Sanitization of FHE Ciphertexts

Léo Ducas[1], Damien Stehlé[2]

[1] Cryptology Group, CWI, Amsterdam, The Netherlands.
[2] ENS de Lyon, Laboratoire LIP (U. Lyon, CNRS, ENSL, INRIA, UCBL), France.

Abstract. By definition, fully homomorphic encryption (FHE) schemes support homomorphic decryption, and all known FHE constructions are bootstrapped from a Somewhat Homomorphic Encryption (SHE) scheme via this technique. Additionally, when a public key is provided, ciphertexts are also re-randomizable, e.g., by adding to them fresh encryptions of 0. From those two operations we devise an algorithm to sanitize a ciphertext, by making its distribution canonical. In particular, the distribution of the ciphertext does not depend on the circuit that led to it via homomorphic evaluation, thus providing circuit privacy in the honest-but-curious model. Unlike the previous approach based on noise flooding, our approach does not degrade much the security/efficiency trade-off of the underlying FHE. The technique can be applied to all lattice-based FHE proposed so far, without substantially affecting their concrete parameters.

1 Introduction

A fully homomorphic encryption (FHE) scheme enables the efficient and compact public transformation of ciphertexts decrypting to plaintexts $\mu_1, \ldots, \mu_k$, into a ciphertext decrypting to $\mathcal{C}(\mu_1, \ldots, \mu_k)$, for any circuit $\mathcal{C}$ with any number k of input wires. Since Gentry's first proposal of a candidate FHE scheme [Gen09a,Gen09b], plenty of FHE schemes have been proposed (see [SV10,DGHV10,BV11a,BV11b,Bra12,GHS12,GSW13], to name just a few).

A typical application of FHE is to offshore heavy computations on privacy-sensitive data: a computationally limited user encrypts its data, sends it to a distant powerful server, tells the server which operations to perform on the encrypted data, retrieves the result and decrypts. For this mainstream application, confidentiality, malleability and compactness seem sufficient. However, for other invaluable applications of FHE, another property, which we will call *ciphertext sanitizability*, has proved central. Statistical (resp. computational) ciphertext sanitizability requires that there exists a probabilistic polynomial time algorithm Sanitize taking as inputs a public key pk and a ciphertext c decrypting to a plain-

text μ under the secret key sk associated to pk, such that the distributions $\mathsf{Sanitize}(pk, c)$ and $\mathsf{Sanitize}(pk, \mathsf{Enc}(pk, \mu))$ are statistically (resp. computationally) indistinguishable, given pk *and* sk (here Enc refers to the encryption algorithm). For all applications we are aware of, computational ciphertext sanitizability suffices. Nevertheless, all known approaches (including ours) provide statistical ciphertext sanitizability.

Importance of ciphertext sanitizability. The ciphertext sanitizability property is closely related to the concept of (honest-but-curious) *circuit privacy*. The latter was introduced in the context of FHE by Gentry (see [Gen09a, Chapter 2]). Ciphertext sanitizability implies that if C_0 and C_1 are respectively obtained by the homomorphic evaluation of circuits $\mathcal{C}_0$ and $\mathcal{C}_1$ on honestly formed public key and ciphertexts, and if they decrypt to the same plaintext, then their distributions should be indistinguishable. This property is convenient in the following context: a first user wants a second user to apply a circuit on its plaintexts, but the first user wants to retain privacy of its plaintexts, while the second user wants to retain privacy of its circuit. A circuit private FHE with compact ciphertexts leads to a 2-flow protocol with communication cost bounded independently of the circuit size (this is not the case when directly using Yao's garbled circuit). The communication cost is proportional to the ciphertext bit-size and the number of data bits owned by the first user.

Two other potential applications of ciphertext sanitizability are mentioned in Section 5.

Flooding-based ciphertext sanitizability. The only known approach to realize ciphertext sanitizability, already described in [Gen09a, Chapter 21], is via the *noise flooding* technique (also called noise smudging and noise drowning). The ciphertexts of existing FHE schemes all contain a noise component, which grows (with respect to the Euclidean norm) and whose distribution gets skewed with homomorphic evaluations. Assume that at the end of the computation, its norm is below some bound B. The noise flooding technique consists in adding a statistically independant noise with much larger standard deviation. This may be done publicly by adding an encryption of plaintext 0 with large noise. The mathematical property that is used to prove ciphertext sanitizability is that the statistical distance between the uniform distribution over $[-B', B']$ and the uniform distribution over $[-B' + c, B' + c]$ for c such that $|c| \leq B$ is $\leq B/B'$ (see [AJL+12]). In the context of noise flooding, the parameter B' is taken of the order of $B \cdot 2^{\lambda}$, where λ refers

to the security parameter, so that the statistical distance is exponentially small.[3]

The noise flooding technique results in impractical schemes. To enable correct decryption, the scheme must tolerate much larger noise components: up to magnitude $B \cdot 2^{\lambda}$ instead of B, where B can be as small as $\lambda^{O(1)}$. In the case of schemes based on the Learning With Errors problem (LWE) [Reg09], the encryption noise rate α must be set exponentially small as a function of λ, to guarantee decryption correctness. Then, to ensure IND-CPA security against all known attacks costing $2^{o(\lambda)}$ operations, the LWE dimension n and modulus q must satisfy the condition $n \log q \geq \lambda^3$ up to poly-logarithmic factors in λ (lattice reduction algorithms [Sch87] may be used to solve LWE with parameters n, q and α in time $2^{n \log q / \log^2 \alpha}$ up to polylogarithmic factors in the exponent). This impacts key sizes, ciphertext expansion, and efficiency of encryption, decryption and homomorphic evaluation. For example, a ciphertext from the Brakerski-Vaikuntanathan FHE [BV11a] would have bit-size $O(n \log q) = \widetilde{O}(\lambda)$ if there is no need to support noise flooding, and $O(n \log q) = \widetilde{O}(\lambda^3)$ if it is to support noise flooding. A related impact is that the weakest hardness assumption on lattice problems allowing to get ciphertext sanitizability via noise flooding is the quantum hardness of standard worst case lattice problems such as SVP with approximation factors of the order of $2^{\sqrt{n}}$ in dimension n (this is obtained via the quantum reduction of [Reg09]).

Contribution. We propose a novel approach to realize the ciphertext sanitizability property, based on successive iterations of bootstrapping. In short, we replace the *flooding* strategy by a *soak-spin-repeat* strategy. It allows to take much smaller parameters (both in practice and in theory) and to rely on less aggressive hardness assumptions. In the case of LWE-based FHE schemes such as [BV11a,BV11b,Bra12,GSW13], the proposed scheme modification to realize ciphertext sanitizability allows to keep the same underlying hardness assumption (up to a small constant factor in the lattice approximation parameter) as for basic FHE without ciphertext sanitizability, and the same parameters (up to a small constant factor). On the downside, sanitizing a ciphertext requires successive iterations of bootstrapping. Note that the cost of bootstrapping has been recently decreased [AP14,DM15].

[3] Note that in some works, it is only required that $\sigma \geq B \cdot \lambda^{\omega(1)}$. These works consider resistance only against polynomial-time attackers. Here we consider the more realistic setting where attackers can have up to sub-exponential run-time $2^{o(\lambda)}$.

FHE bootstrapping consists in encrypting an FHE ciphertext under a second encryption layer, and removing the inner encryption layer by homomorphically evaluating the decryption circuit. If a ciphertext c decrypts to a plaintext μ, bootstrapping produces a ciphertext c' that also decrypts to μ, as if c was decrypted to μ and then μ re-encrypted to c'. The latter simplification is misleading, as one may think that c' is a fresh encryption of μ and hence that its distribution is canonical. This is incorrect. Homomorphic evaluation results in a ciphertext whose distribution may depend on the plaintexts underlying the input ciphertexts. In the context of bootstrapping, the input plaintexts are the bits of the decryption key and the bits of c. The distribution of ciphertext c' output by bootstrapping depends on the distribution of c.

Rather, we propose to bootstrap several times and inject some entropy in the ciphertext between each bootstrapping step. Suppose we start with two ciphertexts c_0 and c_1 decrypting to the same plaintext μ. We randomize them by adding a fresh encryption of 0. After a bootstrapping step, we obtain ciphertexts $c_0^{(1)}$ and $c_1^{(1)}$ decrypting to μ. By the data processing inequality, the statistical distance between them is no greater than before the bootstrapping. We then inject entropy in $c_0^{(1)}$ and $c_1^{(1)}$ to decrease their statistical distance by a constant factor, e.g., by a factor 2: this is achieved by adding a fresh encryption of 0. This process is iterated λ times, resulting in a pair of ciphertexts decrypting to μ and whose statistical distance is $\leq 2^{-\lambda}$. The process is akin to a dynamical system, approaching to a fixed point, canonical, distribution. This technique almost provides a solution to a problem suggested by Gentry in [Gen09a, page 30], stating that bootstrapping could imply circuit privacy.

It remains to explain how to realize the entropy injection step, whose aim is to decrease the statistical distance between the two ciphertexts by a constant factor. In the case of FHEs with a noise component, we use a *tiny flooding*. We add a fresh independent noise to the noise component, by publicly adding a fresh encryption of plaintext 0 to the ciphertext. As opposed to traditional flooding, this noise term is not required to be huge, as we do not aim at statistical closeness in one go. Both noise terms (the polluted one and the fresh one) may be of the same orders of magnitude.

Comparison with other approaches. We have already mentioned that in the case of FHE schemes based on LWE, the flooding based approach requires assuming that LWE with noise rate $\alpha = 2^{-\lambda}$ is hard, and hence setting $n \log q \geq \lambda^3$ (up to poly-logarithmic factors in λ). The inefficacy impact can be mitigated by performing the homomorphic evaluation of the circuit using small LWE parameters, bootstrapping the resulting

ciphertext to large LWE parameters, flooding with noise and then bootstrapping to small parameters (or, when it is feasible, switching modulus) before transmitting the result. This still involves one bootstrapping with resulting LWE parameters satisfying $n \log q \geq \lambda^3$. Our approach compares favorably in terms of sanitization efficiency, as it involves λ bootstrapping with parameters satisfying $n \log q \geq \lambda$ (still up to polylogarithmic factors).

In the context of (honest-but-curious) circuit privacy with communication bounded independently of the circuit size, van Dijk *et al.* [DGHV10, Appendix C] suggested using an FHE scheme and, instead of sending back the resulting ciphertext c, sending a garbling of a circuit taking as input the secret key and decrypting c. Using Yao's garbled circuit results in a communication cost that is at least λ times larger than the decryption circuit, which is itself at least linear in the ciphertext bit-length. Therefore, our approach compares favorably in terms of communication.

Related works. In [OPP14], Ostrovsky *et al.* study circuit privacy in the malicious setting: circuit privacy (or ciphertext sanitizability) must hold even if the public key and ciphertexts are not properly generated. This is a stronger property than the one we study in the present work. Ostrovsky *et al.* combine a compact FHE and a (possibly non-compact) homomorphic encryption scheme that enjoys circuit privacy in the malicious setting, to obtain a compact FHE which is maliciously circuit private. Their construction proceeds in two steps, and our work can be used as an alternative to the first step.

Noise flooding is a powerful technique to obtain new functionalities and security properties in lattice-based cryptography. As explained above, however, it leads to impractical schemes. It is hence desirable to find alternatives that allow for more efficient realizations of the same functionalities. For example, Lyubashesvky [Lyu09] used rejection sampling in the context of signatures (see also [Lyu12,DDLL13]). Alwen *et al.* [AKPW13] used the lossy mode of LWE to prove hardness of the Learning With Rounding problem (LWR) for smaller parameters than [BPR12]. LWR is for example used to designing pseudo-random functions [BPR12,BLMR13,BP14]. Langlois *et al.* [LSS14] used the Rényi divergence as an alternative to the statistical distance to circumvent noise flooding in encoding re-randomization for the Garg *et al.* cryptographic multi-linear map candidate [GGH13].[4] Further, in [BLP+13], Brakerski *et al.* introduced the first-is-errorless LWE problem to prove hardness of

[4] Note that the Garg *et al.* and hence its Langlois *et al.* improvement have recently been cryptanalysed [HJ15].

the Extended LWE problem without noise flooding, hence improving over a result from [OPW11]. They also gave a flooding-free hardness proof for binary LWE based on the hardness of Extended LWE, hence improving a hardness result from [GKPV10]. LWE with binary secrets was introduced to construct a leakage resilient encryption scheme [GKPV10]. Extended LWE was introduced to design a bi-deniable encryption scheme [OPW11], and was also used in the context of encryption with key-dependent message security [AP12]. The tools developed to circumvent noise flooding seem quite diverse, and it is unclear whether a general approach could be used.

ROADMAP. In Section 2, we provide some necessary reminders. In Section 3, we describe our ciphertext sanitation procedure. We instantiate our approach to LWE-based FHE schemes in Section 4.

2 Preliminaries

We give some background definitions and properties on Fully Homomorphic Encryption and probability distributions.

2.1 Fully homomorphic encryption

We let S denote the set of secret keys, P the set of public keys (which, in our convention includes what is usually referred to as the evaluation key), C the ciphertext space and M the message space. For simplicity, we set $M = \{0,1\}$. Additionally, we let C_μ denote the set of all ciphertexts that decrypt to $\mu \in M$ (under an implicitly fixed secret key $sk \in S$). We also assume that every ciphertext decrypts to a message: $C = \bigcup_{\mu \in M} C_\mu$ (i.e., decryption never fails). All those sets implicitly depend on a security parameter λ.

An FHE scheme (for S, P, M, C) is given by four polynomial time algorithms:

- a (randomized) key generation algorithm $\mathsf{KeyGen} : \{1^\lambda\} \to P \times S$,
- a (randomized) encryption algorithm $\mathsf{Enc} : P \times M \to C$,
- a (deterministic) decryption algorithm $\mathsf{Dec} : S \times C \to M$,
- a (deterministic) homomorphic evaluation function $\mathsf{Eval} : \forall k, P \times (M^k \to M) \times C^k \to C$.

Correctness requires that for any input circuit $\mathcal{C}$ with any number of input wires k, and for any $\mu_1, \ldots, \mu_k \in \{0,1\}$, we have (with overwhelming probability $1 - \lambda^{-\omega(1)}$ over the random coins used by the algorithms):

$$\mathsf{Dec}\left(sk, \mathsf{Eval}(pk, \mathcal{C}, (c_1, \ldots, c_k))\right) = \mathcal{C}(\mu_1, \ldots, \mu_k),$$

where $(pk, sk) = \mathsf{KeyGen}(1^\lambda)$ and $c_i = \mathsf{Enc}(pk, \mu_i)$ for all $i \leq k$.

Compactness requires that elements in C can be stored on $\lambda^{O(1)}$ bits.

Indistinguishability under chosen plaintext attacks (IND-CPA) requires that given pk (where $(pk, sk) = \mathsf{KeyGen}(1^\lambda)$), the distributions of $\mathsf{Enc}(pk, 0)$ and $\mathsf{Enc}(pk, 1)$ are computationally indistinguishable.

In addition to the above four algorithms, we define the function

$$\mathsf{Refresh}(pk, c) = \mathsf{Eval}\left(pk, \mathcal{C}_{\mathsf{Dec}}, (bk_1, \ldots, bk_k, c'_1, \ldots, c'_\ell)\right),$$

where $\mathcal{C}_{\mathsf{Dec}}$ refers to a polynomial-size circuit implementing Dec, $bk_i = \mathsf{Enc}(pk, sk_i)$ for all k bits sk_i of secret key sk, and $c'_i = \mathsf{Enc}(pk, c_i)$ for all ℓ bits c_i of ciphertext c. Note that $\mathsf{Refresh}$ is the typical bootstrapping step of current FHE constructions.

We assume that the bk_i's are given as part of pk, and do not impact IND-CPA security of the FHE scheme. This circular security assumption is standard in the context of FHE. We may circumvent it by using a sequence of key pairs (pk_j, sk_j) and encrypting the bits of sk_j under pk_{j+1} for all j. This drastically increases the bit-size of pk and does not provide FHE per say, but only homomorphic encryption for circuits of size bounded by any a priori known polynomial.

2.2 Properties of the statistical distance

For a probability distribution $\mathcal{D}$ over a countable set $\mathcal{S}$, we let $\mathcal{D}(x)$ denote the weight of $\mathcal{D}$ at x, i.e., $\mathcal{D}(x) = \Pr[\tilde{x} = x | \tilde{x} \leftarrow \mathcal{D}]$.

Let X and X' be two random variables taking values in a countable set $\mathcal{S}$. Let $\mathcal{D}$ and $\mathcal{D}'$ be the probability distributions of X and X'. The statistical distance $\Delta(X, X')$ is defined by

$$\Delta(X, X') = \frac{1}{2} \sum_{x \in \mathcal{S}} |\mathcal{D}(x) - \mathcal{D}'(x)|.$$

By abuse of notation, we aso write $\Delta(\mathcal{D}, \mathcal{D}')$. Note that $0 \leq \Delta(X, X') \leq 1$ always holds.

Assuming that $\delta = \Delta(X, X') < 1$, the intersection distribution $\mathcal{C} = \mathcal{D} \cap \mathcal{D}'$ is defined over $\mathcal{S}$ by $\mathcal{C}(x) = \frac{1}{1-\delta} \min(\mathcal{D}(x), \mathcal{D}'(x))$. It may be checked that $\mathcal{C}$ is indeed a distribution (i.e., $\sum_{x \in S} \mathcal{C}(x) = 1$), by using the following identity, holding for any reals a and b: $2\min(a, b) = a + b - |a - b|$. We also define the mixture of two distributions $\mathcal{B} = \alpha \cdot \mathcal{D} + (1 - \alpha) \cdot \mathcal{D}'$ for $0 \leq \alpha \leq 1$ by $\mathcal{B}(x) = \alpha \cdot \mathcal{D}(x) + (1 - \alpha) \cdot \mathcal{D}'(x)$. If X and X' are random variables with distributions $\mathcal{D}$ and $\mathcal{D}'$ respectively, then $\mathcal{B}$ is the density

function of the random variable obtained with the following experiment: sample a bit from the Bernoulli distribution giving probability α to 0; if the bit is 0, then return a sample from X; if the bit is 1, then return a sample from X'.

We will use the following two lemmas.

Lemma 2.1. *For any $\delta \in [0,1)$ and any distributions $\mathcal{B}, \mathcal{B}'$ such that $\delta \geq \Delta(\mathcal{B}, \mathcal{B}')$, there exist two distributions $\mathcal{D}$ and $\mathcal{D}'$ such that:*

$$\mathcal{B} = (1-\delta) \cdot \mathcal{B} \cap \mathcal{B}' + \delta \cdot \mathcal{D} \quad \text{and} \quad \mathcal{B}' = (1-\delta) \cdot \mathcal{B} \cap \mathcal{B}' + \delta \cdot \mathcal{D}'.$$

Proof. Let $\mathcal{C} = \mathcal{B} \cap \mathcal{B}'$. One builds $\mathcal{D}$ as the renormalization to sum 1 of the non-negative function $\mathcal{B}(x) - (1-\delta) \cdot \mathcal{C}(x)$, and proceeds similarly for $\mathcal{D}'$. □

Lemma 2.2. *For any $\alpha \in [0,1]$ and any distributions $\mathcal{C}, \mathcal{D}, \mathcal{D}'$, we have*

$$\Delta\left((1-\alpha) \cdot \mathcal{C} + \alpha \cdot \mathcal{D}, (1-\alpha) \cdot \mathcal{C} + \alpha \cdot \mathcal{D}'\right) \;=\; \alpha \cdot \Delta(\mathcal{D}, \mathcal{D}').$$

Proof. Let $\mathcal{B} = (1-\alpha)\mathcal{C} + \alpha\mathcal{D}$ and $\mathcal{B}' = (1-\alpha)\mathcal{C} + \alpha\mathcal{D}'$. We derive

$$\begin{aligned} 2 \cdot \Delta(\mathcal{B}, \mathcal{B}') &= \sum |((1-\alpha)\mathcal{C}(x) + \alpha\mathcal{D}(x)) - ((1-\alpha)\mathcal{C}(x) + \alpha\mathcal{D}'(x))| \\ &= \sum |\alpha\mathcal{D}(x) - \alpha\mathcal{D}'(x)| \\ &= 2\alpha \cdot \Delta(\mathcal{D}, \mathcal{D}'). \end{aligned}$$

This completes the proof. □

The following lemma is at the core of our main result. It states that if applying a randomized function f to any two inputs $a, b \in \mathcal{S}$ leads to two somewhat close-by distributions, then iterating f several times provides extremely close distributions.

Lemma 2.3. *Let $\delta \in [0,1]$ and $f : \mathcal{S} \to \mathcal{S}$ be a randomized function such that $\Delta(f(a), f(b)) \leq \delta$ holds for all $a, b \in \mathcal{S}$. Then:*

$$\forall k \geq 0, \forall a, b \in \mathcal{S},\ \Delta(f^k(a), f^k(b)) \leq \delta^k.$$

Proof. We prove the result by induction on $k \geq 0$. It trivially holds for $k = 0$. We now assume that $\Delta(f^k(a), f^k(b)) \leq \delta^k$ holds for all $a, b \in \mathcal{S}$ and some $k \geq 0$, and aim at showing that $\Delta(f^{k+1}(a), f^{k+1}(b)) \leq \delta^{k+1}$.

By Lemma 2.1, there exist two distributions $\mathcal{D}$ and $\mathcal{D}'$ such that:

$$\begin{aligned} f^k(a) &= (1-\delta^k) \cdot f^k(a) \cap f^k(b) + \delta^k \cdot \mathcal{D}, \\ f^k(b) &= (1-\delta^k) \cdot f^k(a) \cap f^k(b) + \delta^k \cdot \mathcal{D}'. \end{aligned}$$

By composing with f, we obtain that:

$$f^{k+1}(a) = (1-\delta^k) \cdot f(f^k(a) \cap f^k(b)) + \delta^k \cdot f(\mathcal{D}),$$
$$f^{k+1}(b) = (1-\delta^k) \cdot f(f^k(a) \cap f^k(b)) + \delta^k \cdot f(\mathcal{D}').$$

Now, Lemma 2.2 implies that

$$\Delta(f^{k+1}(a), f^{k+1}(b)) = \delta^k \cdot \Delta\left(f(\mathcal{D}), f(\mathcal{D}')\right).$$

To complete the proof, note that

$$\begin{aligned}
\Delta\left(f(\mathcal{D}), f(\mathcal{D}')\right) &= \sum_{x \in \mathcal{S}} \Big| \sum_{a' \in \mathcal{S}} \mathcal{D}(a') \Pr_f[f(a') = x] - \sum_{b' \in \mathcal{S}} \mathcal{D}'(b') \Pr_f[f(b') = x] \Big| \\
&= \sum_{x \in \mathcal{S}} \Big| \sum_{a', b' \in \mathcal{S}} \mathcal{D}(a')\mathcal{D}'(b') \big[\Pr_f[f(a') = x] - \Pr_f[f(b') = x] \big] \Big| \\
&\leq \sum_{a', b' \in \mathcal{S}} \mathcal{D}(a')\mathcal{D}'(b') \Big| \sum_{x \in \mathcal{S}} \big[\Pr_f[f(a') = x] - \Pr_f[f(b') = x] \big] \Big|.
\end{aligned}$$

The latter quantity is $\leq \delta$, by assumption. □

3 Sanitization of ciphertexts

We first formally state the correctness and security requirements of a sanitization algorithm for an encryption scheme $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ with secret key space S, public key space P, message space M and ciphertext space C.

Definition 3.1 (Sanitization algorithm). *A polynomial-time (randomized) algorithm* $\mathsf{Sanitize} : P \times C \to C$ *is said to be message-preserving if the following holds with probability* $\geq 1 - \lambda^{-\omega(1)}$ *over the choice of* $(pk, sk) = \mathsf{KeyGen}(1^\lambda)$*:*

$$\forall c \in C : \mathsf{Dec}(sk, \mathsf{Sanitize}(pk, c)) = \mathsf{Dec}(sk, c).$$

It is said (statistically) sanitizing if the following holds with probability $\geq 1 - 2^{-\lambda}$ *over the choice of* $(pk, sk) = \mathsf{KeyGen}(1^\lambda)$*: for all* $c, c' \in C$ *such that* $\mathsf{Dec}(sk, c) = \mathsf{Dec}(sk, c')$*, we have*

$$\Delta(\mathsf{Sanitize}(pk, c) | (pk, sk), \mathsf{Sanitize}(pk, c') | (pk, sk)) \leq 2^{-\lambda}.$$

In what follows, we fix the key pair $(pk, sk) = \mathsf{KeyGen}(1^\lambda)$ and assume it is given. To simplify notations, we will omit the conditioning of distributions $\mathsf{Sanitize}(pk, c)$ and $\mathsf{Sanitize}(pk, c')$ by (pk, sk).

3.1 Generic algorithm

For each $\mu \in M$, we let C^*_μ denote $\mathsf{Refresh}(pk, C_\mu)$.[5] We assume that one may build an efficient randomized algorithm $\mathsf{Rerand} : P \times C \mapsto C$ such that

$$c \in C^*_\mu \;\Rightarrow\; \mathsf{Rerand}(pk, c) \in C_\mu. \tag{1}$$

We choose a cycle parameter $\kappa > 0$ as an implicit function of λ. We now define

$$\mathsf{Wash} : (pk, c) \mapsto \mathsf{Rerand}(pk, \mathsf{Refresh}(pk, c)),$$

and $\mathsf{Sanitize}(pk, c)$ as the κ-th iteration of $(pk, c) \mapsto \mathsf{Wash}(pk, c)$. The following statement follows from the definitions.

Lemma 3.2 (Sanitize is message-preserving). *Under assumption (1), algorithms* Wash *and* Sanitize *are message-preserving.*

In practical FHEs, implication (1) would typically only hold with overwhelming probability $1 - \lambda^{-\omega(1)}$ over the random coins used during key generation, encryption and execution of Rerand: guaranteeing that those bounds always hold requires larger parameters, leading to slightly worse practical performance. If so, the membership $\mathsf{Sanitize}(pk, c) \in C_\mu$ of Lemma 3.2 holds only with overwhelming probability. This impacts our main result, Theorem 3.3 below, as follows: the statistical distance bound becomes

$$\Delta\left(\mathsf{Sanitize}(pk, c), \mathsf{Sanitize}(pk, c')\right) \leq \delta^\kappa + \kappa \cdot \lambda^{-\omega(1)}.$$

Such a bound does not allow to prove that all sub-exponential attacks can be prevented. To obtain this, one can increase the scheme parameters a little to enable correct decryption with probability $\geq 1 - 2^{-\Omega(\lambda)}$. Then the statistical distance bound of Theorem 3.3 becomes

$$\Delta\left(\mathsf{Sanitize}(pk, c), \mathsf{Sanitize}(pk, c')\right) \leq \delta^\kappa + \kappa \cdot 2^{-\Omega(\lambda)},$$

hence providing security against all sub-exponential attackers.

[5] To give intuition, note that in our LWE instantiation, the set C^*_μ will correspond to low-noise ciphertexts decrypting to μ.

3.2 Security

Note that the trivial case $C_\mu = C_\mu^*$ and $\mathsf{Rerand}(pk, \cdot) = \mathrm{Id}$ with $\mathsf{Refresh}$ replaced by the identity map fits our assumptions, but is exactly the possibly non-sanitized initial scheme. For security, we require that $\mathsf{Rerand}(pk, c)$ does introduce some ambiguity about c, but unlike the previous flooding-based techniques, we do not require that it completely updates the distribution of c. More precisely:

Theorem 3.3 (Sanitization security). *Assume that (1) holds, and that*

$$\forall \mu \in M, \forall c, c' \in C_\mu^*,\ \ \Delta\left(\mathsf{Rerand}(pk, c), \mathsf{Rerand}(pk, c')\right) \leq \delta$$

for some constant $\delta \in [0, 1]$. Then

$$\Delta\left(\mathsf{Sanitize}(pk, c), \mathsf{Sanitize}(pk, c')\right) \leq \delta^\kappa.$$

In particular if $\delta^\kappa \leq 2^{-\lambda}$, then $\mathsf{Sanitize}$ *is statistically sanitizing.*

Proof. The result is obtained by applying Lemma 2.3, with $\mathcal{S} = C_\mu^*$, $k = \kappa$ and f set to $c \mapsto \mathsf{Rerand}(pk, c)$. □

4 Application to some FHE schemes

We now apply our technique to LWE-based schemes built upon Regev's encryption scheme [Reg09]. These include the schemes following the designs of [BV11a] and [GSW13]. We comment practical aspects for HElib [HS] and FHEW [DM].

Our technique can also be applied to Gentry's original scheme and its variants [Gen09a,Gen09b,Gen10,SV10,SS10]. It may also be applied to the FHE scheme "based on the integers" of van Dijk *et al.* [DGHV10] and its improvements (see [CS15] and references therein).

4.1 Rerandomizing a Regev ciphertext

We let $\mathrm{LWE}_{\boldsymbol{s}}^q(\mu, \eta)$ denote the set of LWE-encryptions of $\mu \in M$ under key $sk = \boldsymbol{s} \in \mathbb{Z}_q^n$ with modulus q and error rate less than η, i.e., the set

$$\mathrm{LWE}_{\boldsymbol{s}}^q(\mu, \eta) = \left\{(\boldsymbol{a}, \langle \boldsymbol{a}, \boldsymbol{s}\rangle + \mu \cdot \lfloor q/2 \rfloor + e) \in \mathbb{Z}_q^{n+1} \text{ such that } |e| < \eta q\right\}.$$

One may recover μ from an element $(\boldsymbol{c}_1, c_2)$ from $\mathrm{LWE}_{\boldsymbol{s}}^q(\mu, \eta)$ by looking at the most significant bit of $c_2 - \langle \boldsymbol{c}_1, \boldsymbol{s}\rangle \bmod q$. Correctness of decryption is ensured up to $\eta < 1/4$.

We assume that the public key pk contains $\ell = O(n \log q)$ encryptions of 0, called rerandomizers:

$$\forall i \leq \ell,\ r_i = (\boldsymbol{a}_i, b_i = \langle \boldsymbol{a}, \boldsymbol{s} \rangle + e_i) \in \mathrm{LWE}^q_{\boldsymbol{s}}(0, \eta).$$

We also assume that the $\boldsymbol{a}_i$'s are uniform and independent (they have been freshly sampled).

For a ciphertext $c \in \mathrm{LWE}^q_{\boldsymbol{s}}(\mu, \eta)$, we may now define

$$\mathsf{Rerand}(pk, c) = c + \sum_i \varepsilon_i r_i + (\boldsymbol{0}, f),$$

where the ε_i's are uniformly and independently sampled from $\{0, \pm 1\}$, and f is sampled uniformly in an interval $[-B, B]$ for some B to be chosen below. By an appropriate version of the leftover hash lemma (see, e.g., [Reg09, Section 5]), writing

$$c' = c + \sum_i \varepsilon_i r_i = (\boldsymbol{a}', \langle \boldsymbol{a}', \boldsymbol{s} \rangle + \mu \lfloor q/2 \rfloor + e'),$$

we know that $\boldsymbol{a}'$ is (within exponentially small distance from) uniform in $\mathbb{Z}_q^n$, independently of c. That is, the only information about c contained in c' is carried by e' (and plaintext μ). Additionally, we have that $|e'| < (\ell + 1) \cdot \eta \cdot q$.

To conclude, it remains to remark that for any $x, y \in [-(\ell+1)\eta q, (\ell + 1)\eta q]$, we have:

$$\Delta(x + U([-B, B]), y + U([-B, B])) \leq \frac{(\ell + 1)\eta q}{B} =: \delta.$$

Therefore, for any $c_0, c_1 \in \mathrm{LWE}^q_{\boldsymbol{s}}(\mu, \eta)$, it holds that

$$\Delta(\mathsf{Rerand}(pk, c_0), \mathsf{Rerand}(pk, c_1)) \leq \delta,$$

and that

$$\mathsf{Rerand}(pk, c_0), \mathsf{Rerand}(pk, c_1) \in \mathrm{LWE}^q_{\boldsymbol{s}}(\mu, \frac{(\delta + 1)B}{q}).$$

To ensure the correctness of decryption after rerandomization, we may set the parameters so that $(\delta + 1)B/q < 1/4$.

4.2 Application to FHE à la [BV11a]

For simplicity, we only present the case of the (non-ring) LWE-based FHE scheme of [BV11a].

Let us first recall how an FHE scheme is bootstrapped from a given SHE scheme. Assume the SHE scheme supports the homomorphic evaluation of any (binary) circuit of multiplicative depth f, and that the decryption operation can be implemented by a circuit of multiplicative depth $g < f$. The SHE scheme is bootstrapped to an FHE scheme using the $\mathsf{Refresh}$ function, and evaluates sub-circuit of depth $f - g \geq 1$ between each refreshing procedure.

The construction of the SHE from [BV11a] is made more efficient by the use of modulus switching. This induces a leveled ciphertext-space: for each $i \leq f$, the ciphertext space C^i is $\mathrm{LWE}_{\boldsymbol{s}}^{q_i}(\cdot, \eta)$ for a sequence of $q_0 > q_1 > \cdots > q_f$ and a fixed $\eta < 1/4$.

The modulus switching technique allows, without any key material, to map $\mathrm{LWE}_{\boldsymbol{s}}^{q}(\mu, \eta)$ to $\mathrm{LWE}_{\boldsymbol{s}}^{q'}(\mu, \eta')$ where $\eta' = \eta + n \cdot (\log\ n)^{O(1)}/q'$ (or even $\eta + \sqrt{n} \cdot (\log\ n)^{O(1)}/q'$ allowing up to negligible probability of incorrect computation).

By sequentially applying so-called ciphertext tensoring, key switching and modulus switching steps, one may compute—given appropriate key material—a ciphertext $c'' \in \mathrm{LWE}_{\boldsymbol{s}}^{q_{i+1}}(\mu\mu', \eta)$ from two ciphertexts $c \in \mathrm{LWE}_{\boldsymbol{s}}^{q_i}(\mu, \eta)$ and $c' \in \mathrm{LWE}_{\boldsymbol{s}}^{q_i}(\mu', \eta)$, on the condition that $q_{i+1}/q_i \geq n \cdot (\log\ n)^{O(1)}$.

Technically, the $\mathsf{Refresh}$ function may only be applied to ciphertext $c \in C^f$, as the naive decryption of ciphertexts with a large modulus $q_i > q_f$ could require larger multiplicative depth. To extend $\mathsf{Refresh}$ over the whole ciphertext space, one can switch the modulus to the last level beforehand, which, for appropriate parameters q_i's does not affect the error bound.

Instantiating Rerand. Let $C_\mu^g = \mathrm{LWE}_{\boldsymbol{s}}^{q_g}(\mu, \eta)$. Then, according to the description above, we have $C_\mu^* = \mathsf{Refresh}(pk, C_\mu) \subseteq C_\mu^g$. We use the Rerand function described in Section 4.1, with $q = q_g$.

To ensure the correctness of the whole scheme, it suffices that

$$(\eta(\ell + 1) + B/q_g) + n(\log n)^{O(1)}/q_f < 1/4.$$

Setting $B \geq 2\eta(\ell + 1)q_g$, $\eta < 1/(8(\ell + 1))$ and $q_f \geq 8n^{1+o(1)}$ allows to fulfill the conditions of Theorem 3.3 for some $\delta \leq 1/2$.

A larger gap $q_f/q_g > n^{f-g}$ is beneficial to our sanitizing technique, as it allows one to choose $\delta \approx 1/n^{f-g-1}$, and therefore decrease the length κ

of the washing program: soaking in a large bucket makes the soak-spin-repeat program shorter. A striking example is given below.

Application to HElib. It turns out that the parameters given in the bootstrappable version of HElib [HS15] lead to $\kappa = 1$ or 2, which means that, in this setting, the flooding strategy is, or almost is, already applicable. Indeed, choosing for example the set of parameters corresponding to $n = \phi(m) = 16384$, we have $f = 22$ and $f - g = 10$. The parameters q_f and q_g are not given, yet it is typical to have $q_{i+1}/q_i = \sqrt{n} \cdot (\log\ n)^{O(1)}$ (guaranteeing correctness only with probability $1 - n^{-\omega(1)}$). We can therefore assume that a single soaking step may achieve $\delta \leq n/\sqrt{n}^{f-g} \approx 2^{-14 \cdot 10/2 + 14} = 2^{-56}$. This gives, according to [HS15] a batch sanitization procedure of 720 ciphertexts in 500 to 1000 seconds with the current software [HS15,HS] (on an IntelX5570 processor at 2.93GHz, with a single thread).

4.3 Application to FHEW

Because the constructions à la [BV11a] rely on the hardness of LWE with inverse noise rate $2^{(\log n)^c}$ for some $c > 1$ in theory (and necessarily larger than $\sqrt{n}^f \approx 2^{14 \cdot 22/2} = 2^{154}$ in practice), it is not so surprising that the implementations allow to straightforwardly apply the flooding strategy in practice (which theoretically requires assuming the hardness of LWE with inverse noise rate $2^{\sqrt{n}}$). It is therefore more interesting to study our sanitization strategy for FHE schemes based on the hardness of LWE with inverse polynomial noise rates [GSW13,BV14,AP14], in particular the concrete instantiation FHEW proposed in [DM15]. For comparison, the security of this scheme is based on a (Ring)-LWE problem [LPR10] with inverse noise rate $\approx 2^{32}$.

Warning. The following analysis is only given as an attempt to estimate the practical cost of our technique, yet the application with the original parameters of FHEW is not to be considered secure. Indeed, for efficiency purposes, the authors [DM15] have chosen to guarentee correctness only *heuristically*, and with a rather large failure probability $\approx 2^{-45}$. Because decryption correctness is essential in our argument (see remark at the end of Section 3.1), a serious implementation should first revise the parameters to *provably* ensure decryption correctness with *higher probability*.

Sanitizing FHEW. We proceed to modify the original scheme recalled in Figure 1 to implement the sanitizing strategy, as described in Figure 2.

This scheme uses two plaintext moduli $t = 2, 4$, and this extends the definition of LWE ciphertexts as follows.

$$\mathrm{LWE}_{\boldsymbol{s}}^{t:q}(\mu, \eta) = \left\{ (\boldsymbol{a}, \langle \boldsymbol{a}, \boldsymbol{s} \rangle + \mu \cdot \lfloor q/t \rfloor + e) \in \mathbb{Z}_q^{n+1} \text{ such that } |e| < \eta q \right\}.$$

Correct decryption now requires $\eta < q/(2t)$. The scheme uses two LWE dimensions: dimension $n = 500$ for a first secret vector $\boldsymbol{s}$, and dimension $N = 1024$ for a second secret vector $\boldsymbol{z}$. It also switches between two ciphertext moduli $q = 2^9$ and $Q = 2^{32}$. According to the analysis from [DM15], the parameters allow to securely encrypt in dimension N and modulus Q, with a (discrete) Gaussian error of standard deviation $\varsigma = 1.4$.

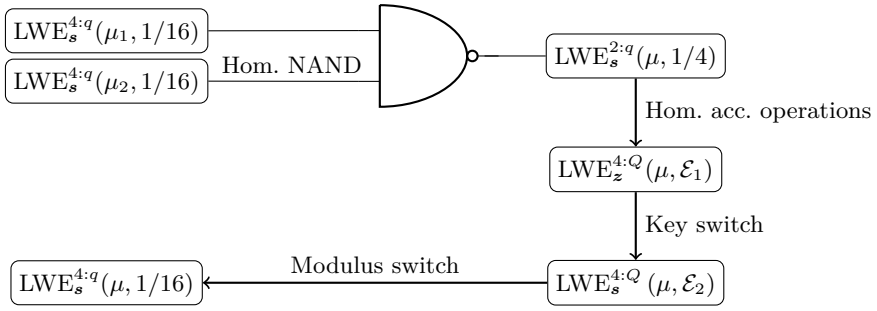

Fig. 1. Original cycle of FHEW

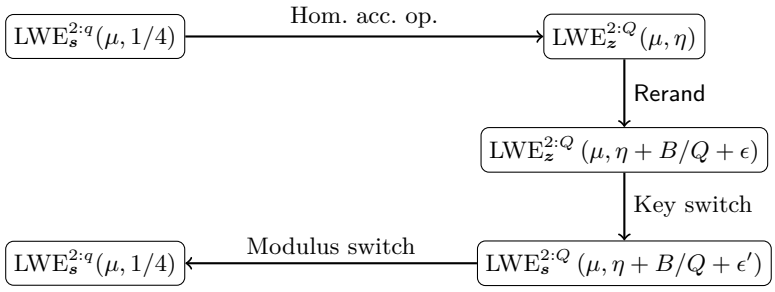

Fig. 2. Washing cycle for FHEW. The only internal modification required is setting $u = Q/4 + 1$ instead of $Q/8 + 1$. See [DM15] for more details.

Following the heuristic central-limit estimate of [DM15], the first step of Figure 2 (i.e., the homomorphic accumulator operations) returns a ciphertext with a Gaussian-like error of standard deviation $\approx 2^{18}$, so that error is of magnitude less than $Q\eta = 2^{21}$ (with probability $\geq 1 - 2^{-45}$). Also, the choice $\varsigma = 1.4$ makes the error introduced by the key switch negligible. Similarly, the re-randomization of the $\boldsymbol{a}$ part of the ciphertext $c = (\boldsymbol{a}, b)$ using fresh encryption of 0 with error parameter ς given in the public-key ensure that (with notation similar than in the previous section) $b = Q\eta + Q\varepsilon$ where ε is small compared to η.

Not having to compute any NAND also improves the error tolerance from 1/16 to 1/4. We may, in return, introduce a soaking noise of parameter B such that $Bq/Q \approx 3q/16$, that is $B \approx 2^{29}$. This results in $\delta = b/B \approx 2^{-8}$.

In conclusion, setting κ between 8 and 16 (depending on the desired security level) should suffice to achieve appropriate statistical sanitation. This gives sanitization of a single ciphertext in 5 to 10 seconds with the current software [DM] (on an unspecified Intel processor at 3Ghz, with a single thread).

5 Conclusion and open problems

We have shown that both in theory and in practice, the sanitization of FHE ciphertexts can be performed at a reasonable cost and without substantial modification of current schemes. It remains that FHE is too slow for many real world scenarios and SHE is often much preferable. In a credible scenario where the circuit to evaluate is shallow, with potentially many inputs but few outputs, the best strategy may be to use HElib in SHE mode for the main computation, and sanitizing the final result using FHEW.

When applied to circuit privacy, our approach only provides passive (honest-but-curious) security. Standard (interactive or not) zero-knowledge proofs help prevent malicious attackers using fake public keys and/or fake ciphertexts. Yet ad-hoc techniques surely need to be developed: with public key size of several gigabytes, the statement to be proved is gigantic.

A worthy remark toward this goal, is that malicious ciphertexts are easily tackled once the honest generation of the public key has been established. Indeed, a single Refresh operation on each input ciphertexts will ensure that they are in the subset of valid ciphertexts (formally proving such statement using, e.g., the circuit privacy definition of [OPP14]

is rather direct). This strategy may effectively reduce interactivity in secure multi-party computation (MPC) protocols based on FHE, and offer amortization of an initial zero-knowledge proof on the public key.

Ciphertext sanitizability may have further applications in MPC based on FHE, or, more precisely, based on Threshold FHE. Threshold FHE is a variant of FHE in which 1- several parties can execute a key generation protocol to generate a common public key and secret key shares, and 2- to decrypt, the parties execute a decryption protocol using their secret key shares. It is theoretically possible to generically convert any FHE into a Threshold FHE, but this is too cumbersome for practical use: in particular, it results in a significant number of communication rounds. Instead, Threshold FHE schemes have been designed directly by modifying existing FHE schemes [AJL+12,LTV12,CLO+13,CM15,MW15], eventually allowing for MPC in two communication rounds [MW15]. A crucial security property of Threshold FHE, called simulatability of partial decryptions, is that the partial decryptions obtained by individual users do not reveal anything about the confidential data of the other users. Ciphertext sanitization may help enforce this property without resorting to noise flooding.

Acknowledgments. The authors thank Lisa Kohl, Ron Steinfeld and Daniel Wichs for helpful discussions. This work has been supported by an NWO Free Competition Grant and by ERC Starting Grant ERC-2013-StG-335086-LATTAC.

References

[AJL+12] G. Asharov, A. Jain, A. López-Alt, E. Tromer, V. Vaikuntanathan, and D. Wichs. Multiparty computation with low communication, computation and interaction via threshold FHE. In *Proc of EUROCRYPT*, volume 7237 of *LNCS*, pages 483–501. Springer, 2012.

[AKPW13] J. Alwen, S. Krenn, K. Pietrzak, and D. Wichs. Learning with rounding, revisited - new reduction, properties and applications. In *Proc. of CRYPTO*, volume 8042 of *LNCS*, pages 57–74. Springer, 2013.

[AP12] J. Alperin-Sheriff and C. Peikert. Circular and KDM security for identity-based encryption. In *Proc. of PKC*, volume 7293 of *LNCS*, pages 334–352. Springer, 2012.

[AP14] J. Alperin-Sheriff and C. Peikert. Faster bootstrapping with polynomial error. In *Proc. of CRYPTO*, volume 8616 of *LNCS*, pages 297–314. Springer, 2014.

[BLMR13] D. Boneh, K. Lewi, H. W. Montgomery, and A. Raghunathan. Key homomorphic prfs and their applications. In *Proc. of CRYPTO*, volume 8042 of *LNCS*, pages 410–428. Springer, 2013.

[BLP+13] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé. Classical hardness of learning with errors. In *Proc. of STOC*, pages 575–584. ACM, 2013.

[BP14] A. Banerjee and C. Peikert. New and improved key-homomorphic pseudorandom functions. In *Proc. of CRYPTO*, volume 8616 of *LNCS*, pages 353–370. Springer, 2014.

[BPR12] A. Banerjee, C. Peikert, and A. Rosen. Pseudorandom functions and lattices. In *Proc. of EUROCRYPT*, volume 7237 of *LNCS*, pages 719–737. Springer, 2012.

[Bra12] Z. Brakerski. Fully homomorphic encryption without modulus switching from classical GapSVP. In *Proc. of CRYPTO*, volume 7417 of *LNCS*, pages 868–886. Springer, 2012.

[BV11a] Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *Proc. of FOCS*, pages 97–106. IEEE Computer Society Press, 2011.

[BV11b] Z. Brakerski and V. Vaikuntanathan. Fully homomorphic encryption from Ring-LWE and security for key dependent messages. In *Proc. of CRYPTO*, volume 6841 of *LNCS*, pages 505–524. Springer, 2011.

[BV14] Z. Brakerski and V. Vaikuntanathan. Lattice-based FHE as secure as PKE. In *Proc. of ITCS*, pages 1–12. ACM, 2014.

[CLO+13] A. Choudhury, J. Loftus, E. Orsini, A. Patra, and N. P. Smart. Between a rock and a hard place: Interpolating between MPC and FHE. In *Proc. of ASIACRYPT*, volume 8270 of *LNCS*, pages 221–240. Springer, 2013.

[CM15] M. Clear and C. McGoldrick. Multi-identity and multi-key leveled FHE from learning with errors. In *Proc. of CRYPTO*, volume 9216 of *LNCS*, pages 630–656. Springer, 2015.

[CS15] J. H. Cheon and D. Stehlé. Fully homomophic encryption over the integers revisited. In *Proc. of EUROCRYPT*, volume 9056 of *LNCS*, pages 513–536. Springer, 2015.

[DDLL13] L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky. Lattice signatures and bimodal Gaussians. In *Proc. of CRYPTO*, volume 8042 of *LNCS*, pages 40–56. Springer, 2013.

[DGHV10] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In *Proc. of EUROCRYPT*, volume 6110 of *LNCS*, pages 24–43. Springer, 2010.

[DM] L. Ducas and D. Micciancio. Fhew. `https://github.com/lducas/FHEW`.

[DM15] L. Ducas and D. Micciancio. FHEW: bootstrapping homomorphic encryption in less than a second. In *Proc. of EUROCRYPT*, volume 9056 of *LNCS*, pages 617–640. Springer, 2015.

[Gen09a] C. Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. Manuscript available at `http://crypto.stanford.edu/craig`.

[Gen09b] C. Gentry. Fully homomorphic encryption using ideal lattices. In *Proc. of STOC*, pages 169–178. ACM, 2009.

[Gen10] C. Gentry. Toward basing fully homomorphic encryption on worst-case hardness. In *Proc. of CRYPTO*, volume 6223 of *LNCS*, pages 116–137. Springer, 2010.

[GGH13] S. Garg, C. Gentry, and S. Halevi. Candidate multilinear maps from ideal lattices. In *EUROCRYPT*, volume 7881 of *LNCS*, pages 1–17, 2013.

[GHS12] C. Gentry, S. Halevi, and N. P. Smart. Homomorphic evaluation of the AES circuit. In *Proc. of CRYPTO*, volume 7417 of *LNCS*, pages 850–867. Springer, 2012.

[GKPV10] S. Goldwasser, Y. T. Kalai, C. Peikert, and V. Vaikuntanathan. Robustness of the learning with errors assumption. In *Proc. of Innovations in Computer Science - ICS*, pages 230–240. Tsinghua University Press, 2010.

[GSW13] C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Proc. of CRYPTO*, volume 8042 of *LNCS*, pages 75–92. Springer, 2013.

[HJ15] Y. Hu and H. Jia. Cryptanalysis of GGH map. *IACR Cryptology ePrint Archive*, 2015:301, 2015.

[HS] S. Halevi and V. Shoup. Helib. `https://github.com/shaih/HElib`.

[HS15] S. Halevi and V. Shoup. Bootstrapping for HElib. In *Proc. of EUROCRYPT*, volume 9056 of *LNCS*, pages 641–670. Springer, 2015.

[LPR10] V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In *Proc. of EUROCRYPT*, volume 6110 of *LNCS*, pages 1–23. Springer, 2010.

[LSS14] A. Langlois, D. Stehlé, and R. Steinfeld. GGHLite: More efficient multilinear maps from ideal lattices. In *Proc. of EUROCRYPT*, LNCS, pages 239–256. Springer, 2014.

[LTV12] A. López-Alt, E. Tromer, and V. Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *Proc. of STOC*, pages 1219–1234. ACM, 2012.

[Lyu09] V. Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In *Proc. of ASIACRYPT*, volume 5912 of *LNCS*, pages 598–616. Springer, 2009.

[Lyu12] V. Lyubashevsky. Lattice signatures without trapdoors. In *Proc. of EUROCRYPT*, volume 7237 of *LNCS*, pages 738–755. Springer, 2012.

[MW15] P. Mukherjee and D. Wichs. Two round MPC from LWE via multi-key FHE. *IACR Cryptology ePrint Archive*, 2015:345, 2015.

[OPP14] R. Ostrovsky, A. Paskin-Cherniavsky, and B. Paskin-Cherniavsky. Maliciously circuit-private FHE. In *Proc. of CRYPTO*, volume 8616 of *LNCS*, pages 536–553. Springer, 2014.

[OPW11] A. O'Neill, C. Peikert, and B. Waters. Bi-deniable public-key encryption. In *Proc. of CRYPTO*, volume 6841 of *LNCS*, pages 525–542. Springer, 2011.

[Reg09] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009.

[Sch87] C. P. Schnorr. A hierarchy of polynomial lattice basis reduction algorithms. *Theor. Comput. Science*, 53:201–224, 1987.

[SS10] D. Stehlé and R. Steinfeld. Faster fully homomorphic encryption. In *Proc. of ASIACRYPT*, volume 6477 of *LNCS*, pages 377–394. Springer, 2010.

[SV10] N. P. Smart and F. Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In *Proc. of PKC*, volume 6056 of *LNCS*, pages 420–443. Springer, 2010.