically evaluating circuits of arbitrary depth, with a bounded number of distinct input ciphertexts.

*aberkoff@cs.brown.edu

†fenghao@cs.umd.edu (Most of the work was completed while author was a graduate student at Brown University.)

1 Introduction and Related Work

Fully homomorphic encryption is a way of encrypting data that allows a user to perform arbitrary computation on that data without decrypting it first. The problem of creating a fully homomorphic encryption scheme was suggested by Rivest, Adleman, and Dertouzos in 1978 [34]. It has received renewed attention in recent years and has obvious applicability to cloud computing— If a user stores her data on someone else's servers, she may wish to store her data encrypted under a public key encryption scheme, yet still take advantage of that untrusted server's computation power to work with her data.

The first candidate for fully homomorphic encryption was proposed by Gentry in 2009 [16]. Since then, candidate schemes have been based on a variety of computational assumptions (see, for example: [38, 37, 10, 8]) including the decision learning with errors (*DLWE*) assumption [5, 6, 7, 17]. The latest *DLWE*-based work is due to Gentry, Sahai, and Waters (GSW) [17], and it is this work we focus most closely on in our paper.

We note that public key encryption schemes based on the *DLWE* assumption have typically been based on one of two schemes both described by Regev in the latest version of [33]. Regev originally constructed so-called "primal Regev" (referred to in this work as RPKE) and Gentry, Peikert, and Vaikuntanathan constructed so-called "dual Regev" [**?**] in 2008. The instantiations in the papers describing all the *DLWE*-based homomorphic schemes cited above use "primal Regev" as a building block. The Regev schemes have also been used as building blocks to achieve identity based encryption, attribute based encryption, and, as described in Section 1.2, leakage resilient encryption.

The term "leakage resilience" is meant to capture the security of a cryptographic algorithm when an adversary uses non-standard methods to learn about the secret key. Typically in security proofs, attackers are modeled as probabilistic polynomial time machines with only input/output access to the given cryptographic algorithm. Leakage resilience is a theoretical framework for addressing security when an attacker learns information about the secret key not obtainable through the standard interface, for example by obtaining physical access to a device, or by identifying imperfect or correlated randomness used in secret key generation.

Starting with the work of Ishai, Sahai and Wagner [22], and Micali and Reyzin [26], the cryptographic community has worked towards building general theories of security in the presence of information leakage. This has been an active topic of research over the past 15 years (see [1, 2, 4, 9, 11, 12, 13, 15, 19, 30, 32, 35, 22, 26] and the references therein), resulting in many different leakage models, and cryptographic primitives such as public key encryption schemes and signature schemes secure in each model.

In our work, we, for the first time, apply the framework of leakage resilience to fully homomorphic schemes.

1.1 Non-Adaptive Leakage on FHE

We start with the observation that the Decision Learning With Errors problem is, with appropriate parameter settings, leakage resilient – Goldwasser, Kalai, Peikert and Vaikuntanathan showed that the *DLWE* problem with a binary secret, and a carefully chosen bound on the size of the error term, with a leakage function applied to the secret, reduces from a *DLWE* problem with smaller dimension, modulus, and error bound, but no leakage [18]. Recently, Alwen, Krenn, Pietrzak, and Wichs extended this result to apply to a wider range of secrets and error bounds [3].

Since many FHE schemes (for example [5, 6, 7, 17]) can be instantiated based on the *DLWE* assumption, an obvious first attempt to create leakage resilient FHE is to directly apply those results by instantiating an FHE scheme with parameters that make the underlying *DLWE* problem leakage resilient. Indeed, doing so leads immediately to non-adaptive leakage resilient FHE. We describe these results in Appendix C.

We note as well that the leakage resilience of *DLWE* leads to leakage resilient symmetric-key encryption [18], and closely related results lead to non-adaptive leakage resilience of RPKE [1].

The differentiation between adaptive and non-adaptive leakage is crucial. In the non-adaptive leakage model, an adversary can learn any arbitrary (poly-time computable, bounded output-length) function of the secret key, with the caveat that he cannot adaptively choose the function based on the scheme's public key. This leakage model is not entirely satisfactory, as typically one assumes that if a value is public, everyone, including the adversary will be able to see it at all times. In contrast, the adaptive leakage resilience model assumes that an adversary has full access to all the scheme's public parameters, and can choose its leakage function accordingly.

1.2 Adaptive Leakage on Leveled FHE

Given the gap between the non-adaptive leakage resilience model and the expected real-life powers of an adversary, in this work we primarily consider the adaptive bounded memory leakage model. The model is described in, for example, the works [1, 2]. Since an adversary can choose its leakage function after seeing the public key(s), in effect we consider functions that leak on the public and secret keys together. This framework has been previously considered for non-homomorphic public key and identity based encryption schemes based on bilinear groups, lattices, and quadratic residuosity [2, 35, 23]. Additionally, both RPKE and "dual Regev", schemes based on *DLWE*, can be made leakage resilient; Akavia, Goldwasser, and Vaikunatanathan achieve adaptive leakage-resilient RPKE [1], and Dodis, Goldwasser, Kalai, Peikert, and Vaikuntanathan construct leakage-resilient "dual Regev" [11]. In fact, the latter scheme is secure against auxiliary input attacks—essentially, they consider a larger class of leakage functions—ones whose output length has no bound, but which no probabilistic polynomial time adversary can invert with non-negligible probability.

Unfortunately, the non-adaptive leakage resilient scheme described in Appendix C does not lead in a straightforward way to an adaptively leakage resilient scheme. The crux of the problem is that the public key is a function of the secret key, and when an adversary has leakage access to both the public and secret keys, it can choose a function which simply asks if the two are related. Existing proofs of security for *DLWE*-based FHE schemes all start by proving the public key indistinguishable from random, and such leakage functions make this impossible.

In fact, one might expect the same problem when analyzing the adaptive leakage resilience of RPKE, as the original security proof for this scheme followed the same outline [33]. Akavia, Goldwasser, and Vaikuntanathan (AGV) succeeded in constructing a leakage-resilient variant of RPKE despite this hindrance by writing a new security proof. They directly show that the *ciphertexts* are indistinguishable from random, without making any statements about the *public key* [1].

Inspired by the success of AGV, one might try to use a variation on their technique to prove prove an FHE scheme secure. We note that typically the public key of an FHE scheme consists of two parts: an "encryption key," which is used to generate new ciphertexts, and an "evaluation key," which is used to homomorphically combine the ciphertexts. A strengthening of the AGV technique leads to a secure scheme if the adversary sees the encryption key before choosing its leakage function, but unfortunately the proof fails if it also sees the evaluation key. The evaluation

key is not just a *function of*, but actually an *encryption of* the secret key, and proving security when an adversary could potentially see actual decryptions of some bits of the secret key is a more complicated proposition.

Since the presence of an evaluation key is what hampers the proof, our next step is to apply this technique to a scheme *without* an evaluation key. The first leveled FHE scheme without an evaluation key was recently constructed by Gentry, Sahai, and Waters (GSW) [17]. We strengthen the results of Akavia, Goldwasser, and Vaikuntanathan to apply to a much broader range of parameters, and use this new result to construct LRGSW, a leakage-resilient variant of GSW. We present these results in sections 3 and 4.

1.3 Overcoming the "Leveled" Requirement

Note that so far, we have achieved leakage resilient leveled FHE, meaning we have a scheme where if a maximum circuit depth is provided at the time of key generation, the scheme supports homomorphic evaluation of all circuits up to that depth. In contrast, in a true, non-leveled, fully homomorphic encryption scheme, one should not need to specify a maximum circuit depth ahead of time.

The standard technique for creating a non-leveled FHE scheme, first proposed by Gentry in his original construction, is to first create a "somewhat-homomorphic" encryption scheme (all leveled schemes are automatically "somewhat homomorphic"), make it "bootstrappable" in some way, and then "bootstrap" it to achieve full homomorphism [16]. Although LRGSW is somewhat homomorphic, it needs a separate evaluation key to be bootstrappable. In fact, every known bootstrappable scheme has an evaluation key containing encryptions of the secret key, leaving us back with the same issue we sidestepped by choosing to modify the GSW scheme.

Our key insight is that while we need encryptions of the secret key to perform bootstrapping, these encryption do not need to be part of the public key. We combine a leakage resilient *leveled* FHE scheme with a N-key multi-key FHE scheme in a novel way, which allows us to store these encryptions as part of the ciphertext, letting us achieve a *non-leveled* leakage resilient FHE scheme. We provide an instantiation of this using LRGSW and the López-Alt, Tromer, and Vaikuntanathan multi-key FHE scheme [24]. We discuss these results in section 5. Our contribution is a step towards true fully homomorphic encryption, as we remove the circuit *depth* bound. An artifact of our construction is that the N from our N-key multi-key FHE scheme becomes a bound on the *arity* of our circuit instead. The problem of creating leakage resilient, true FHE is still open, and seems intimately related to the problem of creating true, non-leveled FHE without bootstrapping.

2 Preliminaries

We let bold capital letters (e.g. $\mathbf{A}$) denote matrices, and bold lower-case letters (e.g. $\mathbf{x}$) denote vectors. We denote the inner product of two vectors as either $\mathbf{x} \cdot \mathbf{y}$ or $\langle \mathbf{x}, \mathbf{y} \rangle$.

For a real number x, we let $\lfloor x \rfloor$ be the closest integer $\leq x$, and $\lfloor x \rceil$ be the closest integer to x. For an integer y, we let $[y]_q$ denote $y \mod q$. For an integer N, we let $[N]$ denote the set $\{1, 2, \ldots, N\}$.

We use $x \leftarrow \mathcal{D}$ to denote that x was drawn from a distribution $\mathcal{D}$. We use $x \xleftarrow{\$} S$ to denote that x was drawn uniformly from a set S. To denote computational indistinguishability, we write $\mathcal{X} \approx_c \mathcal{Y}$, and to denote statistical indistinguishability, we write $\mathcal{X} \approx_s \mathcal{Y}$. To denote the statistical

distance between two distributions, we write $\Delta(\mathcal{X}, \mathcal{Y})$. Throughout this work, we use η to denote our security parameter.

In this work, we refer to the **ϵ-smooth average min-entropy** (first defined in [14]) of X conditioned on Y as $\tilde{H}^{\epsilon}_{\infty}(X|Y)$. We refer the reader to Appendix A where we fully define this, and other related concepts of min-entropy, and state versions of the leftover hash lemma that hold true for these concepts.

2.1 Homomorphism

Definition 1. A **homomorphic (public-key) encryption scheme**

$$\mathsf{HE} = (\mathsf{HE.Keygen}, \mathsf{HE.Enc}, \mathsf{HE.Dec}, \mathsf{HE.Eval})$$

is a quadruple of probabilistic polynomial time algorithms as described below:

- **Key Generation**[1] The algorithm $(pk, sk) \leftarrow \mathsf{HE.Keygen}(1^{\kappa})$ takes a unary representation of the security parameter, and outputs a public key pk and a secret decryption key sk.
- **Encryption** The algorithm $c \leftarrow \mathsf{HE.Enc}_{pk}(\mu)$ takes the public key pk and a message $\mu \in \{0,1\}$ and outputs a ciphertext c.
- **Decryption** The algorithm $\mu^* \leftarrow \mathsf{HE.Dec}_{sk}(c)$ takes the secret key sk, a ciphertext c, and outputs a message $\mu^* \in \{0,1\}$.
- **Homomorphic Evaluation** The algorithm $c_f \leftarrow \mathsf{HE.Eval}_{pk}(f, c_1, \ldots, c_t)$ takes the public key, pk, a function $f : \{0,1\}^t \rightarrow \{0,1\}$, and a set of t ciphertexts $c_1, \ldots, c_t$ and outputs a ciphertext c_f. In our paper, we will represent functions f as binary circuits constructed of NAND gates.

Definition 2. For any class of circuits $\mathcal{C} = \{\mathcal{C}_\eta\}_{\eta \in \mathbb{N}}$ over $\{0,1\}$. A scheme HE is $\mathcal{C}$ **$-$ homomorphic** if for any function $f \in \mathcal{C}$, and respective inputs $\mu_1, \ldots, \mu_t \in \{0,1\}$, it holds that

$$Pr[\mathsf{HE.Dec}_{sk}(\mathsf{HE.Eval}_{pk}(f, c_1, \ldots, c_t) \neq f(\mu_1, \ldots, \mu_t)] = negl(\eta)$$

where $(pk, sk) \leftarrow \mathsf{HE.Keygen}(1^{\kappa})$ and $c_i \leftarrow \mathsf{HE.Enc}_{pk}(\mu_i)$.

Definition 3. A homomorphic scheme HE is **compact** if there exists a polynomial $p = p(\eta)$ such that the output length of $\mathsf{HE.Eval}(\cdots)$ is at most p bits long (regardless of f or the number of inputs).

Definition 4. A scheme is **leveled fully homomorphic** if it takes 1^L as additional input in key generation, where $L = poly(\eta)$, and otherwise satisfies the definitions for a compact, $\mathcal{L}$-homomorphic encryption scheme, where $\mathcal{L}$ is the set of all circuits over $\{0,1\}$ of depth $\leq L$.

Definition 5. A scheme is **bounded arity fully homomorphic** if it takes $T = poly(\eta)$ as an additional input in key generation, and is $\mathcal{T}$-homomorphic for $\mathcal{T} = \{\mathcal{T}_\eta\}_{\eta \in \mathbb{N}}$, the set of all arithmetic circuits over $\{0,1\}$ with arity $\leq T$ and depth $poly(\eta)$.

Definition 6. A scheme HE is **fully homomorphic** if it is both compact and $\mathcal{C}$- homomorphic, where $\mathcal{C} = \{\mathcal{C}_\eta\}_{\eta \in \mathbb{N}}$ is the set of all circuits with arity and depth polynomial in η.

[1] In many schemes, the public key is split into two parts, the pk, which is used to encrypt fresh messages, and the evaluation key (evk) that is used to homomorphically evaluate circuits, so the output of the algorithm is: $(pk, evk, sk) \leftarrow \mathsf{HE.Keygen}(1^{\kappa})$.

2.2 Leakage Resilience

Definition 7. Let λ be a non-negative integer. A scheme HE is **adaptively leakage resilient** to λ bits of leakage, if for any PPT adversary $\mathcal{A}$ it holds that

$$\mathbf{ADV}_{ALR^{\lambda}(b=0),ALR^{\lambda}(b=1)}(\mathcal{A}) = negl(\lambda)$$

where the notation $\mathbf{ADV}_{\mathcal{X},\mathcal{Y}}(\mathcal{A}) := |\mathbf{Pr}[\mathcal{A}(\mathcal{X}) = 1] - \mathbf{Pr}[\mathcal{A}(\mathcal{Y}) = 1]|$

and the experiment ALR^{λ} is defined as follows:

1. The challenger generates $(pk, sk) \leftarrow \mathsf{HE.KeyGen}(1^{\eta})$ and sends pk to the adversary.
2. The adversary $\mathcal{A}$ selects a leakage function $h : \{0,1\}^* \rightarrow \{0,1\}^{\lambda}$ and sends it to the challenger.
3. The challenger replies with $h(sk)$.
4. The adversary $\mathcal{A}$ replies with (m_0, m_1)
5. The challenger chooses $b \xleftarrow{\$} \{0,1\}$, computes $c \leftarrow \mathsf{HE.Enc}(pk, m_b)$ and sends c to $\mathcal{A}$.
6. $\mathcal{A}$ outputs $b' \in \{0,1\}$

In the above definition, **adaptive** refers to the fact that $\mathcal{A}$ can choose h after having seen the scheme's public parameters. In fact, an adversary could "hard-code" the scheme's public key into its leakage function, in effect seeing $h(pk, sk)$. In the remainder of this paper, we therefore consider leakage functions that leak on both the public key and the secret key together. There is a corresponding weaker notion of leakage resilience called **non-adaptive** where the adversary must choose h independently of the scheme's public key, and learns only $h(sk)$.

2.3 Learning With Errors

The learning with errors problem (*LWE*), and the related decision learning with errors problem (*DLWE*) were first introduced by Regev [33] in 2005.

Definition 8. The Decision Learning with Errors Problem:
Given a secret $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, $m = poly(n)$ samples $\mathbf{a}_i \xleftarrow{\$} \mathbb{Z}_q^n$, and corresponding noise $x_i \leftarrow \chi$, Distinguish $\{A_{s,\chi}\}_i = \{\mathbf{a}_i, \langle \mathbf{a}_i, \mathbf{s} \rangle + x_i\}_i$ from $\{\mathbf{a}_i, b_i\}_i \xleftarrow{\$} \mathbb{Z}_q^{\ell} \times \mathbb{Z}_q$.
We denote an instance of the problem as $DLWE_{n,q,\chi}$. The **decision learning with errors assumption** is that no probabilistic polynomial time adversary can solve $DLWE_{n,q,\chi}$ with more than negligible advantage.

Definition 9. A family of distributions χ is called **β-bounded** if $\mathbf{Pr}_{x \leftarrow \chi(\eta)}[||x|| > \beta] = negl(\eta)$.

Definition 10. The Gaussian distribution in one dimension with standard deviation β is $D_{\beta} := \exp(-\pi(x/\beta)^2)/\beta$. For $\beta \in \mathbb{Z}_q$, the **discretized Gaussian**, $\overline{\Psi}_{\beta}$, is defined by choosing β' such that $\beta = \beta' \cdot q$, then choosing $x \xleftarrow{\$} D_{\beta'}$ and computing $\lfloor q \cdot x \rceil$. Note that $\overline{\Psi}_{\beta}$ is β-bounded when β is super-polynomial in η. When $\chi = \overline{\Psi}_{\beta}$ we denote the *DLWE* instance as $DLWE_{n,q,\beta}$.

The following statement summarizes much of the recent work analyzing the hardness of *DLWE*.

Statement 1. *(Theorem 1 in [17], due to work of [33, 31, 27, 28])*

Let $q = q(n) \in \mathbb{N}$ be either a prime power or a product of small (size poly(n)) distinct primes, and let $\beta \geq \omega(\log n) \cdot n$ Then there exists an efficiently sampleable β − bounded distribution χ such that if there is an efficient algorithm that solves the average-case LWE problem for parameters n, q, χ, then:

- *There is an efficient quantum algorithm that solves $GapSVP_{\tilde{O}(nq/\beta)}$ on any n-dimensional lattice.*
- *If $q \geq \tilde{O}(2^{n/2})$, there is an efficient classical algorithm for $GapSVP_{\tilde{O}(nq/\beta)}$ on any n-dimensional lattice.*

In both cases, if one also considers distinguishers with sub-polynomial advantage, then we require $\beta \geq \tilde{O}(n)$ and the resulting approximation factor is slightly larger than $\tilde{O}(n^{1.5}q/\beta)$.

The $GapSVP_\gamma$ problem is, given an arbitrary basis of an n dimensional lattice, to determine whether the shortest vector of that lattice has length less than 1 or greater than γ.

Statement 2. *(from [5])*

The best known algorithms for $GapSVP_\gamma$ [36, 29] require at least $2^{\tilde{\Omega}(n/(\log \gamma))}$ time.

These hardness results guide the setting of parameters for our scheme.

3 The LRGSW scheme

We now present LRGSW, an adaptively leakage resilient variant of the Gentry, Sahai, and Waters (GSW) FHE scheme [17]. We box the differences between our scheme and GSW in our description below. The scheme encrypts messages under the "approximate eigenvector" method: For a message $\mu \in \mathbb{Z}_q$, ciphertexts are matrices $\mathbf{C} = \mathsf{Enc}(pk, \mu)$ and have the property that $\mathbf{C} \cdot \mathbf{sk} \approx \mu \cdot \mathbf{sk}$, where $\mathbf{sk}$ is the secret key vector. This means that to homomorphically multiply two ciphertexts $\mathbf{C}_1 = \mathsf{Enc}(pk, \mu_1)$ and $\mathbf{C}_2 = \mathsf{Enc}(pk, \mu_2)$, one simply computes $\mathbf{C}_{mult} = \mathbf{C_1} \cdot \mathbf{C_2}$. Crucially, this intuitive method for homomorphic evaluation removes the need for an "evaluation key" present in other fully homomorphic schemes. Note that for the error-growth reasons Gentry, Sahai, and Waters gave in Section 3.3 of their paper [17], our modification of their scheme is designed to homomorphically evaluate only binary circuits constructed of NAND gates.

3.1 Our Leveled Scheme

(note: we define $\mathsf{PowersOfTwo}, \mathsf{Flatten}, \mathsf{BitDecomp}$ and $\mathsf{BitDecomp}^{-1}$ in Section 3.2 below)

$\mathsf{LRGSW.Setup}(1^\eta, 1^L)$**:** Recalling that η is security parameter of the scheme, and $L = poly(\eta)$ is the maximum circuit depth our scheme must evaluate, let $\tau = \max\{L, \eta^2\}$. Choose a lattice dimension $n = \tau^2$, modulus $\boxed{q \geq \tau \cdot 2^{2\tau \log^2 \tau}}$, and error distribution $\boxed{\chi = \overline{\Psi}_\beta, \text{ where } \beta = \tau \cdot \tau^{\log \tau} \text{ bounded}}$ Choose $\boxed{m = m(\eta, L) \geq 2n \log q + 3\eta.}$ Let $params = (n, q, \chi, m)$. Let $\ell = \lfloor \log q \rfloor + 1$ and $N = (n+1) \cdot \ell$.

$\mathsf{LRGSW.SecretKeyGen}(params)$**:** Choose $\mathbf{t} \xleftarrow{\$} \mathbb{Z}_q^n$. Let $sk = \mathbf{s} = (1, -\mathbf{t}_1, \ldots, -\mathbf{t}_n)$. Let $\mathbf{v} = \mathsf{PowersOfTwo}(\mathbf{s})$.

$\mathsf{LRGSW.PublicKeyGen}(\mathbf{s}, params)$**:** Let $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$. Let $\mathbf{e} \xleftarrow{\$} \chi^m$. Let $\mathbf{b} = \mathbf{At} + \mathbf{e}$. Let $pk = \mathbf{K} = [\mathbf{b}||\mathbf{A}]$.

$\mathsf{LRGSW.Encrypt}(\mathbf{K}, \mu)$**:** For message $\mu \in \{0,1\}$, choose $R \xleftarrow{\$} \{0,1\}^{N \times m}$. Let $\mathbf{I}_N$ be the $N \times N$ identity matrix.

$$\mathbf{C} = \mathsf{Flatten}(\mu \cdot \mathbf{I}_N + \mathsf{BitDecomp}(\mathbf{R} \cdot \mathbf{K})) \in \mathbb{Z}_q^{N \times N}$$

$\mathsf{LRGSW.Decrypt}(\mathbf{s}, \mathbf{C})$**:** Let i be the index among the first ℓ elements of $\mathbf{v}$ such that $\mathbf{v}_i = 2^i \in (\frac{q}{4}, \frac{q}{2}]$. Let $\mathbf{C}_i$ be the i^{th} row of $\mathbf{C}$. Compute $x_i = \langle \mathbf{C}_i, \mathbf{v} \rangle$. Output $\mu' = \left\lfloor \frac{x_i}{v_i} \right\rceil$

$\mathsf{LRGSW.NAND}(\mathbf{C}_1, \mathbf{C_2})$**:** Output $\mathsf{Flatten}(\mathbf{I}_N - \mathbf{C}_1 \cdot \mathbf{C}_2)$

3.2 Elementary Vector Operations in LRGSW

The above scheme description makes use of a number of vector operations that we describe below. Let $\mathbf{a}, \mathbf{b}$ be vectors of dimension k. Let $\ell = \lfloor \log q \rfloor + 1$. Note that the operations we describe are also defined over matrices, operating row by row on the matrix, and that all arithmetic is over $\mathbb{Z}_q$.

$\mathsf{BitDecomp}(\mathbf{a})$ = the $k \cdot \ell$ dimensional vector $(a_{1,0}, \ldots, a_{1,\ell-1}, \ldots, a_{k,0}, \ldots a_{k,\ell-1})$ where $a_{i,j}$ is the j^{th} bit in the binary representation of a_i, with bits ordered from least significant to most significant.

$\mathsf{BitDecomp}^{-1}(\mathbf{a}')$ For $\mathbf{a}' = (a_{1,0}, \ldots, a_{1,\ell-1}, \ldots, a_{k,0}, \ldots a_{k,\ell-1})$, let $\mathsf{BitDecomp}^{-1}(\mathbf{a}') = (\sum_{j=0}^{\ell-1} 2^j a_{1,j}, \ldots, \sum_{j=0}^{\ell-1} 2^j a_{k,j})$, but defined even when $\mathbf{a}'$ isn't binary.

$\mathsf{Flatten}(\mathbf{a}')$ = $\mathsf{BitDecomp}(\mathsf{BitDecomp}^{-1}(\mathbf{a}'))$

$\mathsf{PowersOfTwo}(\mathbf{b})$ = $(b_1, 2b_1, 4b_1, \ldots, 2^{\ell-1}b_1, \ldots, b_k, \ldots 2^{\ell-1}b_k)$.

3.3 Correctness

Correctness of the scheme follows because:

$$\mathbf{Cv} = \mu\mathbf{v} + \mathbf{RAs} = \mu\mathbf{v} + \mathbf{Re}$$

so, $x_i = \mu \cdot \mathbf{v}_i + \langle \mathbf{R}_i, \mathbf{e} \rangle$. Since $\mathbf{v}_i > \frac{q}{4}$, if we let $B = ||\mathbf{e}||_\infty$, since $\mathbf{R}_i$ is an N-dimensional binary vector, as long as $NB < \frac{q}{8}$, decryption will be correct.

Gentry et al. analyze the error growth of GSW and determine that if χ is β-bounded, and if $\mathbf{C}$ is the result of L levels of homomorphic evaluation, then with overwhelming probability, $B < \beta(N+1)^L$. To maintain correctness of their scheme, they set $B = \frac{q}{8}$, which gives us: $\frac{q}{\beta} > 8(N+1)^L$. This same analysis applies to LRGSW, and we set our ratio of q to β the same way.

4 Leakage Resilient Leveled FHE

Below we prove that LRGSW is leakage resilient, describe the efficiency tradeoffs we make to achieve leakage resilience, and briefly describe and why our leveled result but does not extend easily to full non-leveled homomorphism.

4.1 Adaptive Leakage Resilience of LRGSW

Theorem 4.1. *The leveled* LRGSW *scheme is resilient to adaptive bounded leakage of λ bits, where $\lambda \leq n - 2\log q - 4\eta$.*

Proof. We consider a probabilistic polynomial time adversary's advantage at playing the ALR^λ game (described in Definition 7). Recall that in this game, the adversary's view is $(\mathbf{K}, \mathbf{C}_b, h(\mathbf{K}, \mathbf{s}))$ where $\mathbf{C}_b$ is a correctly formed encryption of $b \in \{0, 1\}$.

Let $\mathbf{C}'_b = \mathsf{BitDecomp}^{-1}(\mathbf{C}_b) = \mathsf{BitDecomp}^{-1}(b \cdot \mathbf{I}_N) + \mathbf{R} \cdot \mathbf{K}$. Since $\mathsf{BitDecomp}^{-1}$ is a deterministic operation, it suffices to consider a probabilistic polynomial time adversary who plays the ALR^λ game with $\mathbf{C}'_b$.

In fact, an adversary's view after playing the ALR^λ game is $(\mathbf{K}, \mathsf{BitDecomp}^{-1}(b \cdot \mathbf{I}_N) + \mathbf{R} \cdot \mathbf{K}, h(\mathbf{K}, \mathbf{s}))$. Therefore, it is sufficient to show $(\mathbf{K}, \mathbf{RK}, h(\mathbf{K}, \mathbf{s})) \approx_c (\mathbf{K}, \mathbf{U} \xleftarrow{\$} \mathbb{Z}_q^{N \times n}, h(\mathbf{K}, \mathbf{s}))$.

Recall that $\mathbf{K} = [\mathbf{b}||\mathbf{A}]$ where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$, $\mathbf{t} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e} \xleftarrow{\$} \chi^m$, $\mathbf{b} = \mathbf{At} + \mathbf{e}$, and $\mathbf{s} = (1, -\mathbf{t}_1, \ldots, -\mathbf{t}_n)$. So define:

$$\mathcal{H}_{ALR} := (\mathbf{b}, \mathbf{A}, \mathbf{Rb}, \mathbf{RA}, h(\mathbf{A}, \mathbf{t}, \mathbf{e})), \mathcal{H}_{RAND} := (\mathbf{b}, \mathbf{A}, \mathbf{u'}, \mathbf{U}, h(\mathbf{A}, \mathbf{t}, \mathbf{e}))$$

Our goal is to show that $\mathcal{H}_{ALR} \approx_c \mathcal{H}_{RAND}$. We can think of the matrix $\mathbf{R}$ as a collection of N independent binary vectors $\mathbf{r_i} \xleftarrow{\$} \{0, 1\}^m$. So, $\mathcal{H}_{ALR} = (\mathbf{b}, \mathbf{A}, \{\mathbf{r_i} \cdot \mathbf{b}\}_{i \in [N]}, \{\mathbf{r_i A}\}_{i \in [N]}, h(\mathbf{A}, \mathbf{t}, \mathbf{e}))$

Now, define a series of hybrid games $\mathcal{H}_i$, for $0 \leq i \leq N$, where in game i, for $j < i$, $\mathbf{r_j} \cdot \mathbf{b}$ is replaced by $u'_j \xleftarrow{\$} \mathbb{Z}_q$, and $\mathbf{r_j A}$ is replaced by $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^n$, and for $j \geq i$, those terms are generated as they were in game $\mathcal{H}_{i-1}$.

It follows by inspection that $\mathcal{H}_0 = \mathcal{H}_{ALR}$ and $\mathcal{H}_N = \mathcal{H}_{RAND}$, so all that remains to show is that $\mathcal{H}_i \approx_c \mathcal{H}_{i+1}$.

We use Lemma 4.1, stated below, together with a simple reduction to prove this. Lemma 4.1 says that for a single $\mathbf{r} \xleftarrow{\$} \{0, 1\}^m$, $\mathcal{H}_{real} := (\mathbf{b}, \mathbf{A}, \mathbf{r} \cdot \mathbf{b}, \mathbf{rA}, h(\mathbf{A}, \mathbf{t}, \mathbf{e})) \approx_c \mathcal{H}_{rand} := (\mathbf{b}, \mathbf{A}, u', \mathbf{u}, h(\mathbf{A}, \mathbf{t}, \mathbf{e}))$.

So, given an input $\mathcal{H} = (\mathbf{b}, \mathbf{A}, \mathbf{b'}, \mathbf{a'}, h(\mathbf{A}, \mathbf{t}, \mathbf{e}))$ that is equal to either $\mathcal{H}_{real}$ or $\mathcal{H}_{rand}$, if, for $j \leq i$ choose $u'_j \xleftarrow{\$} \mathbb{Z}_q$, $\mathbf{u_j} \xleftarrow{\$} \mathbb{Z}_q^n$, and for $j > i + 1$, choose $r_j \xleftarrow{\$} \{0, 1\}^m$, we prepare the following distribution:

$$\left(\mathbf{b}, \mathbf{A}, \left\{u'_j\right\}_{j \leq i}, b', \{\mathbf{r_j} \cdot \mathbf{b}\}_{j > i+1}, \{\mathbf{u_j}\}_{j \leq i}, \mathbf{a'}, \{r_j \mathbf{A}\}_{j > i}, h(\mathbf{A}, \mathbf{t}, \mathbf{e})\right)$$

Then if $\mathcal{H} = \mathcal{H}_{real}$, this distribution is equal to $\mathcal{H}_i$, whereas if $\mathcal{H} = \mathcal{H}_{rand}$, the distribution is equal to $\mathcal{H}_{i+1}$. Since Lemma 4.1 tells us that $\mathcal{H}_{real} \approx_c \mathcal{H}_{rand}$, we conclude that no probabilistic polynomial time adversary can distinguish $\mathcal{H}_i$ and $\mathcal{H}_{i+1}$ with non-negligible advantage. □

We now state and prove Lemma 4.1.

Lemma 4.1. *Given $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$, $\mathbf{e} \leftarrow \chi^m$, $\mathbf{t} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{r} \xleftarrow{\$} \{0, 1\}^m$, $\mathbf{b} = \mathbf{At} + \mathbf{e}$, and $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^n$, and $u' \xleftarrow{\$} \mathbb{Z}_q$, and m, q, n defined as in the* LRGSW *scheme,*

$$\mathcal{H}_{real} := (\mathbf{b}, \mathbf{A}, \mathbf{r} \cdot \mathbf{b}, \mathbf{rA}, h(\mathbf{A}, \mathbf{t}, \mathbf{e})) \approx_c \mathcal{H}_{rand} := (\mathbf{b}, \mathbf{A}, u', \mathbf{u}, h(\mathbf{A}, \mathbf{t}, \mathbf{e}))$$

Proof. Our proof proceeds as follows:

- We define a series of intermediate hybrid games, $\mathcal{H}_a, \mathcal{H}_b, \mathcal{H}_c$, and show: $\mathcal{H}_{real} \approx_s \mathcal{H}_a \approx_c \mathcal{H}_b \approx_s \mathcal{H}_c \approx_c \mathcal{H}_{rand}$. Our hybrids are:
  - $\mathcal{H}_a := (\mathbf{At} + \mathbf{e}, \mathbf{A}, \mathbf{ut} + \mathbf{r} \cdot \mathbf{e}, \mathbf{u}, h(\mathbf{A}, \mathbf{t}, \mathbf{e}))$, where $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^N$.
  - $\mathcal{H}_b := (\tilde{\mathbf{A}}\mathbf{t} + \mathbf{e}, \tilde{\mathbf{A}}, \mathbf{ut} + \mathbf{r} \cdot \mathbf{e}, \mathbf{u}, h(\tilde{\mathbf{A}}, \mathbf{t}, \mathbf{e}))$, where $\tilde{\mathbf{A}} \leftarrow \mathsf{Lossy}$, as defined by Lemma 4.2.
  - $\mathcal{H}_c := (\tilde{\mathbf{A}}\mathbf{t} + \mathbf{e}, \tilde{\mathbf{A}}, u', \mathbf{u}, h(\tilde{\mathbf{A}}, \mathbf{t}, \mathbf{e}))$, where $u' \xleftarrow{\$} \mathbb{Z}_q$.
- Lemma 4.2, stated below, immediately gives us $\mathcal{H}_a \approx_c \mathcal{H}_b$, and $\mathcal{H}_c \approx_c \mathcal{H}_{rand}$, because it tells us that $\tilde{\mathbf{A}} \approx_c \mathbf{A}$. Thus, no further work is needed for these two steps.
- We use Claim 1 to show that $\mathcal{H}_{real} \approx_s \mathcal{H}_a$.
- Finally, we use Claim 2 to prove $\mathcal{H}_b \approx_s \mathcal{H}_c$.

Claim 1. $\mathcal{H}_{real} \approx_s \mathcal{H}_a$

Proof. The only difference between games $\mathcal{H}_{real}$ and $\mathcal{H}_a$ is that $\mathbf{rA}$ is replaced by $\mathbf{u}$ where $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^N$. Note that if we can show:

$$(\mathbf{At} + \mathbf{e}, \mathbf{A}, \mathbf{rAt}, \mathbf{r} \cdot \mathbf{e}, \mathbf{rA}, h(\mathbf{A}, \mathbf{t}, \mathbf{e})) \approx_s (\mathbf{At} + \mathbf{e}, \mathbf{A}, \mathbf{u} \cdot \mathbf{t}, \mathbf{r} \cdot \mathbf{e}, \mathbf{u}, h(\mathbf{A}, \mathbf{t}, \mathbf{e}))$$

this implies our claim.

To prove the above, we use the generalized form of the leftover hash lemma (Lemma A.2 in Appendix A of this paper), which tells us that for any random variable x, if $\tilde{H}_\infty(\mathbf{r}|x)$ is high enough, then $(\mathbf{A}, \mathbf{rA}, x) \approx_s (\mathbf{A}, \mathbf{u}, x)$, which in turn implies that for any $\mathbf{t}$, $(\mathbf{A}, \mathbf{rA}, \mathbf{rAt}, x) \approx_s (\mathbf{A}, \mathbf{u}, \mathbf{u} \cdot \mathbf{t}, x)$. So, set $x = (\mathbf{At} + \mathbf{e}, \mathbf{r} \cdot \mathbf{e}, h(\mathbf{A}, \mathbf{t}, \mathbf{e}))$. Since $\mathbf{r}$ is an m-dimensional binary vector chosen uniformly at random and $\mathbf{r} \cdot \mathbf{e}$ is $\ell = \lfloor \log q \rfloor + 1$ bits long, and $\mathbf{r}$ is independent of $\mathbf{e}$, we have:

$$\begin{aligned} &\tilde{H}_\infty(\mathbf{r}|\mathbf{At} + \mathbf{e}, \mathbf{r} \cdot \mathbf{e}, h(\mathbf{A}, \mathbf{t}, \mathbf{e})) \\ \geq &\tilde{H}_\infty(\mathbf{r}|\mathbf{r} \cdot \mathbf{e}, \mathbf{e}) \geq \tilde{H}_\infty(\mathbf{r}|\mathbf{e}) - \ell = m - \ell \end{aligned}$$

For Lemma A.2 to hold, we need $n \leq \frac{m - \ell - 2\eta - O(1)}{\log q}$. Choosing $m \geq 2n \log q + 3\eta$ suffices. □

Claim 2. $\mathcal{H}_b \approx_s \mathcal{H}_c$

Proof. The difference between $\mathcal{H}_b$ and $\mathcal{H}_c$ is that $\mathbf{u} \cdot \mathbf{t} + \mathbf{r} \cdot \mathbf{e}$ is replaced by $u' \xleftarrow{\$} \mathbb{Z}_q$. We employ a similar strategy to that from claim Claim 1, using the leftover hash lemma to show

$$(\tilde{\mathbf{A}}\mathbf{t} + \mathbf{e}, \tilde{\mathbf{A}}, \mathbf{ut}, \mathbf{r} \cdot \mathbf{e}, \mathbf{u}, h(\tilde{\mathbf{A}}, \mathbf{t}, \mathbf{e})) \approx_s (\tilde{\mathbf{A}}\mathbf{t} + \mathbf{e}, \tilde{\mathbf{A}}, v, \mathbf{r} \cdot \mathbf{e}, \mathbf{u}, h(\tilde{\mathbf{A}}, \mathbf{t}, \mathbf{e}))$$

where $v \xleftarrow{\$} \mathbb{Z}_q$. Note that this distribution contains both $\mathbf{ut}$ and $\mathbf{r} \cdot \mathbf{e}$, whereas the adversary only sees $\mathbf{ut} + \mathbf{r} \cdot \mathbf{e}$. Proving that $\mathbf{ut}$ can be replaced by v implies that in the adversary's actual view, $\mathbf{ut} + \mathbf{re}$ can be replaced by $u' \xleftarrow{\$} \mathbb{Z}_q$.

Now, we bound the ϵ-smooth min-entropy of $\mathbf{t}$. There exists $\epsilon = negl(\eta)$ such that

$$\begin{aligned} &\tilde{H}_\infty^\epsilon(\mathbf{t}|\tilde{\mathbf{A}}\mathbf{t} + \mathbf{e}, \tilde{\mathbf{A}}, \mathbf{r} \cdot \mathbf{e}, h(\tilde{\mathbf{A}}, \mathbf{t}, \mathbf{e}))) \\ \geq &\tilde{H}_\infty^\epsilon(\mathbf{t}|\tilde{\mathbf{A}}\mathbf{t} + \mathbf{e}, \tilde{\mathbf{A}}) - BitLength(\mathbf{r} \cdot \mathbf{e}) - BitLength(h(\tilde{\mathbf{A}}, \mathbf{t}, \mathbf{e})) \\ \geq &\tilde{H}_\infty^\epsilon(\mathbf{t}|\tilde{\mathbf{A}}\mathbf{t} + \mathbf{e}, \tilde{\mathbf{A}}) - \ell - \lambda \end{aligned}$$

and Lemma 4.2 (stated and proven below), tells us that $\tilde{H}^{\epsilon}_{\infty}(\mathbf{t}|\tilde{\mathbf{A}}\mathbf{t}+\mathbf{e}, \tilde{\mathbf{A}}) \geq n$.

Applying the ϵ-smooth variant of the leftover hash lemma (Corollary A.2.1), we see that we need $n - \ell - \lambda$ to be high enough that $\log q \leq (n - \ell - \lambda) - 2\eta - O(1)$. So, if we set h to leak at most $\lambda \leq n - 2\log q - 4\eta$ bits, the claim follows. □

Since $\mathcal{H}_{real} \approx_s \mathcal{H}_a \approx_c \mathcal{H}_b \approx_s \mathcal{H}_c \approx_c \mathcal{H}_{rand}$, we know that $\mathcal{H}_{real} \approx_c \mathcal{H}_{rand}$.

□

We now state and prove Lemma 4.2, used both to prove Claim 2, and to show $\mathcal{H}_a \approx_c \mathcal{H}_b$, and $\mathcal{H}_c \approx_c \mathcal{H}_{rand}$.

Lemma 4.2. *There exists a distribution* Lossy *such that* $\tilde{\mathbf{A}} \leftarrow \mathsf{Lossy} \approx_c \mathbf{U} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$ *and given* $\mathbf{t} \xleftarrow{\$} \mathbb{Z}_q^n$, *and* $\mathbf{e} \leftarrow \chi$, $\tilde{H}^{\epsilon}_{\infty}(\mathbf{t}|\tilde{\mathbf{A}}, \tilde{\mathbf{A}}\mathbf{t}+\mathbf{e}) \geq n$, *where* $\epsilon = negl(\eta)$.

Proof. Define Lossy as follows:

- Choose $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{m \times n'}$, $\mathbf{D} \xleftarrow{\$} \mathbb{Z}_q^{n' \times n}$, and $\mathbf{Z} \leftarrow \overline{\Psi}_{\alpha}^{m \times n}$, where $\frac{\alpha}{\beta} = negl(\eta)$ and $n' \log q \leq n - 2\eta + 2$.
- Let $\tilde{\mathbf{A}} = \mathbf{CD} + \mathbf{Z}$
- output $\tilde{\mathbf{A}}$.

We note that this distribution was first used in [18] and we refer the reader to that paper for more details.

1. $\tilde{\mathbf{A}} \approx_c U \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$:
   $\tilde{\mathbf{A}}$ is a *DLWE* instance, with $\mathbf{D}$ as the secret and $\mathbf{Z}$ as the error term, so as long as $DLWE_{n',q,\alpha}$ is hard, then $\tilde{A} \approx_c \mathbb{Z}_q^{m \times n}$.
2. $\tilde{H}^{\epsilon}_{\infty}(\mathbf{t}|\tilde{\mathbf{A}}\mathbf{t}+\mathbf{e}) = n$, where $\epsilon = negl(\eta)$:

   - First, note that $\mathbf{t} \xleftarrow{\$} \mathbb{Z}_q^m$ is identically distributed to $\mathbf{t} = \mathbf{t_0} + \mathbf{t_1}$ where $\mathbf{t_0} \xleftarrow{\$} \{0,1\}^m$, and $\mathbf{t_1} \xleftarrow{\$} \mathbb{Z}_q^m$, so consider $\mathbf{t} = \mathbf{t_0} + \mathbf{t_1}$.
   - Clearly for any ϵ, $\tilde{H}^{\epsilon}_{\infty}(\mathbf{t}|\tilde{\mathbf{A}}\mathbf{t}+\mathbf{e}) \geq \tilde{H}^{\epsilon}_{\infty}(\mathbf{t_0}|\tilde{\mathbf{A}}\mathbf{t}+\mathbf{e})$, so any lower bound on the min-entropy of $\mathbf{t_0}$ will apply to $\mathbf{t}$ as well. We therefore only consider the min-entropy of $\mathbf{t_0}$.
   - Rewriting the above, we know that $\tilde{\mathbf{A}}\mathbf{t} + \mathbf{e} = \mathbf{CDt} + \mathbf{Zt} + \mathbf{e}$

     $$= \mathbf{CDt_0} + \mathbf{Zt_0} + \mathbf{CDt_1} + \mathbf{Zt_1} + \mathbf{e}$$

   - Since $\mathbf{e}$ is drawn from a discretized Gaussian distribution, and since each element of $\mathbf{Zt_0}$ is negligibly small compared to the corresponding element of $\mathbf{e}$, we know that $\mathbf{e} + \mathbf{Zt_0} \approx_s \mathbf{e}$. Thus there exists some $\epsilon_1 = negl(\eta)$ such that

     $$\tilde{H}^{\epsilon_1}_{\infty}(\mathbf{t_0}|\mathbf{CDt_0} + \mathbf{CDt_1} + \mathbf{Zt_1} + \mathbf{e}) \geq \tilde{H}_{\infty}(\mathbf{t_0}|\mathbf{CDt_0} + \mathbf{CDt_1} + \mathbf{Zt_1} + \mathbf{Zt_0} + \mathbf{e})$$

- Since $\tilde{H}_\infty(\mathbf{t_0}|\mathbf{CDt_1}+\mathbf{Zt_1}+\mathbf{e}) \geq n$, by a variant of the leftover hash lemma (Lemma A.2), for our choice of n', we know that $(\mathbf{CDt_0}+\mathbf{CDt_1}+\mathbf{Zt_1}+\mathbf{e}) \approx_s (\mathbf{Cu_0}+\mathbf{CDt_1}+\mathbf{Zt_1}+\mathbf{e})$, where $\mathbf{u_0} \xleftarrow{\$} \mathbb{Z}_q^{n'}$. Since the statistical distance between these two distributions is some $\epsilon_2 = negl(\eta)$, we can conclude that there exists some $\epsilon = \epsilon_1 + \epsilon_2 = negl(\eta)$ such that

$$\tilde{H}^\epsilon_\infty(\mathbf{t_0}|\mathbf{Cu_0} + \mathbf{CDt_1} + \mathbf{Zt_1} + \mathbf{e}) \geq \tilde{H}_\infty(\mathbf{t_0}|\mathbf{CDt_0} + \mathbf{CDt_1} + \mathbf{Zt_1} + \mathbf{Zt_0} + \mathbf{e})$$

- Since each of $\mathbf{C}, \mathbf{u_0}, \mathbf{D}, \mathbf{Z}, \mathbf{t_1}, \mathbf{e}$, is independent of $\mathbf{t_0}$, this quantity equals $H_\infty(\mathbf{t_0}) = n$. Therefore there exists $\epsilon = negl(\eta)$ such that $\tilde{H}^\epsilon_\infty(\mathbf{t}|\tilde{\mathbf{A}}\mathbf{t} + \mathbf{e}) \geq n$ as well.

□

4.2 The Cost of Leakage Resilience: GSW v. LRGSW

In order to make the GSW scheme leakage resilient, we needed to make a number of tradeoffs. First, there's a penalty to efficiency, as a number of the scheme's parameters need to be set higher than they are in GSW in order to maintain equivalent security in the presence of leakage. Second, our proof relies crucially on the fact that the LRGSW scheme does not have an evaluation key. The leveled version of the GSW scheme does not have an evaluation key, but the version that allows for full (non-leveled) FHE does have one. For this reason, LRGSW cannot be easily extended to a non-leveled scheme.

4.2.1 Parameter Setting

The hardness constraints and the correctness constraints of our scheme are in conflict. The hardness constraints tell us that the ratio of the dimension to the error bound affects the relative hardness of the *DLWE* problems, with a higher β leading to more security. However, the correctness constraint shows us that $\frac{q}{\beta}$ must grow exponentially with the depth of the circuit, which shows both that β should be set low, and since there is a limit to how low β can be set, q must grow exponentially with depth. However, the hardness constraints also tell us that if the depth is $O(n)$ or bigger, since L, the circuit depth, is in the exponent of q, the underlying *GapSVP* problems become easy. To protect against this, we must ensure that n is polynomial in L. We describe these constraints in more detail and show how to set the parameters to meet all of them in Appendix B.

Also in the appendix, we present Lemma B.1, which can replace Lemma 4.2 in our proofs above. This new lemma uses techniques from Alwen, Krenn, Pietrzak, and Wichs [3] which, as summarized in Corollary B.1.1, allow us to reduce the size of q and β (in particular, β is no longer super-polynomial in η), at a cost of a lower value for λ.

In Table 1 we provide sample parameter settings that simultaneously meet all correctness and security constraints. We compare these settings to those of GSW. In the table, $\tau_1 = \max\{L, \eta^2\}$, and $\tau_2 = \max\{L, \eta^3\}$.

4.2.2 Evaluation Keys and the Problem with Bootstrapping

Our current techniques are sufficient for proving leakage resilience of a leveled fully homomorphic encryption scheme, but do not extend to a non-leveled scheme. The bootstrapping paradigm, first defined by Gentry in [16], is to take a scheme that is capable of homomorphically evaluating its own decryption circuit and transform it into one that can evaluate functions f of arbitrary depth

Table 1: Sample settings of GSW v. LRGSW

Parameter	GSW	LRGSW with Lemma 4.2	LRGSW with Lemma B.1
n	$O(\eta)$	τ_1^4	τ_2^3
q	$2^{L \log n}$	$2^{\tau_1 \log^2 \tau_1}$	$2^{\tau_2 \log n}$
χ	$O(n)$ -bounded	$\overline{\Psi}_\beta$, $\beta = 2^{\log^2 \tau_1}$	$\beta = 3n^3\tau_2^3$
m	$2n \log q$	$2n \log q + 3\eta$	$2n \log q + 3\eta$
λ	0	$n - 2 \log q - 4\eta$	$n - (2+\eta) \log q - \eta \log m - 4\eta$

by performing the homomorphic-decrypt operation after each gate in f. All existing fully homomorphic schemes, including the GSW scheme, achieve full, as opposed to leveled fully homomorphic encryption through bootstrapping.

The bootstrapping paradigm tells us that given a somewhat homomorphic scheme, publishing an encryption of the scheme's secret key, together with any other data necessary to allow the scheme to homomorphically evaluate its own decryption procedure, makes the scheme fully homomorphic [16]. Thus, the scheme must be secure when an adversary sees $(pk, \mathsf{Enc}_{pk}(sk))$. However, a scheme that is secure when the adversary sees $(pk, \mathsf{Enc}_{pk}(sk))$ or when the adversary sees $(pk, h(pk, sk))$, as is the case in the leakage resilience definition, is not necessarily secure when it sees $(pk, \mathsf{Enc}_{pk}(sk), h(pk, sk, \mathsf{Enc}_{pk}(sk)))$ all together.

Below we provide formal definitions of bootstrapping:

Definition 11. Let HE be $\mathcal{L}-$*homomorphic* and let f_{nand} be the augmented decryption function defined below:

$$f_{nand} = \mathsf{HE.Dec}(sk, c_1) \text{ NAND } \mathsf{HE.Dec}(sk, c_2)$$

Then HE is **bootstrappable** if $f_{nand} \in \mathcal{L}$

Definition 12. A public key encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ has **weak circular security** if it is secure even against an adversary with auxiliary information containing encryptions of all secret key bits.

If we tried to make the LRGSW scheme bootstrappable, we would need not only circular security (which current FHE schemes assume rather than prove), but circular security in the presence of leakage.

If we were to create an *evk* that contained an encryption of the secret key under that same secret key, we would have something of the form $\mathbf{A}, \mathbf{At} + \mathbf{e} + \mathsf{BitDecompose}(\mathbf{t})$. One might try to follow the same technique outlined in the proof of Lemma 4.2, and show that the average min-entropy of $\mathbf{t}$, conditioned on seeing $\mathbf{A}, \mathbf{At}+\mathbf{e}+\mathsf{BitDecompose}(\mathbf{t})$, is still high. Unfortunately, for this technique to work, $\mathbf{t}$ needs to be only in the secret term, not in the error term as well.

To get around this, we might consider trying to "chain" our *DLWE* secrets, so that we have two *DLWE* secrets: $\mathbf{t}$ and $\mathbf{t'}$, but only consider our secret key to be $\mathbf{t'}$. In this case, our encryption key would be $(\mathbf{A}, \mathbf{At} + \mathbf{e})$, and our evaluation key would be $(\mathbf{A'}, \mathbf{A't'} + \mathbf{e'} + \mathsf{BitDecomp}(\mathbf{t}))$. In this case, we would still need to show that $\tilde{H}_\infty(\mathbf{t}|\mathbf{A't'} + \mathbf{e'} + \mathsf{BitDecomp}(\mathbf{t}))$ was sufficiently high, and since $\mathbf{t}$ is in the error term instead of the secret term, our current techniques will not suffice.

Notice, as well, that these limitations apply to any LWE-based FHE scheme with an evaluation key. Since all other existing LWE based FHE schemes use an evaluation key, our result for the GSW scheme cannot be easily extended to these schemes either.

5 Going Beyond Leveled Homomorphism

In this section we present several new ideas for achieving full (as opposed to leveled) FHE that is also leakage resilient.

5.1 Our First Approach

We observe that by definition, a leakage function h is a function of the scheme's public and secret keys. This means an adversary can see $h(pk, sk, \mathsf{Enc}_{pk}(sk))$ only if $\mathsf{Enc}_{pk}(sk)$ is part of the scheme's public key. If instead, we can somehow generate $\mathsf{Enc}_{pk}(sk)$ on-the-fly as it is needed, the adversary sees only $h(pk, sk)$, instead.

More precisely, let $\mathsf{E} = (\mathsf{KeyGen}(), \mathsf{Enc}(), \mathsf{Dec}())$ be any encryption scheme (*not* necessarily homomorphic) that is also resilient to adaptive bounded leakage of λ bits, and let $\mathsf{HE} = (\mathsf{KeyGen}(), \mathsf{Enc}(), \mathsf{Dec}(), \mathsf{Eval}())$ be any (leveled) fully homomorphic encryption scheme. Then we consider the following hybrid scheme:

$\mathsf{Scheme1.KeyGen}(1^\eta)$**:** Run $(pk, sk) \leftarrow \mathsf{E.KeyGen}(1^\eta)$. Set the public and secret keys to be pk, sk.

$\mathsf{Scheme1.Enc}_{pk}(m)$**:** To encrypt a message m, first run $(pk', sk') \leftarrow \mathsf{HE.KeyGen}(1^\eta)$. Then output $(pk', \mathsf{HE.Enc}_{pk'}(m), \mathsf{E.Enc}_{pk}(sk'))$ as the ciphertext.

$\mathsf{Scheme1.Dec}_{sk}(c)$**:** To decrypt a ciphertext c, first parse $c = (pk', c_1, c_2)$, and obtains $sk' = \mathsf{E.Dec}_{sk}(c_2)$. Then output $\mathsf{HE.Dec}_{sk'}(c_1)$.

$\mathsf{Scheme1.Eval}_{pk}(f, c)$**:** To evaluate a function f over a ciphertext c, first parse $c = (pk', c_1, c_2)$ and then output $(pk', \mathsf{HE.Eval}_{pk'}(f, c_1), c_2)$.

It is not hard to obtain the following theorem:

Theorem 5.1. *If* E *is an encryption scheme that is resilient to adaptive bounded leakage of* λ *bits and* HE *is a (leveled) fully homomorphic encryption scheme, then* Scheme1 *is a (leveled) fully homomorphic scheme that has the following properties:*

1. *It is resilient to adaptive bounded leakage of* λ *bits.*
2. *It allows unary homomorphic evaluation over any single ciphertext.*
3. *If* HE *is fully homomorphic, then* Scheme1 *has succinct ciphertexts (whose lengths do not depend on the size of circuits supported by the evaluation), while if* HE *is* L*-leveled homomorphic, then the size of the ciphertexts in* Scheme1 *depends on* L*.*

A word is in order about property 2 above. If HE is a bit-encryption scheme, then we can think of the message space as bit-strings, so a message $\mathbf{m} \in \{0,1\}^t$, and define encryption to be bit-by bit.

In this case, "unary" refers to functions over the bits of $\mathbf{m}$. Another way to think of this is that $\mathsf{Scheme1}$ is (leveled) fully homomorphic for any group of bits batch-encrypted at the same time.

The proof of this theorem is simple and quite similar to that of Theorem 5.2, so we omit the proof here, and refer the reader to our proof of that theorem below.

5.2 Our Second Approach

Our next step is to extend our result so that we can homomorphically combine ciphertexts regardless of when they were created. The reason we cannot do so above is because two ciphertexts formed at different times will be encrypted under different public keys of the underlying HE scheme. To solve this issue, we consider instantiating HE with a multi-key FHE scheme, as recently defined and constructed by López-Alt, Tromer and Vaikuntanathan (LTV) [24].

A scheme $\mathsf{HE}^{(\mathsf{N})}$ is a N-Key Multikey (leveled) FHE scheme if it is a (leveled) FHE scheme with the following two additional algorithms:

- $\mathsf{mEval}(f, pk_1, \ldots, pk_t, c_1, \ldots, c_t)$ that takes as input an t-ary function f, t evaluation keys and ciphertexts, and output a combined ciphertext c^*.
- $\mathsf{mDec}(sk_1, \ldots, sk_t, c^*)$ that takes c^*, generated by mEval and t secret keys such that sk_i corresponds to pk_i for $i \in [t]$, and outputs $f(m_1, m_2, \ldots m_t)$.

where the above holds for any $t \leq T$, with $c_1, \ldots c_t$ any ciphertexts under $pk_1, \ldots pk_t$, i.e. $c_i = \mathsf{Enc}_{pk_i}(m_i)$ for all $i \in [t]$.

If we replace HE with $\mathsf{HE}^{(\mathsf{N})}$, we get the following evaluation function:

> $\mathsf{Scheme2.Eval}_{pk}(f, c_1, \ldots, c_t)$: To evaluate a function f over ciphertexts $c_1, \ldots c_t$, first parse $c_i = (pk'_i, c_{i,1}, c_{i,2})$ for $i \in [t]$. Then, calculate $c^*_1 = \mathsf{HE}^{(\mathsf{N})}.\mathsf{Eval}(pk'_1, \ldots, pk'_t, c_{1,1}, \ldots, c_{t,1})$. Finally, output $(pk'_1, \ldots pk'_t, c^*_1, c_{1,2}, \ldots, c_{t,2})$.

The problem with this approach is that the resulting ciphertext needs to include all the public keys and secret keys from $\mathsf{HE}^{(\mathsf{N})}$ in order to run multikey decryption ($\mathsf{HE}^{(\mathsf{N})}.\mathsf{mDec}$). This means that outputs of the Eval function will have a different format than freshly generated ciphertexts, and no longer be compact. Thus $\mathsf{Scheme2}$ cannot possibly meet the definition of fully homomorphic.

5.3 The Final Scheme

We now observe that the LTV construction actually achieves multi-key FHE with a more fine-grained definition than we provided above: one where not only *ciphertexts*, but also *keys* can be combined. As described in Section 3.4 of their paper, given $c_1 = \mathsf{LTV.Enc}(pk_1, m_1)$, $c_2 = \mathsf{LTV.Enc}(pk_2, m_2)$, one step of $\mathsf{LTV.Eval}$ is to calculate $pk^* = pk_1 \cup pk_2$. We can separate out this step and generalize it, defining $\mathsf{CombinePK}(pk_1, pk_2, \ldots, pk_t) = \bigcup_{i=1}^{t} pk_i$. Similarly, in their scheme, the secret keys are polynomials, and they show how to create a "joint secret key" by multiplying the polynomials together. We give this procedure a name, defining $\mathsf{CombineSK}(sk_1, sk_2, \ldots sk_t) = \prod_{i=1}^{t} sk_k$.

Definition 13. A scheme $\mathsf{HE}^{(\mathsf{N})}$ is an **N-Key Multikey (leveled) FHE scheme** if it is a (leveled) FHE scheme with the following additional algorithms: For any $t \leq N$, let $c_1, \ldots c_t$ be any ciphertexts under $pk_1, \ldots pk_t$, i.e. $c_i = \mathsf{Enc}_{pk_i}(m_i)$ for all $i \in [t]$.

- $pk^* = \mathsf{CombinePK}(pk_1, pk_2, \ldots, pk_t)$.
- A multi-key encryption algorithm $\mathsf{mEval}(f, pk_1, \ldots, pk_t, c_1, c_2, \ldots, c_t)$ that first calls $pk^* = \mathsf{CombinePK}(pk_1, pk_2, \ldots, pk_t)$, and then produces c^*, and outputs c^* and pk^*. Note that this c^* and pk^* can be used as input for successive calls to mEval.
- $sk^* = \mathsf{CombineSK}(sk_1, sk_2, \ldots, sk_t)$.
- A multikey decryption algorithm $\mathsf{mDec}(sk_1, \ldots, sk_t, c^*)$ that calls $\mathsf{CombineSK}$ and then runs $\mathsf{Dec}(sk^*, c^*)$ to produce $f(m_1, m_2, \ldots m_t)$.

As long as the outputs of $\mathsf{CombineSK}$ and $\mathsf{CombinePK}$ are succinct, we can update our scheme to make ciphertexts succinct.

Let $\mathsf{SHE} = (\mathsf{KeyGen}(), \mathsf{Enc}(), \mathsf{Dec}(), \mathsf{Eval}())$ be any somewhat[2] homomorphic encryption scheme that is also resilient to adaptive bounded leakage of λ bits, and let $\mathsf{HE}^{(\mathsf{N})} = (\mathsf{KeyGen}(), \mathsf{Enc}(), \mathsf{Dec}(), \mathsf{mEval}(), \mathsf{CombinePK}(), \mathsf{CombineSK}())$ be any N-key multikey fully homomorphic encryption scheme. Then we consider the following combined scheme:

$\mathsf{Scheme3.KeyGen}(1^\eta)$**:** Run $(pk, sk) \leftarrow \mathsf{SHE.KeyGen}(1^\eta)$. Set the public and secret keys to be pk, sk.

$\mathsf{Scheme3.Enc}(pk, m)$**:** First, run $(pk', sk') \leftarrow \mathsf{HE.KeyGen}(1^\eta)$.
Then output $(pk', \mathsf{HE.Enc}(pk', m), \mathsf{SHE.Enc}(pk, sk'))$ as the ciphertext.

$\mathsf{Scheme3.Eval}(pk, f, c_1, \ldots, c_t)$**:** First parse $c_i = (pk'_i, c_{i,1}, c_{i,2})$ for $i \in [t]$.
Then, calculate $c_1^* = \mathsf{HE}^{(\mathsf{N})}.\mathsf{Eval}(pk'_1, \ldots, pk'_t, f, c_{1,1}, \ldots, c_{t,1})$,
$pk'^* = \mathsf{HE}^{(\mathsf{N})}.\mathsf{CombinePK}(pk'_1, \ldots, pk'_t)$,
$c_2^* = \mathsf{SHE.Eval}(pk, \mathsf{HE.CombineSK}, c_{1,2}, \ldots, c_{t,2})$.
Finally, output (pk'^*, c_1^*, c_2^*).

$\mathsf{Scheme3.Dec}(sk, c)$**:** To decrypt a ciphertext c, first parse $c = (pk', c_1, c_2)$, and obtain $sk' = \mathsf{SHE.Dec}(sk, c_2)$. Then output $\mathsf{HE.Dec}(sk', c_1)$.

This lets us achieve the following theorem.

Theorem 5.2. *Let* SHE *be a* $\mathcal{C}$*-homomorphic encryption scheme for some circuilt class* $\mathcal{C}$ *such that* $\mathsf{HE}^{(\mathsf{N})}.\mathsf{CombineSK} \in \mathcal{C}$*. Let* $\mathsf{HE}^{(\mathsf{N})}$ *be an* N*-Key multikey FHE scheme. If* SHE *is resilient to adaptive, bounded leakage of* λ *bits, then* $\mathsf{Scheme3}$ *has the following properties:*

1. *It allows homomorphic evaluation of (up to)* N*-ary circuits of arbitrary (poly*(η)*) depth.*
2. *If* SHE *is a leveled homomorphic encryption scheme, then the ciphertext size depends on* N*. If* SHE *is fully homomorphic, then* $\mathsf{Scheme3}$ *has succinct ciphertexts (whose lengths do not depend* N*).*
3. *It is resilient to adaptive bounded leakage of* λ *bits.*

Proof. We address each statement in turn.

1. This follows immediately from the fact that by definition, $\mathsf{HE}^{(\mathsf{N})}$ allows homomorphic evaluation of (up to) N-ary circuits of arbitrary (*poly*(η)) depth.

[2] SHE must support circuits large enough to evaluate $\mathsf{CombineSK}$, but does not need to be fully homomorphic.

2. If SHE is leveled, its key-size is dependent on L, the number of levels of homomorphic evaluation it can support. To instantiate Scheme3, we need SHE to homomorphically evaluate CombineSK, an N-ary circuit whose depth is a function of its arity. Thus, the key size of SHE, and by extension, of Scheme3 is a function of N. In contrast, if SHE is not leveled, its key size is independent of L, and thus of N as well.

3. A simple reduction shows that if SHE is leakage resilient, then Scheme3 will be as well. Given a probabilistic polynomial time adversary $\mathcal{A}$ who wins the ALR game with Scheme3 with non-negligible advantage, it is easy to construct a ppt $\mathcal{B}$ who wins the ALR game with SHE with the same advantage. Upon receiving the public key from SHE, $\mathcal{B}$ simply forwards this information to $\mathcal{A}$. Whenever $\mathcal{A}$ requests an encryption of a message, $\mathcal{B}$ simply runs HE.KeyGen, and then follows Scheme3.Enc(), and forwards the result to $\mathcal{A}$. When $\mathcal{A}$ decides upon a leakage function, $\mathcal{B}$ uses that same leakage function. $\mathcal{A}$'s view when interacting with $\mathcal{B}$ is exactly its view when interacting with Scheme3 so its advantage is the same. Therefore, $\mathcal{B}$ would have the same advantage when interacting with Scheme3.

□

5.4 Instantiation

We instantiate Scheme3 using LRGSW for SHE and LTV for $\mathsf{HE}^{(\mathsf{N})}$. The LTV construction can be summarized by the following theorem:

Theorem 5.3. *(from theorem 4.5 in [24]) For every $N = poly(\eta)$, under the DSPR[3] and RLWE[4] assumptions with proper parameters, there exists an N-key multi-key (leveled) Fully Homomorphic Encryption Scheme. Under the additional assumption of weak circular security, we can remove the "leveled" constraint.*

The above theorem lets us instantiate Scheme3 with LTV and LRGSW, and together with with theorem 4.1 gives us the following corollary:

Corollary 5.0.1. *For every $T = poly(\eta)$ there exists an FHE scheme that supports homomorphic evaluation of all t-nary circuits for $t \leq T$, and depth $poly(\eta)$, under appropriate DSPR, RLWE, and DLWE assumptions. Under appropriate choices of n and q chosen so that certain DLWE assumptions hold, the scheme is resilient to adaptive bounded leakage of λ bits, where $\lambda \leq n - 2\log q - 4\eta$.*

6 Acknowledgements

This work was done partially under the support of NSF grants 1012060 and 0964541 We would like to thank Anna Lysyanskaya for many useful discussions and our TCC reviewers for their helpful comments.

[3]The *DSPR* assumption is the "Decisional Small Polynomial Ratio" introduced in [24].

[4]*RLWE* stands for "Ring Learning With Errors," first introduced in [25].

References

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

[2] Joël Alwen, Yevgeniy Dodis, Moni Naor, Gil Segev, Shabsi Walfish, and Daniel Wichs. Public-key encryption in the bounded-retrieval model. In *EUROCRYPT*, pages 113–134, 2010.

[3] Joël Alwen, Stephan Krenn, Krzysztof Pietrzak, and Daniel Wichs. Learning with rounding, revisited - new reduction, properties and applications. In *CRYPTO (1)*, pages 57–74, 2013.

[4] Victor Boyko. On the security properties of oaep as an all-or-nothing transform. In *CRYPTO*, pages 503–518, 1999.

[5] Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical gapsvp. *IACR Cryptology ePrint Archive*, 2012:78, 2012.

[6] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *ICTS*, pages 309–325, 2012.

[7] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. In *FOCS*, pages 97–106, 2011.

[8] Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-lwe and security for key dependent messages. In *CRYPTO*, pages 505–524, 2011.

[9] 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.

[10] Jean-Sébastien Coron, Avradip Mandal, David Naccache, and Mehdi Tibouchi. Fully homomorphic encryption over the integers with shorter public keys. In *CRYPTO*, pages 487–504, 2011.

[11] Yevgeniy Dodis, Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. In *TCC*, pages 361–381, 2010.

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

[13] Yevgeniy Dodis, Shien Jin Ong, Manoj Prabhakaran, and Amit Sahai. On the (im)possibility of cryptography with imperfect randomness. In *FOCS*, pages 196–205, 2004.

[14] 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, March 2008.

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

[16] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, STOC '09, pages 169–178, New York, NY, USA, 2009. ACM.

[17] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. *IACR Cryptology ePrint Archive*, 2013:340, 2013.

[18] Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Robustness of the learning with errors assumption. In *ICS*, pages 230–240, 2010.

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

[20] Shai Halevi and Huijia Lin. After-the-fact leakage in public-key encryption. In *TCC*, pages 107–124, 2011.

[21] Russell Impagliazzo, Leonid A. Levin, and Michael Luby. Pseudo-random generation from one-way functions (extended abstracts). In *STOC*, pages 12–24, 1989.

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

[23] Jonathan Katz and Vinod Vaikuntanathan. Signature schemes with bounded leakage resilience. In *ASIACRYPT*, pages 703–720, 2009.

[24] Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. *IACR Cryptology ePrint Archive*, 2013:94, 2013.

[25] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT*, pages 1–23, 2010.

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

[27] Daniele Micciancio and Petros Mol. Pseudorandom knapsacks and the sample complexity of lwe search-to-decision reductions. In *CRYPTO*, pages 465–484, 2011.

[28] Daniele Micciancio and Chris Peikert. Hardness of sis and lwe with small parameters. *IACR Cryptology ePrint Archive*, 2013:69, 2013.

[29] Daniele Micciancio and Panagiotis Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations. In *STOC*, pages 351–358, 2010.

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

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

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

[33] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, STOC '05, pages 84–93, New York, NY, USA, 2005. ACM.

[34] R.L. Rivest, L. Adleman, and M.L. Dertouzos. On data banks and privacy homomorphisms. In *Foundations on Secure Computation, Academia Press*, pages 169–179, 1978.

[35] Ronald L. Rivest. All-or-nothing encryption and the package transform. In *FSE*, pages 210–218, 1997.

[36] Claus-Peter Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.*, 53:201–224, 1987.

[37] Nigel P. Smart and Frederik Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In *Public Key Cryptography*, pages 420–443, 2010.

[38] Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In *EUROCRYPT*, pages 24–43, 2010.

A Min-Entropy and Leftover Hash Variants

Below we provide full, formal definitions of concepts used throughout our paper.

A.1 Min-Entropy and the Leftover Hash Lemma

Definition 14. A distribution $\mathcal{X}$ has **min entropy** $\geq k$, denoted $H_\infty(\mathcal{X}) \geq k$, if

$$\forall x \in \mathcal{X}, \mathbf{Pr}[\mathcal{X} = x] \leq 2^{-k}$$

Definition 15. (From [14]) For two random variables X and Y, the **average min-entropy** of X conditioned on Y, denoted $\tilde{H}_\infty(X|Y)$ is

$$\tilde{H}_\infty(X|Y) := -\log \mathop{\mathbf{E}}_{y \leftarrow Y}\left[\max_x \mathbf{Pr}[X = x|Y = y]\right] = -\log\left[\mathop{\mathbf{E}}_{y \leftarrow Y}\left[2^{-H_\infty(X|Y=y)}\right]\right]$$

Definition 16. (From [14]) For two random variables X and Y, the ϵ-**smooth average min-entropy** of X conditioned on Y, denoted $\tilde{H}^\epsilon_\infty(X|Y)$ is

$$\tilde{H}^\epsilon_\infty(X|Y) = \max_{(X',Y'):\Delta((X,Y),(X',Y'))<\epsilon} \tilde{H}_\infty(X'|Y')$$

Note that in particular, for any random variable X, given distributions $\mathcal{D}_Y \approx_s \mathcal{D}_Z$ with $Y \leftarrow \mathcal{D}_Y$, $Z \leftarrow \mathcal{D}_Z$, there exists some ϵ such that $\Delta(Y, Z) < \epsilon = negl(\eta)$, and

$$\tilde{H}^\epsilon_\infty(X|Y) \geq \tilde{H}^\epsilon_\infty(X|Z)$$

We now-restate a version of the leftover hash lemma [21] relating to matrix-vector multiplication in $\mathbb{Z}_q$, as it was stated in, for example, [18].

Lemma A.1. *[Leftover Hash Lemma] For a security parameter η, let $n = poly(\eta)$, let $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$ Let $\mathbf{s} \leftarrow \mathcal{D} \in \mathbb{Z}_q^n$, and let $k = H_\infty(\mathcal{D})$. If $m \log q \leq k - 2\log(\frac{1}{\epsilon}) + 2$ then $\Delta((\mathbf{C}, \mathbf{Cs})(\mathbf{C}, \mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m)) \leq \epsilon$. In particular, by setting $\epsilon = 2^{-\eta}$, if $m \log q \leq k - 2\eta + 2$ then $(\mathbf{C}, \mathbf{Cs}) \approx_s (\mathbf{C}, \mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m)$*

The leftover hash lemma can easily be generalized to the case where $\mathbf{s}$ has high conditional average min-entropy.

Lemma A.2. *[Generalized Leftover Hash Lemma] (from lemma 2.4 in [14]) For a security parameter η, let $n = poly(\eta)$, let $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$ Let $\mathbf{s} \leftarrow \mathcal{D} \in \mathbb{Z}_q^n$, let t be any random variable, and let $k = \tilde{H}_\infty(\mathbf{s}|t)$. If $m \log q \leq k - 2\log(\frac{1}{\epsilon}) + 2$ then $\Delta((\mathbf{C}, \mathbf{Cs}, t)(\mathbf{C}, \mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m), t) \leq \epsilon$. In particular, setting $\epsilon = 2^{-\eta}$, if $m \log q \leq k - 2\eta + 2$ then $(\mathbf{C}, \mathbf{Cs}, t) \approx_s (\mathbf{C}, \mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m, t)$*

An immediate consequence of the above lemma is the following corollary:

Corollary A.2.1 (Epsilon-Smooth Variant of LHL)**.** *For a security parameter η, let $n = poly(\eta)$, let $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$ Let $\mathbf{s} \leftarrow \mathcal{D} \in \mathbb{Z}_q^n$, let t be any random variable, and let $\tilde{H}_\infty^{\epsilon_1}(\mathbf{s}|t) \geq k$. If $m \log q \leq k - 2\log(\frac{1}{\epsilon_2}) + 2$ Then $\Delta((\mathbf{C}, \mathbf{Cs}, t)(\mathbf{C}, \mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m), t) \leq 2\epsilon_1 + \epsilon_2$.*

Proof. The definition of ϵ-smooth average min-entropy means there exists a random variable $\mathbf{s}'$ over the same domain as $\mathbf{s}$ and a random variable t' over the same domain as t such that $\Delta((\mathbf{s}, t)(\mathbf{s'}, t')) \leq \epsilon_1$, and $\tilde{H}_\infty(\mathbf{s'}|t') \geq k$. Lemma A.2 tell us that $\Delta((\mathbf{C}, \mathbf{Cs'}, t')(\mathbf{C}, \mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m), t') \leq \epsilon_2$. Furthermore, clearly $\Delta(t, t') \leq \epsilon_1$. Finally, since statistical distance is a metric, we can conclude $\Delta((\mathbf{C}, \mathbf{Cs}, t)(\mathbf{C}, \mathbf{u}, t)) \leq 2\epsilon_1 + \epsilon_2$ □

B More details about Parameter Setting

We now describe in more detail the constraints that drive our setting of parameters. We include full proofs of Lemma 4.2 and Lemma B.1, which drive the setting of many of our parameters.

When using Lemma 4.2, the following constraints affect our parameter setting:

1. **Statistical Indistinguishability:** There are three different places in our hybrid argument where we prove that two distributions are statistically indistinguishable.

   - In Lemma 4.2, we argue that the distribution $\mathbf{Zt_0} + \mathbf{e}$ is statistically close to $\mathbf{e}$, because the magnitude of each element of $\mathbf{Zt_0}$ is small. This argument requires that $\mathbf{e}$ be a discretized Gaussian distribution, rather than just a bounded distribution, as required by the original GSW scheme.
   - In Claim 1, inside our proof of Lemma 4.1, we use the leftover hash lemma to show we can replace $\mathbf{rA}$ with $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^n$. This step is part of the security proof of all variations on the RPKE scheme, but an artifact of our proof technique is that we consider an adversary who can see $\mathbf{r} \cdot \mathbf{e}$, which is $\ell = O(\log q)$ bits long. So for $\mathbf{r}$ of dimension m, we have $\tilde{H}_\infty(\mathbf{r}) = m - \ell$. The analogous step in the GSW security proof assumes $H_\infty(\mathbf{r}) = m$. This leads us to increase the value of m. In our scheme, m is set to $2n \log q + 3\eta$.

- Again in Lemma 4.1, in Claim 2, we use the leftover hash lemma to show that given $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^n$, we can replace $\mathbf{u} \cdot \mathbf{t}$ with $u' \xleftarrow{\$} \mathbb{Z}_q$. As described in our proof, the ϵ-smooth average min-entropy of $\mathbf{t}$ is $n - \ell - \lambda$, where λ is the the number of bits of leakage we can tolerate. Thus, we must set λ to a value that keeps $\tilde{H}_\infty(t)$ high enough for the leftover hash lemma to apply. That is how we arrive at $\lambda \leq n - 2\log q - 4\eta$.

2. ***DLWE* Considerations:** The security of our scheme is based on the hardness of two different *DLWE* problems: $DLWE_{n',q,\alpha}$, where the n' and α come from Lemma 4.2, and $DLWE_{n,q,\beta}$. For our scheme to be secure, the following three things need to be true:

   - $\frac{\alpha}{\beta} = negl(\eta)$. This is a necessary condition in our proof of Lemma 4.2.
   - $DLWE_{n',q,\alpha}$ is hard. We refer to Statement 1, which shows that this problem is at least as hard as $GapSVP_{n'q/\alpha}$, and to Statement 2, which says the best known algorithms for solving $GapSVP_{n'q/\alpha}$ run in time $2^{\tilde{\Omega}\left(\frac{n'}{log(n'q/\alpha)}\right)}$. This quantity should be at least super-polynomial in our security parameter for the scheme to be secure.
   - $DLWE_{n,q,\beta}$ is hard. Using the same theorems, we see that we need $2^{\tilde{\Omega}\left(\frac{n}{log(nq/\beta)}\right)}$ to be super-polynomial in η as well.

3. **Correctness:** The scheme needs $8(N+1)^L < \frac{q}{\beta}$, where L is the depth of the circuit, β is the error bound, and $N = (\log q + 1)n$, in order to ensure the noise never gets large enough to hamper accurate decryption.

Since our FHE scheme supports evaluation of circuits whose depth is polynomial in the security parameter as long as that polynomial is pre-specified, we know that there exists some constant c such that $L \leq \eta^c$. Let $\tau = \max\{L, \eta^2\}$. Setting the parameters as follows satisfies all of the hardness and correctness constraints for the scheme:

Let $n = \tau^4$. Let $q = 2^{\tau \log^2 \tau}$. Let $\beta = 2^{\log^2 \tau}$. Recall that $n' = (n - 2\eta)/\log q$, and let $\alpha = n'$.

Note that $\frac{\alpha}{\beta}$ is clearly negligible in η as required. Since the best algorithm for $GapSVP_{n'q/\alpha}$ runs in time $2^{\tilde{\Omega}\left(\frac{n'}{log(n'q/\alpha)}\right)}$, we look more closely at the exponent $\frac{n'}{\log(n'q/\alpha)}$ We can rewrite it as $\frac{n'}{\log(q)} = \frac{n-2\eta}{\log^2 q} = \frac{\tau^4 - 2\eta}{\tau^2 \log^4 \tau}$. Since $\tau \geq \eta^2$, we know that the above quantity is $\geq \frac{\eta^8 - 2\eta}{2\eta^4 \log^2 \eta} \geq \eta$ for $\eta \geq 16$. Thus the hardness is $2^{\tilde{\Omega}(\eta)}$.

Similarly, to bound the hardness of $GapSVP_{n,q,\beta}$ we consider the exponent of $2^{\frac{n}{\log(nq/\beta)}}$.

$$\begin{aligned}
\frac{n}{\log(nq/\beta)} &= \frac{n}{\log n + \log q - \log \beta} \\
&= \frac{\tau^4}{4\log\tau + \tau \log^2 \tau - \log^2 \tau} \\
&\geq \tau \\
&\geq \eta
\end{aligned}$$

This means that $DLWE_{n,q,\beta}$ is exponentially hard as well. Finally, we verify that our parameter settings maintain the correctness of the scheme: We need $8(N+1)^L < \frac{q}{\beta}$, and since we chose $\tau \geq L$,

it is sufficient to show $8(N+1)^\tau < \frac{q}{\beta}$. We can upper bound the left hand side of this inequality as follows:

$$\begin{aligned} 8(N+1)^\tau &\leq 8(2n\log q)^\tau \\ &= 8(2\tau^4\tau\log^2\tau)^\tau \\ &\leq 2^3 2^\tau \tau^{6\tau} \\ &= 2^{3+\tau+6\log^2\tau} \end{aligned}$$

Meanwhile, the right hand side is equal to $2^{(\tau-1)\log^2\tau}$, which is clearly greater than the left hand side for sufficiently high τ.

Finally, the number of bits of leakage we can support is $n - 2\log q - 4\eta = \tau^4 - 2\tau\log^2\tau - 4\eta = \eta^8 - 32\eta^4\log\eta - 4\eta$, which is positive for any $\eta \geq 3$.

B.1 Efficiency/Leakage Tradeoff

Recall that we can prove our scheme secure using the following alternate theorem, which gives us better efficiency but a lower leakage bound.

Lemma B.1. *For $n, m, n', q, \alpha, \beta$ such that $DLWE_{n',q,\alpha}$ and $DLWE_{n,q,\beta}$ are hard, if $\beta \geq \alpha nm$, there exists a distribution Lossy' such that $\tilde{\mathbf{A}} \leftarrow \mathsf{Lossy}' \approx_c \mathbf{U} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}$ and given $\mathbf{t} \xleftarrow{\$} \mathbb{Z}_q^n$, and $\mathbf{e} \leftarrow \overline{\Psi}_\beta$, $\tilde{H}_\infty^\epsilon(\mathbf{t}|\tilde{\mathbf{A}}, \tilde{\mathbf{A}}\mathbf{t}+\mathbf{e}) \geq n - \eta(\log m + 2\log n)$, where $\epsilon = negl(\eta)$.*

Proof. Define Lossy' as follows:

- Choose $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{m\times n'}$, $\mathbf{D} \xleftarrow{\$} \mathbb{Z}_q^{n'\times n}$, and $\mathbf{Z} \leftarrow \overline{\Psi}_\alpha^{m\times n}$, where $\beta \geq \alpha nm$ and $n'\log q \leq n - \eta(\log m + 2\log n) - 2\eta + 2$.
- Let $\tilde{\mathbf{A}} = \mathbf{CD} + \mathbf{Z}$
- output $\tilde{\mathbf{A}}$.

1. $\tilde{\mathbf{A}} \approx_c U \xleftarrow{\$} \mathbb{Z}_q^{m\times n}$:
   $\tilde{\mathbf{A}}$ is a series of *DLWE* instances, with the columns of $\mathbf{D}$ as the secrets $\mathbf{Z}$ containing the error terms, so as long as $DLWE_{n',q,\alpha}$ is hard, $\tilde{\mathcal{A}} \approx_c \mathbb{Z}_q^{m\times n}$.

2. $\tilde{H}_\infty^\epsilon(\mathbf{t}|\tilde{\mathbf{A}}\mathbf{t}+\mathbf{e}) \geq n - \eta(\log m + 2\log n)$, where $\epsilon = negl(\eta)$:

   - As in the proof of Lemma 4.2 we can think of $\mathbf{t} = \mathbf{t_0} + \mathbf{t_1}$ where $\mathbf{t_0} \xleftarrow{\$} \{0,1\}^m$, and $\mathbf{t_1} \xleftarrow{\$} \mathbb{Z}_q^m$, and it is sufficient to consider $\tilde{H}_\infty^\epsilon(\mathbf{t_0}|\tilde{\mathbf{A}}\mathbf{t}+\mathbf{e})$ where $\tilde{\mathbf{A}}\mathbf{t}+\mathbf{e} = \mathbf{CDt_0} + \mathbf{Zt_0} + \mathbf{CDt_1} + \mathbf{Zt_1} + \mathbf{e}$.
   - We now prove use the proof of Lemma B.4 in [3], to rewrite $\mathbf{Zt_0}+\mathbf{e}$ as $f(\mathbf{e}, F(\mathbf{Zt_0}))$, where with probability $\geq 1 - 2^{-\eta}$, the bit length of $F(\mathbf{Zt_0})$ is $\leq \eta(\log m + \log q)$. Thus,

there exists some $\epsilon_1 = negl(\eta)$ such that

$$\begin{aligned}
&\tilde{H}_\infty^{\epsilon_1}(\mathbf{t_0}|\mathbf{CDt_0} + \mathbf{CDt_1} + \mathbf{Zt_0} + \mathbf{Zt_1} + \mathbf{e}) \geq \\
&\tilde{H}_\infty^{\epsilon_1}(\mathbf{t_0}|\mathbf{CDt_0} + \mathbf{CDt_1} + \mathbf{Zt_1}, f(\mathbf{e}, F(\mathbf{Zt_1}))) = \\
&\tilde{H}_\infty^{\epsilon_1}(\mathbf{t_0}|\mathbf{CDt_0} + \mathbf{CDt_1} + \mathbf{Zt_1}, \mathbf{e}, F(\mathbf{Zt_1}))) \geq \\
&\tilde{H}_\infty^{\epsilon_1}(\mathbf{t_0}|\mathbf{CDt_0} + \mathbf{CDt_1} + \mathbf{Zt_1}, \mathbf{e}) - \eta(\log m + \log q)
\end{aligned}$$

- Since $\tilde{H}_\infty(\mathbf{t_0}|\mathbf{CDt_1} + \mathbf{Zt_1}, \mathbf{e}) \geq n$, by Lemma A.2, for our choice of n', we know that $(\mathbf{CDt_0} + \mathbf{CDt_1} + \mathbf{Zt_1}, \mathbf{e}) \approx_s (\mathbf{Cu_0} + \mathbf{CDt_1} + \mathbf{Zt_1}, \mathbf{e})$, where $\mathbf{u_0} \xleftarrow{\$} \mathbb{Z}_q^{n'}$, meaning their statistical distance is some $\epsilon_2 = negl(\eta)$. Therefore, for $\epsilon = \epsilon_1 + \epsilon_2 = negl(\eta)$, we know:

$$\begin{aligned}
&\tilde{H}_\infty^{\epsilon}(\mathbf{t_0}|\mathbf{CDt_0} + \mathbf{CDt_1} + \mathbf{Zt_0} + \mathbf{Zt_1} + \mathbf{e}) \geq \\
&\tilde{H}_\infty^{\epsilon}(\mathbf{t_0}|\mathbf{Cu_0} + \mathbf{CDt_1} + \mathbf{Zt_1}, \mathbf{e}) - \eta(\log m + \log q)
\end{aligned}$$

- Since each of $\mathbf{C}, \mathbf{u_0}, \mathbf{D}, \mathbf{Z}, \mathbf{t_1}, \mathbf{e}$, is independent of $\mathbf{t_0}$, $\tilde{H}_\infty^{\epsilon}(\mathbf{t_0}|\mathbf{Cu_0} + \mathbf{CDt_1} + \mathbf{Zt_1}, \mathbf{e}) = H_\infty(\mathbf{t_0}) = n$. Therefore there exists $\epsilon = negl(\eta)$ such that $\tilde{H}_\infty^{\epsilon}(\mathbf{t}|\tilde{\mathbf{A}}\mathbf{t} + \mathbf{e}) \geq n - \eta(\log m + \log q)$ as well.

□

This new lemma leads immediately to the following:

Corollary B.1.1. *The* LRGSW *scheme is resilient to* $\lambda \leq n - (2+\eta)\log q - \eta \log m - 4\eta$ *bits of leakage when* $\overline{\Psi}_\beta$ *is chosen so that* $\frac{\beta}{m} \geq \frac{n^2}{\log q}$.

Proof. This corollary is true as long as with the new parameter settings, the scheme still maintains its correctness, so $8(N+1)^L \leq \frac{q}{\beta}$ and its hardness: $DLWE_{n',q,\alpha}$, and $DLWE_{n,q,\beta}$, as well as the new requirement that $\frac{\beta}{m} \geq n\alpha$. If we choose an $\alpha = O(n')$, which we need for $DLWE_{n',q,\alpha}$ to be hard, then $\alpha \leq n/\log q$, so our setting of β is sufficient to meet this new requirement. Note that with these settings, β is no longer super-polynomial in η, and though q will remain superpolynomial in β, this allows for a much smaller value of q as well. For example, if we let $\tau = \max\{L, \eta^3\}$, $n = \tau^3$, $q = 2^{\tau \log n}$, $m = 2n\log q + 3\eta$, $n' = (n - 2\eta + 2)/\log q$, $\alpha = \tau^2$, $\beta = 3n^3\tau^3$, then we have:

- $DLWE_{n',q,\alpha}$ is hard:

  note that $n' = \frac{\tau^3 - 2\tau + 2}{3\tau \log \tau} \leq \tau^2 = \alpha$. So $2^{\frac{n'}{\log(n'q/\alpha)}} \geq 2^{n'/\log q}$. We can rewrite that exponent as $\frac{n - 2\eta + 2}{\log^2 q} = \frac{\eta^9 - 2\eta + 2}{81\eta^6 \log \eta}$. So for $\eta \geq 20$, we have $2^{n'/\log q} \geq 2^\eta$. Thus we can conclude that since $DLWE_{n',q,\alpha}$ takes time $2^{\tilde{\Omega}\left(\frac{n'}{\log(n'q/\alpha)}\right)}$ to solve, it is super-polynomially hard to solve in η.

- $DLWE_{n,q,\beta}$ is hard:

  Since $\beta > n$, we know that $\frac{n}{\log(nq/\beta)} > \frac{n}{\log q} = \frac{\tau^3}{3\tau \log \tau} \geq \tau/3 \geq \eta^6/3$. So $2^{\tilde{\Omega}\left(\frac{n}{\log nq/\beta}\right)}$ is exponential in η as well.

- The scheme is correct:
  We need to show: $8(N+1)^L \leq \frac{q}{\beta}$. First, we rewrite $N+1$.

$$\begin{aligned} N+1 &= n(\log q + 1) + 1 \\ &= \tau^3(\tau \log \tau^3) + \tau^3 + 1 \\ &= 3\tau^4 \log \tau + \tau^3 + 1 \\ &\leq 4\tau^4 \log \tau \end{aligned}$$

  So we have that $8(N+1)^L \leq 2^3 2^{2L} 2^{4\log\tau} 2^{\log\log\tau}$, and since $L \leq \tau$, we have, that this is $\leq 2^{2\tau+4\log\tau+3+\log\log\tau} \leq 2^{2\tau+5\log\tau}$.

  Meanwhile,

$$\begin{aligned} \frac{q}{\beta} &= 2^{\tau\log n}/(3n^3\tau^3) \\ &= \frac{1}{3}2^{\tau\log n - 3\log n - 3\log\tau} \\ &\geq 2^{3\tau\log\tau - 12\log\tau} \end{aligned}$$

  This quantity is $\geq 2^{2\tau+5\log\tau}$ for sufficiently high τ, (for example, if $\tau \geq 9$, meaning $\eta \geq 3$), so the scheme is secure.

□

C Non-Adaptive Leakage Resilience

We briefly note that it is straightforward to transform the GSW scheme (and, in fact, many of the existing FHE schemes based on *DLWE*) to achieve the much weaker concept of non-adaptive leakage resilience. Recall that non-adaptive leakage resilience requires an adversary to choose its leakage function independent of the scheme's public parameters. When the leakage is not dependent on the public key, it becomes straightforward to directly prove that the public key is indistinguishable from random using the leakage resilience results of [18] related in Theorem C.1 below. In fact, even when the scheme has an evk, it is straightforward to show that the evk is indistinguishable from random as well. This means that not only can we achieve leakage resilient full (bootstrapped) FHE in LRGSW if we are only concerned with non-adaptive leakage, but also that we can achieve it in many of the existing FHE schemes (for example the Brakerski scheme [5] and the Brakerski-Vaikuntanathan Scheme [7]) as long as we modify them to use binary secret keys. Furthermore, we can tolerate a higher amount of leakage.

Theorem C.1. *[Theorem 4 in [18]] Let $n, q \geq 1$ be integers, let $\mathcal{D}$ be any distribution over $\{0,1\}^n$ having min-entropy at least k, and let $\alpha, \beta > 0$ be such that $\alpha/\beta = negl(\eta)$. Then for any $\ell \leq \frac{k-\omega(\log n)}{\log q}$ there is a PPT reduction from $DLWE_{\ell,q,\alpha}$ to $DLWE_{n,q,\beta}(\mathcal{D})$*[5]

Theorem C.2. *The leveled* LRGSW *scheme is resilient to non-adaptive bounded leakage.*

[5]This notation means that $\mathbf{s} \leftarrow \mathcal{D}$.

Proof. (sketch)
To prove non-adaptive leakage resilience of LRGSW, we simply need to show that $(\mathbf{A}, h(\mathbf{s})) \approx_c (\mathbf{U}, h(\mathbf{s}))$, where

- in the first distribution $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$, $\mathbf{e} \xleftarrow{\$} \chi^m$, $\mathbf{b} = \mathbf{A}\mathbf{t} + \mathbf{e}$, and $\mathbf{K} = [\mathbf{b}||\mathbf{A}]$.
- in the second distribution $\mathbf{U} \xleftarrow{\$} \mathbb{Z}_q^{(m+1) \times n}$
- and in both distributions $\mathbf{t} \xleftarrow{\$} \{0,1\}^n$, and $\mathbf{s} = (1, -\mathbf{t}_1, \ldots, -\mathbf{t}_n)$.

Then, it will immediately follow that $\mathbf{RA}$ is computationally indistinguishable from random (where $\mathbf{R} \xleftarrow{\$} \{0,1\}^{N \times m+1}$), which as the authors explain in their security proof in [17], is enough to complete the proof.

Computational indistinguishability of the above distributions follows directly from Theorem C.1, as long as our dimension, modulus, and error bound simultaneously satisfy that theorem, imply a reduction from a hard instance of *GapSVP* (Statement 1), and maintain correctness of the scheme.

1. As explained in our parameter setting section, to maintain correctness of the scheme, we need $\beta(N+1)^L < \frac{q}{8}$. In practice, it will suffice to set $q = 2^{L \log^3 n}$.
2. To satisfy Theorem C.1 we need to choose α, β such that $\frac{\alpha}{\beta} = negl(\eta)$. Letting λ be the leakage bound of h, define $n' = \frac{n - \lambda - 2\eta}{\log q}$ Then the computational indistinguishability proof relies on the $DLWE_{(m-n')n,q,\alpha}$ and $DLWE_{n',q,\beta}$ problems. In fact, since $\beta > \alpha$ and $(m-n')n > n'$, and the *DLWE* problem is easier the higher the dimension and the lower the error bound, computational indistinguishability will follow as long as $DLWE_{(m-n')n,q,\alpha}$ is hard.
3. From Statement 1, we know that $DLWE_{(m-n')n,q,\alpha}$ is as hard as $GapSVP_{\gamma_1}$ $\gamma_1 = \frac{n^* q}{\alpha}$, with $n^* = (m-n')n$.
4. We also know that for $\alpha = \tilde{O}(n^*)$, this problem will be hard. So, for example, setting $\alpha = n^2 \log q$ and $\beta = Ln^{\log n}$ will work.

□

Corollary C.0.2. *The bootstrapped* LRGSW *scheme is resilient to non-adaptive bounded leakage.*

This follows using the same proof techniques as Theorem C.2, applied to the evaluation key, as well as to the public key.

Corollary C.0.3. *Both the Brakerski [5] and the Brakerski-Vaikuntanathan [7] schemes, with $q = superpoly(n)$ and a binary secret key are resilient to non-adaptive bounded leakage.*