Multi-Key FHE from LWE, Revisited

Chris Peikert* Sina Shiehian†

February 23, 2016

Abstract

Traditional fully homomorphic encryption (FHE) schemes only allow computation on data encrypted under a single key. López-Alt, Tromer, and Vaikuntanathan (STOC 2012) proposed the notion of *multi-key* FHE, which allows homomorphic computation on ciphertexts encrypted under different keys, and also gave a construction based on a (somewhat nonstandard) assumption related to NTRU. More recently, Clear and McGoldrick (CRYPTO 2015), followed by Mukherjee and Wichs (EUROCRYPT 2016), proposed a multi-key FHE based on learning with errors (LWE). However, unlike the original construction of López-Alt *et al.*, these later LWE-based schemes have the somewhat undesirable property of being "single-hop" with respect to keys, i.e., all relevant keys must be known at the start of the homomorphic computation, and the output cannot be usefully combined with ciphertexts encrypted under other keys (unless an expensive "bootstrapping" step is performed).

In this work we construct two multi-key FHE schemes, based on LWE assumptions, which are *multi-hop with respect to keys*: the output of a homomorphic computation on ciphertexts encrypted under a set of keys can be used in further homomorphic computation involving *additional* keys, and so on. Our systems also have smaller ciphertexts than the previous LWE-based ones; indeed, ciphertexts in our second construction are simply GSW ciphertexts with no auxiliary data.

*Computer Science and Engineering, University of Michigan. Email: cpeikert@umich.edu. This material is based upon work supported by the National Science Foundation under CAREER Award CCF-1054495 and CNS-1606362, and by the Alfred P. Sloan Foundation. The views expressed are those of the authors and do not necessarily reflect the official policy or position of the National Science Foundation or the Sloan Foundation.

†Computer Science and Engineering, University of Michigan. Email: shiayan@umich.edu.

1 Introduction

Secure *multiparty computation* (MPC) is an important and well-studied problem in cryptography. In MPC, multiple users want to jointly perform a computation on their respective inputs via an interactive protocol. Informally, the goal is for the protocol to reveal nothing more than the *output* of the computation.

Fully homomorphic encryption (FHE) is a powerful tool for constructing secure MPC protocols. One approach suggested in Gentry's seminal work [Gen09], and later optimized by Asharov *et al.* [AJL+12], is to have an initial phase in which all parties run a protocol to generate a sharing of an FHE secret key, then use the public key to encrypt their inputs and publish the ciphertexts. The parties then *locally* compute an encryption of the output using homomorphic operations. Finally, they run a protocol to decrypt the encrypted output, using their secret key shares. Overall, this approach requires the set of involved parties to be known in advance, and for them to run interactive protocols both *before* and *after* their local computation.

López-Alt *et al.* [LTV12] (hereafter LTV) introduced the interesting notion of *on-the-fly MPC*, in which the set of parties who contribute inputs to the computation, and even the computation itself, need not be fixed in advance, and can even be chosen adaptively. In addition, there is no interaction among the parties at the outset: any user whose data might potentially be used simply uploads her encrypted input to a central server in advance, and can then go offline. The server then uses the uploaded data to compute (or continue computing) a desired function, and when finished, outputs an encrypted output. Finally, the parties whose inputs were used in the computation—and *only* those parties—run an interactive protocol to jointly decrypt the ciphertext.

Multi-key FHE. Traditional FHE schemes only allow computation on data encrypted under a *single* key, and therefore are not suitable for on-the-fly MPC, where users' inputs must be encrypted under different keys. As a tool for constructing on-the-fly protocols, LTV proposed a new type of FHE scheme, which they called *multi-key FHE* (MK-FHE). Such a scheme extends the FHE functionality to allow homomorphic computation on ciphertexts encrypted under different, independent keys. Decrypting the result of such a computation necessarily requires all of the corresponding secret keys.

In [LTV12], LTV constructed an MK-FHE scheme based on a variant of the NTRU cryptosystem [HPS98]. The security of this scheme was based on a new and somewhat non-standard assumption on polynomial rings, which, unlike the commonly used learning with errors (LWE) assumption [Reg05] (or its ring-based analogue [LPR10]), is not currently supported by a worst-case hardness theorem. (They also constructed MK-FHE based on ring-LWE, but limited only to a *logarithmic* number of keys and circuit depth.) Recently, Clear and McGoldrick [CM15] gave an LWE-based construction for an *unlimited* number of keys, using a variant of the FHE scheme of Gentry *et al.* [GSW13] (hereafter GSW). Later, Mukherjee and Wichs [MW15] provided another exposition of the Clear-McGoldrick scheme, and built a two-round (plain) MPC protocol upon it.

How many hops? We observe that the multi-key FHE constructed by LTV is, to extend the terminology of [GHV10], "*multi-hop* with respect to keys:" one can perform a homomorphic computation on a collection of ciphertexts encrypted under some set of keys, then use the resulting ciphertext as an input to further homomorphic computation on ciphertexts encrypted under *additional* keys, and so on. (Multi-hop homomorphic computation is supported by essentially all natural single-key FHE schemes as well.) The on-the-fly MPC protocol of [LTV12] naturally inherits this multi-hop flavor, which is very much in the spirit of "on the fly" computation, since it allows reusing encrypted results across different computations.

By contrast, it turns out that neither of the MK-FHE constructions from [CM15, MW15] appear to be

multi-hop with respect to keys, but are instead only *single-hop*: once a homomorphic computation has been performed on a collection of ciphertexts encrypted under some set of keys, the output cannot easily be used in further computation involving additional keys. Instead, one must restart the whole computation from scratch (incorporating all the relevant keys from the very beginning), or perform an expensive "bootstrapping" step (which may be even more costly). This rules out a multi-hop computation, since all involved parties must be known before the computation begins. In summary, existing constructions of MK-FHE and on-the-fly MPC from standard (worst-case) lattice assumptions still lack basic functionality that has been obtained from more heuristic assumptions.

1.1 Our Results

In this work we construct two (leveled) multi-key FHE schemes, for any number of keys, from LWE assumptions. Like the original MK-FHE scheme of [LTV12] (and unlike those of [CM15, MW15]), both of our schemes are "*multi-hop* with respect to keys," and hence are suitable for multi-hop on-the-fly MPC. That is, one can homomorphically compute on ciphertexts encrypted under several keys, then use the result in further homomorphic computation on ciphertexts encrypted under additional keys, and so on.

We now describe our two systems in more detail, and discuss their different efficiency and security tradeoffs.

Scheme #1: large ciphertexts, standard LWE. The security of our first scheme, which is described in Section 3, is based on the standard n-dimensional decision-LWE assumption (appropriately parameterized), but has rather large ciphertexts and correspondingly slow homomorphic operations. Actually, the ciphertexts are about an n factor *smaller* than fresh ciphertexts in the systems from [CM15, MW15], but unlike in those systems, our ciphertexts remain rather large even *after* multi-key homomorphic operations. Essentially, this is the price of being multi-hop w.r.t. keys—indeed, it is possible at any point to "downgrade" our ciphertexts to ordinary GSW ciphertexts, by giving up the ability to extend ciphertexts to additional keys.

Scheme #2: small ciphertexts, circular LWE. In our second scheme, which is described in Section 4, ciphertexts are simply GSW ciphertexts, and are therefore (relatively) small and admit correspondingly efficient homomorphic operations. This efficiency comes at the price of rather large *public keys* (which are comparable to fresh ciphertexts in the systems from [CM15, MW15]) and a correspondingly slow algorithm for extending ciphertexts to additional keys. This profile seems preferable to our first scheme's, because applications of MK-FHE would typically involve many more homomorphic operations than extensions to new keys. Therefore, we consider this scheme to be our main contribution.

Interestingly, the security of our second scheme appears to require a natural *circular security* assumption for LWE. Despite some positive results for circular security of LWE-based encryption [ACPS09], we do not yet see a way to prove security under standard LWE. We point out, however, that our assumption is no stronger than the circular-security assumptions that are used to "bootstrap" FHE, because any circular-secure FHE is itself fully key-dependent message secure [Gen09]. So in a context where our system is bootstrapped to obtain unbounded FHE, we actually incur no additional assumption.

1.2 Technical Overview

For context, we start with a brief overview of the prior (single-hop w.r.t. keys) MK-FHE constructions of [CM15, MW15], and the challenge in making them multi-hop. In these systems, a fresh ciphertext that decrypts under secret key $\mathbf{t} \in \mathbb{Z}^n$ is a GSW ciphertext $\mathbf{C} \in \mathbb{Z}_q^{n \times m}$ encrypted to the corresponding public

key $\mathbf{P}$, along with an encryption $\mathbf{D}$ of the *encryption randomness* used to produce $\mathbf{C}$ from $\mathbf{P}$. (Specifically, each entry of the randomness matrix is encrypted as a separate GSW ciphertext.)

To perform a homomorphic computation on fresh ciphertexts $(\mathbf{C}_i, \mathbf{D}_i)$ that are respectively encrypted under secret keys $\mathbf{t}_i$ for (say) $i = 1, 2$, we first *extend* each ciphertext to an ordinary GSW ciphertext

$$\hat{\mathbf{C}}_i = \begin{bmatrix} \mathbf{C}_i & \mathbf{X}_i \\ & \mathbf{C}_i \end{bmatrix} \in \mathbb{Z}_q^{2n \times 2m} \tag{1.1}$$

that decrypts to the same message under the *concatenated* key $(\mathbf{t}_1, \mathbf{t}_2)$, and then perform normal GSW homomorphic operations on these extended ciphertexts. Essentially, extending $\mathbf{C}_1$ is done by considering the extra "junk" term $(\mathbf{t}_2 - \mathbf{t}_1) \cdot \mathbf{C}_1$ that arises from decrypting $\mathbf{C}_1$ under the wrong secret key $\mathbf{t}_2$, and cancelling it out via a ciphertext $\mathbf{X}_1$ that "decrypts" under $\mathbf{t}_1$ to (the negation of) the same junk term. To produce $\mathbf{X}_1$ we use linearly homomorphic operations on $\mathbf{D}_1$ (the encryption of $\mathbf{C}_1$'s randomness relative to $\mathbf{P}_1$), along with some additional information about $\mathbf{t}_1$ relative to a shared public parameter.

We point out that in the above scheme, it is not clear how to obtain an encryption of $\hat{\mathbf{C}}_i$'s underlying encryption randomness—indeed, it is not even clear what composite *public key* $\hat{\mathbf{P}}$ the ciphertext $\hat{\mathbf{C}}_i$ would be relative to, nor whether valid encryption randomness for $\hat{\mathbf{C}}_i$ exists at all! (Indeed, for certain natural ways of combining the public keys $\mathbf{P}_i$, valid encryption randomness is not likely to exist.) This is what prevents the extended ciphertexts from satisfying the same invariant that fresh ciphertexts satisfy, which makes the scheme only single-hop with respect to keys. Moreover, even if we could produce an encryption of the ciphertext randomness (assuming it exists), it is not clear whether we could later re-extend an arbitrary ciphertext $\mathbf{C} \in \mathbb{Z}_q^{2n \times 2m}$ that decrypts under $(\mathbf{t}_1, \mathbf{t}_2)$ to an additional key $\mathbf{t}_3$: the block upper-triangular structure from Equation (1.1) would produce a $4n$-by-$4m$ matrix, which is too large.

1.2.1 Our Approach

To overcome the above difficulties, our ciphertexts and/or public keys consist of different information, whose invariants can be maintained after extension to additional keys. In particular, we forego maintaining *encryption randomness* relative to a *varying* public key, and instead only maintain *commitment randomness* relative to a *fixed* public parameter, along with an encryption of that randomness.[1] Concretely, this works in two different ways in our two schemes, as we now explain.

Scheme #1. In our first system, a ciphertext under a secret key $\mathbf{t} \in \mathbb{Z}^{kn}$—which would typically be the concatenation of $k \geq 1$ individual secret keys—consists of three components:

1. a (symmetric-key) *GSW ciphertext* $\mathbf{C} \in \mathbb{Z}_q^{kn \times km}$ that decrypts under $\mathbf{t}$,
2. a GSW-style *homomorphic commitment* (à la [GVW15]) $\mathbf{F} \in \mathbb{Z}_q^{n \times m}$ to the same message, relative to a public parameter, and
3. a special encryption $\mathbf{D}$ under $\mathbf{t}$ of the *commitment randomness* underlying $\mathbf{F}$.

To extend such a ciphertext to a new secret key $\mathbf{t}^* \in \mathbb{Z}^n$, we simply extend the GSW ciphertext $\mathbf{C}$ to some

$$\mathbf{C}' = \begin{bmatrix} \mathbf{C} & \mathbf{X} \\ & \mathbf{F} \end{bmatrix} \in \mathbb{Z}_q^{(k+1)n \times (k+1)m},$$

[1] We note that the previous constructions from [CM15, MW15] also require a public parameter, so we are not changing the model.

where $\mathbf{X}$ is produced from $\mathbf{D}$ (in much the same way as above) to cancel out the "junk" term that comes from "decrypting" $\mathbf{F}$ with $\mathbf{t}^*$. The commitment $\mathbf{F}$ and its encrypted randomness $\mathbf{D}$ remain unchanged, except that we need to pad $\mathbf{D}$ with zeros to make it valid under $(\mathbf{t}, \mathbf{t}^*)$.

Finally, it is not too hard to design homomorphic addition and multiplication operations for ciphertexts having the above form: as shown in [GVW15], GSW commitments admit exactly the same homomorphic operations as GSW encryption, so we can maintain a proper commitment. The homomorphic operations also have a natural, predictable effect on the underlying commitment randomness, so we can use the encrypted randomness $\mathbf{D}_i$ along with the GSW ciphertexts $\mathbf{C}_i$ to maintain correct encrypted commitment randomness.

Scheme #2. Our second system works differently from all the previous ones. In it, ciphertexts are simply GSW ciphertexts, with no extra components, so they support the standard homomorphic operations. To support extending ciphertexts to additional keys, each *public key* contains a *commitment* to its secret key $\mathbf{t}$, along with an appropriate encryption under $\mathbf{t}$ of the commitment randomness. (This cyclical relation between secret key and commitment randomness is what leads to our circular-security assumption.) We show how to combine two public keys to get a ciphertext, under the *concatenation* of their secret keys $\mathbf{t}_1, \mathbf{t}_2$, that encrypts the *tensor product* $\mathbf{t}_1 \otimes \mathbf{t}_2$ of those keys. By applying homomorphic operations, it is then fairly straightforward to extend a ciphertext that decrypts under one of the keys to a ciphertext that decrypts under their concatenation.

2 Preliminaries

In this work, vectors are denoted by lower-case bold letters (e.g., $\mathbf{a}$), and are *row* vectors unless otherwise indicated. Matrices are denoted by upper-case bold letters (e.g., $\mathbf{A}$). We define $[k] := \{1, \ldots, k\}$ for any non-negative integer k.

Approximations. As in many works in lattice cryptography, we work with "noisy equations" and must quantify the quality of the approximation. For this purpose we use the notation $\approx$ to indicate that the two sides are approximately equal up to some *additive* error, and we always include a bound on the magnitude of this error. For example,

$$x \approx y \qquad\qquad \text{(error } E\text{)}$$

means that $x = y + e$ for some $e \in [-E, E]$. In the case of vectors or matrices, the error bound applies to every entry of the error term, i.e., it is an ℓ_∞ bound.

For simplicity of analysis, in this work we use the following rather crude "expansion" bounds to quantify error growth. (Sharper bounds can be obtained using more sophisticated tools like subgaussian random variables.) Because $\|\mathbf{x} \cdot \mathbf{y}^t\|_\infty \le \|\mathbf{x}\|_\infty \cdot \|\mathbf{y}\|_1$ and $\|\mathbf{y}\|_1 \le \dim(\mathbf{y}) \cdot \|\mathbf{y}\|_\infty$, we have implications like

$$\begin{aligned} &\mathbf{X} \approx \mathbf{Y} && \text{(error } E\text{)} \\ \implies &\mathbf{X} \cdot \mathbf{R} \approx \mathbf{Y} \cdot \mathbf{R}. && \text{(error height}(\mathbf{R}) \cdot \|\mathbf{R}\|_\infty \cdot E\text{)} \end{aligned}$$

for any $\mathbf{X}, \mathbf{Y}, \mathbf{R}$.

Tensor products. The *tensor* (or *Kronecker*) product $\mathbf{A} \otimes \mathbf{B}$ of an m_1-by-n_1 matrix $\mathbf{A}$ with an m_2-by-n_2 matrix $\mathbf{B}$, both over a common ring $\mathcal{R}$, is the $m_1 m_2$-by-$n_1 n_2$ matrix consisting of m_2-by-n_2 blocks, whose (i,j)th block is $a_{i,j} \cdot \mathbf{B}$, where $a_{i,j}$ denotes the (i,j)th entry of $\mathbf{A}$.

It is clear that

$$r(\mathbf{A} \otimes \mathbf{B}) = (r\mathbf{A}) \otimes \mathbf{B} = \mathbf{A} \otimes (r\mathbf{B})$$

for any scalar $r \in \mathcal{R}$. We extensively use the *mixed-product property* of tensor products, which says that

$$(\mathbf{A} \otimes \mathbf{B}) \cdot (\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD})$$

for any matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ of compatible dimensions. In particular,

$$(\mathbf{A} \otimes \mathbf{B}) = (\mathbf{A} \otimes \mathbf{I}_{\text{height}(\mathbf{B})}) \cdot (\mathbf{I}_{\text{width}(\mathbf{A})} \otimes \mathbf{B}) = (\mathbf{I}_{\text{height}}(\mathbf{A}) \otimes \mathbf{B}) \cdot (\mathbf{A} \otimes \mathbf{I}_{\text{width}}(\mathbf{B})).$$

2.1 Cryptographic Definitions

Definition 2.1. A leveled multi-hop, multi-key FHE scheme is a tuple of efficient randomized algorithms $(\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{EvalNAND})$ having the following properties:

- $\mathsf{Setup}(1^\lambda, 1^k, 1^d)$, given the security parameter λ, a bound k on the number of keys, and a bound d on the circuit depth, outputs a public parameter pp. (All the following algorithms implicitly take pp as an input.)
- $\mathsf{Gen}()$ outputs a public key pk and secret key sk.
- $\mathsf{Enc}(pk, \mu)$, given a public key pk and a message $\mu \in \{0,1\}$, outputs a ciphertext c. For convenience, we assume that c implicitly contains a reference to pk.
- $\mathsf{Dec}((sk_1, sk_2, \ldots, sk_t), c)$, given a tuple of secret keys $sk_1, \ldots, sk_t$ and a ciphertext c, outputs a bit.
- $\mathsf{EvalNAND}(c_1, c_2)$, given two ciphertexts c_1, c_2, outputs a ciphertext $\hat{c}$. For convenience, we assume that $\hat{c}$ implicitly contains a reference to each public key associated with either c_1 or c_2 (or both).

These algorithms should satisfy *correctness* and *compactness* functionality properties, as defined below.

We now describe how to homomorphically evaluate a given boolean circuit composed of NAND gates and having one output wire, which is without loss of generality. The algorithm $\mathsf{Eval}(C, (c_1, \ldots, c_N))$, given a circuit C having N input wires, first associates c_i with the ith input wire for each $i = 1, \ldots, N$. Then for each gate (in some topological order) having input wires i, j and output wire k, it computes $c_k \leftarrow \mathsf{EvalNAND}(c_i, c_j)$. Finally, it outputs the ciphertext associated with the output wire.

We stress that the above homomorphic evaluation process is qualitatively different from the ones defined in [LTV12, MW15], because when homomorphically evaluating each gate we can only use the key(s) associated with the input ciphertexts *for that gate alone*; this is what makes the computation multi-hop. By contrast, homomorphic evaluation in [LTV12, MW15] is given all the input ciphertexts and public keys from the start, so it can (and does, in the case of [MW15]) use this knowledge before evaluating any gates.

Definition 2.2 (Correctness). A leveled multi-hop, multi-key FHE scheme is *correct* if for all positive integers λ, k, d, for every circuit C of depth at most d having N input wires, for every function $\pi \colon [N] \to [k]$ (which associates each input wire with a key pair), and for every $x \in \{0,1\}^N$, the following experiment succeeds with $1 - \mathrm{negl}(\lambda)$ probability: generate a public parameter $pp \leftarrow \mathsf{Setup}(1^\lambda, 1^k, 1^d)$, generate key

pairs $(pk_j, sk_j) \leftarrow \mathsf{Gen}()$ for each $j \in [k]$, generate ciphertexts $c_i \leftarrow \mathsf{Enc}(pk_{\pi(i)}, x_i)$ for each $i \in [N]$, let $\hat{c} \leftarrow \mathsf{Eval}(C, (c_1, \ldots, c_N))$, and finally test whether

$$\mathsf{Dec}((sk_j), \hat{c}) = C(x_1, \ldots, x_N),$$

where Dec is given those secret keys sk_j corresponding to the public keys referenced by $\hat{c}$.

Definition 2.3 (Compactness). A leveled multi-hop, multi-key FHE scheme is *compact* if there exists a polynomial $p(\cdot, \cdot, \cdot)$ such that in the experiment from Definition 2.2, $|\hat{c}| \leq p(\lambda, k, d)$. In other words, the length of $\hat{c}$ is independent of C and N, but can depend polynomially on λ, k, and d.

2.2 Learning With Errors

For a positive integer dimension n and modulus q, and an error distribution χ over $\mathbb{Z}$, the LWE distribution and decision problem are defined as follows. For an $\mathbf{s} \in \mathbb{Z}^n$, the LWE distribution $A_{\mathbf{s},\chi}$ is sampled by choosing a uniformly random $\mathbf{a} \leftarrow \mathbb{Z}_q^n$ and an error term $e \leftarrow \chi$, and outputting $(\mathbf{a}, b = \langle \mathbf{s}, \mathbf{a} \rangle + e) \in \mathbb{Z}_q^{n+1}$.

Definition 2.4. The decision-LWE$_{n,q,\chi}$ problem is to distinguish, with non-negligible advantage, between any desired (but polynomially bounded) number of independent samples drawn from $A_{\mathbf{s},\chi}$ for a single $\mathbf{s} \leftarrow \chi^n$, and the same number of *uniformly random* and independent samples over $\mathbb{Z}_q^{n+1}$.[2]

A standard instantiation of LWE is to let χ be a *discrete Gaussian* distribution (over $\mathbb{Z}$) with parameter $r = 2\sqrt{n}$. A sample drawn from this distribution has magnitude bounded by, say, $r\sqrt{n} = \Theta(n)$ except with probability at most 2^{-n}. For this parameterization, it is known that LWE is at least as hard as *quantumly* approximating certain "short vector" problems on n-dimensional lattices, in the worst case, to within $\tilde{O}(q\sqrt{n})$ factors [Reg05]. Classical reductions are also known for different parameterizations [Pei09, BLP+13].

In this work it will be convenient to use a form of LWE that is somewhat syntactically different from, but computationally equivalent to, the one defined above. Letting $\mathbf{s} = (-\bar{\mathbf{s}}, 1) \in \mathbb{Z}^n$ where $\bar{\mathbf{s}} \leftarrow \chi^{n-1}$, notice that an LWE sample $\mathbf{b} = (\mathbf{a}, b = \langle \mathbf{s}, \mathbf{a} \rangle + e) \in \mathbb{Z}_q^n$ drawn from $A_{\bar{\mathbf{s}},\chi}$ is simply a uniformly random vector satisfying

$$\langle \mathbf{s}, \mathbf{b} \rangle = \mathbf{s} \cdot \mathbf{b}^t = e \approx 0. \tag{2.1}$$

Therefore, decision-LWE$_{n-1,q,\chi}$ is equivalent to the problem of distinguishing samples having the above form (and in particular, satisfying Equation (2.1)) from uniformly random ones.

More generally, for $\mathbf{s} \in \mathbb{Z}^n$ as above and some $t = \mathrm{poly}(n)$, we will need to generate uniformly random vectors $\mathbf{b} \in \mathbb{Z}_q^{tn}$ that satisfy

$$(\mathbf{I}_t \otimes \mathbf{s}) \cdot \mathbf{b} = \mathbf{e} \approx \mathbf{0},$$

for some $\mathbf{e} \leftarrow \chi^t$. This is easily done by concatenating t independent samples from $A_{\bar{\mathbf{s}},\chi}$; clearly, the result is indistinguishable from uniform assuming the hardness of decision-LWE$_{n,q,\chi}$.

2.3 Gadgets and Decomposition

Here we recall the notion of a "gadget" [MP12], which is used for decomposing $\mathbb{Z}_q$-elements—or more generally, vectors or matrices over $\mathbb{Z}_q$—into short vectors or matrices over $\mathbb{Z}$. We also define some new notation that will be convenient for our application.

[2] Notice that in the above definition, the coordinates of $\mathbf{s}$ are drawn from the error distribution χ; as shown in [ACPS09], this form of the problem is equivalent to the one where $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ is drawn uniformly at random.

For simplicity, throughout this work we use the standard "powers of two" gadget vector

$$\mathbf{g} = (1, 2, 4, 8, \ldots, 2^{\ell-1}) \in \mathbb{Z}_q^\ell, \quad \text{where } \ell = \lceil \lg q \rceil.$$

The "bit decomposition" function $\mathbf{g}^{-1} \colon \mathbb{Z}_q \to \{0,1\}^\ell$ outputs a binary *column* vector (over $\mathbb{Z}$) consisting of the binary representation of (the canonical representative in $\{0, 1, \ldots, q-1\}$ of) its argument. As such, it satisfies the identity $\mathbf{g} \cdot \mathbf{g}^{-1}[a] = a$. (This identity explains the choice of notation $\mathbf{g}^{-1}$; we stress that $\mathbf{g}^{-1}$ is a *function*, not a vector itself.) Symmetrically, we define the notation

$$[a]\mathbf{g}^{-t} := \mathbf{g}^{-1}[a]^t,$$

which outputs a binary *row* vector and satisfies the identity $[a]\mathbf{g}^{-t} \cdot \mathbf{g}^t = a$. (This identity explains why we place the bracketed argument to the *left* of $\mathbf{g}^{-t}$.)

More generally, we define the operation denoted by $(\mathbf{I}_n \otimes \mathbf{g}^{-1})[\cdot]$, which applies $\mathbf{g}^{-1}$ entrywise to a height-n vector/matrix, and thereby produces a height-$n\ell$ binary output that satisfies the convenient identity

$$(\mathbf{I}_n \otimes \mathbf{g}) \cdot (\mathbf{I}_n \otimes \mathbf{g}^{-1})[\mathbf{A}] = \mathbf{A}.$$

Similarly, we define $[\cdot](\mathbf{I}_n \otimes \mathbf{g}^{-t})$ to apply $\mathbf{g}^{-t}$ entrywise to a width-n vector/matrix, thereby producing a width-$n\ell$ output that satisfies

$$[\mathbf{A}](\mathbf{I}_n \otimes \mathbf{g}^{-t}) \cdot (\mathbf{I}_n \otimes \mathbf{g}^t) = \mathbf{A}.$$

For the reader who is familiar with previous works that use gadget techniques, the matrix $\mathbf{I}_n \otimes \mathbf{g}$ is exactly the n-row gadget matrix $\mathbf{G}$, and $(\mathbf{I}_n \otimes \mathbf{g}^{-1})[\cdot]$ is exactly the bit-decomposition operation $\mathbf{G}^{-1}$ on height-n vectors/matrices. In this work we adopt the present notation because we use several different dimensions n, and because it interacts cleanly with tensor products of vectors and matrices, which we use extensively in what follows.

3 Large-Ciphertext Construction

In this section we describe our first construction of a multi-hop, multi-key FHE, which has small keys but rather large ciphertexts (although fresh ciphertexts are still smaller than in prior constructions). For simplicity, we describe the scheme in the symmetric-key setting, but then note how to obtain a public-key scheme using a standard transformation.

The system is parameterized by a dimension n, modulus q, and error distribution χ for the underlying LWE problem; we also let $m = \lceil 2n \log q \rceil$. For concreteness, we let χ be the standard discrete Gaussian error distribution with parameter $2\sqrt{n}$; to recall, the samples it produces have magnitudes bounded by some $E = \Theta(n)$ except with exponentially small $2^{-\Omega(n)}$ probability. The modulus q is instantiated in Section 3.3 below, based on a desired depth of homomorphic computation and number of distinct keys. The scheme is defined as follows.

- Setup: output a uniformly random $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$.
- $\mathsf{Gen}(\mathbf{A})$: choose $\bar{\mathbf{t}} \leftarrow \chi^{n-1}$ and define $\mathbf{t} := (-\bar{\mathbf{t}}, 1) \in \mathbb{Z}^n$. Choose $\mathbf{e} \leftarrow \chi^m$ and define

$$\begin{aligned} \mathbf{b} &:= \mathbf{t}\mathbf{A} + \mathbf{e} \\ &\approx \mathbf{t}\mathbf{A} \in \mathbb{Z}_q^m. \qquad \text{(error } E\text{)} \end{aligned} \tag{3.1}$$

  Output $\mathbf{t}$ as the secret key and $\mathbf{b}$ as the associated public extension key.

- $\mathsf{Enc}(\mathbf{t}, \mu \in \{0,1\})$: do the following, outputting $(\mathbf{C}, \mathbf{F}, \mathbf{D})$ as the ciphertext.

  1. As described in Section 2.2, choose an LWE matrix $\bar{\mathbf{C}} \in \mathbb{Z}_q^{n \times n\ell}$ that satisfies $\mathbf{t}\bar{\mathbf{C}} \approx \mathbf{0}$, and define

  $$\mathbf{C} := \bar{\mathbf{C}} + \mu(\mathbf{I}_n \otimes \mathbf{g}) \in \mathbb{Z}_q^{n \times n\ell}.$$

  Notice that $\mathbf{C}$ is simply a *GSW ciphertext* encrypting μ under secret key $\mathbf{t}$:

  $$\mathbf{tC} = \mathbf{t}\bar{\mathbf{C}} + \mu(\mathbf{t} \otimes 1) \cdot (\mathbf{I}_n \otimes \mathbf{g}) \approx \mu(\mathbf{t} \otimes \mathbf{g}). \qquad \text{(error } E_{\mathbf{C}}\text{)} \tag{3.2}$$

  2. In addition, choose a uniformly random $\mathbf{R} \in \{0,1\}^{m \times n\ell}$ and define

  $$\mathbf{F} := \mathbf{AR} + \mu(\mathbf{I}_n \otimes \mathbf{g}) \in \mathbb{Z}_q^{n \times n\ell}. \tag{3.3}$$

  We view $\mathbf{F}$ as a *commitment* to the message μ under randomness $\mathbf{R}$.

  3. Finally, choose (as described in Section 2.2) an LWE matrix $\bar{\mathbf{D}} \in \mathbb{Z}_q^{nm\ell \times n\ell}$ that satisfies

  $$(\mathbf{I}_{m\ell} \otimes \mathbf{t}) \cdot \bar{\mathbf{D}} \approx \mathbf{0},$$

  and define $\mathbf{D} := \bar{\mathbf{D}} + (\mathbf{R} \otimes \mathbf{g}^t \otimes \mathbf{e}_n^t)$, where $\mathbf{e}_n \in \mathbb{Z}^n$ is the nth standard basis vector (so $\mathbf{t} \cdot \mathbf{e}^t = 1$). We therefore have

  $$(\mathbf{I}_{m\ell} \otimes \mathbf{t}) \cdot \mathbf{D} \approx \mathbf{R} \otimes \mathbf{g}^t. \qquad \text{(error } E_{\mathbf{D}}\text{)} \tag{3.4}$$

  We view $\mathbf{D}$ as a kind of *encryption* of the commitment randomness $\mathbf{R}$.

- $\mathsf{Dec}(\mathbf{t}, (\mathbf{C}, \mathbf{F}, \mathbf{D}))$: this is standard GSW decryption of $\mathbf{C}$ under $\mathbf{t}$, which works due to Equation (3.2).

Remark 3.1. The above scheme is defined in the symmetric-key setting, i.e., Enc uses the secret key $\mathbf{t}$ to generate LWE samples. We can obtain a public-key scheme using a standard technique, namely, have the encryption algorithm rerandomize some public LWE samples to generate as many additional samples as needed. More formally, we define $\mathbf{B} := \mathbf{A} - \mathbf{e}_n^t \otimes \mathbf{b}$. Then because $\mathbf{t} \cdot \mathbf{e}_n^t = 1$, we have

$$\mathbf{tB} \approx \mathbf{0}. \qquad \text{(error } E\text{)}$$

The public-key encryption algorithm then constructs $\bar{\mathbf{C}}, \bar{\mathbf{D}}$ by generating fresh samples as $\mathbf{B} \cdot \mathbf{x}$ for fresh uniformly random $\mathbf{x} \in \{0,1\}^m$. It is easy to verify that $\mathbf{t}(\mathbf{Bx}) \approx 0$ with error $m \cdot E$. Security follows from a standard argument, using the LWE assumption to make $\mathbf{b}$ (and thereby $\mathbf{B}$) uniformly random, and then the leftover hash lemma to argue that the distribution of the fresh samples is negligibly far from uniform.

Theorem 3.2. *The above scheme is IND-CPA secure assuming the hardness of the decision-LWE$_{n-1,q,\chi}$ problem.*

Proof. We prove that the view of an attacker in the real game is indistinguishable from its view in a game in which the public extension key and every ciphertext are uniformly random and independent of the message; this clearly suffices for IND-CPA security. We proceed by a considering the following sequence of hybrid experiments:

Game 0: This is the real IND-CPA game.

Game 1: In this game the public extension key and the $\mathbf{C}, \mathbf{D}$ components of every ciphertext are uniformly random and independent (but $\mathbf{F}$ is constructed in the same way). More precisely:

1. Choose uniformly random public parameter $\mathbf{A}$ and extension key $\mathbf{b}$, and give them to the adversary.
2. For each encryption query, choose uniformly random and independent $\mathbf{C} \in \mathbb{Z}_q^{n \times n\ell}$ and $\mathbf{D} \in \mathbb{Z}_q^{nm\ell \times n\ell}$, construct $\mathbf{F}$ exactly as in Enc, and give ciphertext $(\mathbf{C}, \mathbf{F}, \mathbf{D})$ to the adversary.

Game 2: This is the ideal game; the only change from the previous game is that each $\mathbf{F}$ is chosen uniformly at random.

We claim that Games 0 and 1 are computationally indistinguishable under the LWE hypothesis. To prove this we describe a simulator $\mathcal{S}$ that is given an unbounded source of samples; when they are LWE samples it simulates Game 0, and when they are uniformly random samples it simulates Game 1. It works as follows:

- Draw m samples and form a matrix $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n \times m}$ with the samples as its columns. Choose a uniformly random extension key $\mathbf{b} \in \mathbb{Z}_q^m$, and let the public parameter $\mathbf{A} = \bar{\mathbf{A}} + \mathbf{e}_n^t \cdot \mathbf{b}$.
- On encryption query μ, draw samples to construct matrices $\bar{\mathbf{C}}$ and $\bar{\mathbf{D}}$, and define $\mathbf{C}, \mathbf{D}$ from these as in Enc. Also construct $\mathbf{F}$ exactly as in Enc.

If the simulator's input distribution is $A_{\bar{\mathbf{t}},\chi}$ for some $\bar{\mathbf{t}} \leftarrow \chi^{n-1}$, then the first $n-1$ rows of $\bar{\mathbf{A}}$ are uniformly random, hence $\mathbf{A}$ is uniformly random by construction. Moreover, $\mathbf{b} \approx (-\bar{\mathbf{t}}, 1) \cdot \mathbf{A}$ has the same distribution as in the real game. Finally, $\bar{\mathbf{C}}$ and $\bar{\mathbf{D}}$ are constructed exactly as in the real game, so $\mathcal{S}$ perfectly simulates Game 0.

By contrast, if the simulator's input distribution is uniform, then $\mathbf{A}$ and $\mathbf{b}$ are uniformly random and independent. Similarly, because $\bar{\mathbf{C}}$ and $\bar{\mathbf{D}}$ are uniform and independent of everything else, so are $\mathbf{C}$ and $\mathbf{D}$. Therefore, $\mathcal{S}$ perfectly simulates Game 1. This proves the first claim.

Finally, we claim that Games 1 and 2 are statistically indistinguishable. This follows directly from the leftover hash lemma. This concludes the proof. □

3.1 Extending Ciphertexts

We first describe how to *extend* a ciphertext to an additional secret key $\mathbf{t}^*$, using the associated public extension key $\mathbf{b}^* \approx \mathbf{t}^*\mathbf{A} \in \mathbb{Z}_q^m$. More precisely, suppose we have a ciphertext that encrypts μ under secret key $\mathbf{t} \in \mathbb{Z}^{n'}$. (Here the dimension n' can be arbitrary, but typically $n' = nk$ for some positive integer k, and $\mathbf{t}$ is the concatenation of k individual secret keys, each of dimension n.) The ciphertext therefore consists of component matrices

$$\mathbf{C} \in \mathbb{Z}_q^{n' \times n'\ell}, \quad \mathbf{F} \in \mathbb{Z}_q^{n \times n\ell}, \quad \mathbf{D} \in \mathbb{Z}_q^{n'm\ell \times n\ell}$$

that satisfy Equations (3.2), (3.3), and (3.4) for some short commitment randomness $\mathbf{R} \in \mathbb{Z}^{m \times n\ell}$. (Notice that the dimensions of $\mathbf{F}$ and the width of $\mathbf{D}$ do not depend on n'.)

Our goal is to extend $(\mathbf{C}, \mathbf{F}, \mathbf{D})$ to a new ciphertext $(\mathbf{C}', \mathbf{F}', \mathbf{D}')$ that satisfies Equations (3.2), (3.3), and (3.4) with respect to the concatenated secret key $\mathbf{t}' = (\mathbf{t}, \mathbf{t}^*) \in \mathbb{Z}^{n'+n}$ and some short commitment randomness $\mathbf{R}'$. We do so as follows.

- The commitment and its randomness are unchanged: we define $\mathbf{F}' := \mathbf{F}$ and $\mathbf{R}' := \mathbf{R}$. This clearly preserves Equation (3.3).

- Similarly, the encrypted randomness also is essentially unchanged, up to some padding by zeros: we define
$$\mathbf{D}' := (\mathbf{I}_{m\ell} \otimes \begin{pmatrix} \mathbf{I}_{n'} \\ \mathbf{0}_{n \times n'} \end{pmatrix}) \cdot \mathbf{D} \in \mathbb{Z}_q^{(n'+n)m\ell \times n\ell}.$$
Then Equation (3.4) is preserved: $(\mathbf{I}_{m\ell} \otimes \mathbf{t}') \cdot \mathbf{D}' = (\mathbf{I}_{m\ell} \otimes \mathbf{t}) \cdot \mathbf{D} \approx \mathbf{R} \otimes \mathbf{g}^t = \mathbf{R}' \otimes \mathbf{g}^t$.
- Lastly, we define
$$\mathbf{C}' := \begin{pmatrix} \mathbf{C} & \mathbf{X} \\ & \mathbf{F} \end{pmatrix} \in \mathbb{Z}_q^{(n'+n) \times (n'+n)\ell}$$
where $\mathbf{X}$ is defined as follows:
$$\begin{aligned} \mathbf{s} &:= [-\mathbf{b}^*](\mathbf{I}_m \otimes \mathbf{g}^{-t}) \in \{0,1\}^{m\ell}, \\ \mathbf{X} &:= (\mathbf{s} \otimes \mathbf{I}_{n'}) \cdot \mathbf{D} \in \mathbb{Z}_q^{n' \times n\ell}. \end{aligned} \tag{3.5}$$

We now do the error analysis for ciphertext extension. Notice that by construction,

$$\begin{aligned} \mathbf{t}\mathbf{X} &= (1 \otimes \mathbf{t}) \cdot (\mathbf{s} \otimes \mathbf{I}_{n'}) \cdot \mathbf{D} \\ &= (\mathbf{s} \otimes 1) \cdot (\mathbf{I}_{m\ell} \otimes \mathbf{t}) \cdot \mathbf{D} \\ &\approx \mathbf{s} \cdot (\mathbf{R} \otimes \mathbf{g}^t) && \text{(Equation (3.4), error } m\ell \cdot E_{\mathbf{D}}\text{)} \\ &= -\mathbf{b}^*\mathbf{R}. && \text{(Equation (3.5))} \end{aligned}$$

Putting everything together, we see that Equation (3.2) is preserved:

$$\begin{aligned} \mathbf{t}'\mathbf{C}' &\approx \begin{pmatrix} \mu(\mathbf{t} \otimes \mathbf{g}) & \mathbf{t}\mathbf{X} + \mathbf{t}^*\mathbf{F} \end{pmatrix} && \text{(Equation (3.2); error } E_{\mathbf{C}}\text{)} \\ &= \begin{pmatrix} \mu(\mathbf{t} \otimes \mathbf{g}) & \mathbf{t}\mathbf{X} + \mathbf{t}^*\mathbf{A}\mathbf{R} + \mu(\mathbf{t}^* \otimes \mathbf{g}) \end{pmatrix} && \text{(Equation (3.3))} \\ &\approx \begin{pmatrix} \mu(\mathbf{t} \otimes \mathbf{g}) & \mathbf{t}\mathbf{X} + \mathbf{b}^*\mathbf{R} + \mu(\mathbf{t}^* \otimes \mathbf{g}) \end{pmatrix} && \text{(Equation (3.1); error } m\|\mathbf{R}\|_\infty \cdot E\text{)} \\ &\approx \mu(\mathbf{t}' \otimes \mathbf{g}). && \text{(error } m\ell \cdot E_{\mathbf{D}}\text{)} \end{aligned}$$

In total, the error in the new ciphertext $\mathbf{C}'$ is

$$E_{\mathbf{C}'} = E_{\mathbf{C}} + m\|\mathbf{R}\|_\infty \cdot E + m\ell \cdot E_{\mathbf{D}}.$$

3.2 Homomorphic Operations

We now describe homomorphic addition and multiplication for the above cryptosystem. Suppose we have two ciphertexts $(\mathbf{C}_1, \mathbf{F}_1, \mathbf{D}_1)$ and $(\mathbf{C}_2, \mathbf{F}_2, \mathbf{D}_2)$ that respectively encrypt μ_1 and μ_2, with commitment randomness $\mathbf{R}_1$ and $\mathbf{R}_2$, under a common secret key $\mathbf{t} \in \mathbb{Z}^{n'}$. (As in the previous subsection, everything below works for arbitrary dimension n' and key $\mathbf{t}$, but typically $n' = nk$ for some positive integer k, and $\mathbf{t}$ is the concatenation of k individual secret keys.) Recall that the ciphertext components

$$\mathbf{C}_i \in \mathbb{Z}_q^{n' \times n'\ell}, \quad \mathbf{F}_i \in \mathbb{Z}_q^{n \times n\ell}, \quad \mathbf{D}_i \in \mathbb{Z}_q^{n'm\ell \times n\ell}$$

satisfy Equations (3.2), (3.3), and (3.4) for some short commitment randomness $\mathbf{R}_i \in \mathbb{Z}^{m \times n\ell}$.

- **Negation and scalar addition.** (These are used to homomorphically compute $\text{NAND}(\mu_1, \mu_2) = 1 - \mu_1\mu_2$ for $\mu_i \in \{0,1\}$.) To homomorphically negate a message for a ciphertext $(\mathbf{C}, \mathbf{F}, \mathbf{D})$, just negate each of the components. It is clear that this has the desired effect, and that the associated commitment randomness and error terms are also negated. To homomorphically add a constant $c \in \mathbb{Z}$ to a message, just add $c(\mathbf{I}_{n'} \otimes \mathbf{g})$ to both $\mathbf{C}$ and $\mathbf{F}$. It is clear that this has the desired effect, and leaves the commitment randomness and error terms unchanged.

- **Addition.** To homomorphically add, we simply add the corresponding matrices, outputting

$$(\mathbf{C}_{\text{add}}, \mathbf{F}_{\text{add}}, \mathbf{D}_{\text{add}}) := (\mathbf{C}_1 + \mathbf{C}_2, \mathbf{F}_1 + \mathbf{F}_2, \mathbf{D}_1 + \mathbf{D}_2).$$

  It is easy to verify that Equations (3.2), (3.3), and (3.4) hold for the new ciphertext with message $\mu_{\text{add}} = \mu_1 + \mu_2$ and commitment randomness $\mathbf{R}_{\text{add}} = \mathbf{R}_1 + \mathbf{R}_2$, where the errors in the approximations are also added.

- **Multiplication.** To homomorphically multiply, we define the short matrices

$$\mathbf{S}_c := (\mathbf{I}_{n'} \otimes \mathbf{g}^{-1})[\mathbf{C}_2] \in \{0,1\}^{n'\ell \times n'\ell}, \tag{3.6}$$

$$\mathbf{S}_f := (\mathbf{I}_n \otimes \mathbf{g}^{-1})[\mathbf{F}_2] \in \{0,1\}^{n\ell \times n\ell}, \tag{3.7}$$

$$\mathbf{S}_d := (\mathbf{I}_{n'm\ell} \otimes \mathbf{g}^{-1})[\mathbf{D}_2] \in \{0,1\}^{n'm\ell^2 \times n\ell}, \tag{3.8}$$

  and output the ciphertext consisting of

$$\begin{aligned}
\mathbf{C}_{\text{mul}} &:= \mathbf{C}_1 \cdot \mathbf{S}_c \\
\mathbf{F}_{\text{mul}} &:= \mathbf{F}_1 \cdot \mathbf{S}_f \\
\mathbf{D}_{\text{mul}} &:= \mathbf{D}_1 \cdot \mathbf{S}_f + (\mathbf{I}_{m\ell} \otimes \mathbf{C}_1) \cdot \mathbf{S}_d.
\end{aligned}$$

  The associated commitment randomness is defined as

$$\mathbf{R}_{\text{mul}} := \mathbf{R}_1 \cdot \mathbf{S}_f + \mu_1 \mathbf{R}_2.$$

We now show that the ciphertext output by homomorphic multiplication satisfies Equations (3.2), (3.3), and (3.4) for key $\mathbf{t}$, message $\mu_{\text{mul}} = \mu_1\mu_2$, and commitment randomness $\mathbf{R}_{\text{mul}}$. We already know that Equation (3.2), the GSW ciphertext relation, is satisfied by construction of $\mathbf{C}_{\text{mul}}$ as the homomorphic product of GSW ciphertexts $\mathbf{C}_1, \mathbf{C}_2$. Specifically:

$$\begin{aligned}
\mathbf{t}\mathbf{C}_{\text{mult}} &= \mathbf{t}\mathbf{C}_1 \cdot \mathbf{S}_c \\
&\approx \mu_1(\mathbf{t} \otimes \mathbf{g}) \cdot \mathbf{S}_c && \text{(error } n'\ell \cdot E_{\mathbf{C}_1}\text{)} \\
&= \mu_1 \mathbf{t}\mathbf{C}_2 && \text{(Equation (3.6))} \\
&\approx \mu_1\mu_2(\mathbf{t} \otimes \mathbf{g}). && \text{(error } \mu_1 E_{\mathbf{C}_2}\text{)}
\end{aligned}$$

Similarly, Equation (3.3) is satisfied by construction of $\mathbf{F}_{\text{mul}}$ as the homomorphic product of commitments $\mathbf{F}_1, \mathbf{F}_2$:

$$\begin{aligned}
\mathbf{F}_{\text{mul}} &= \mathbf{F}_1 \cdot \mathbf{S}_f \\
&= (\mathbf{A}\mathbf{R}_1 + \mu_1(\mathbf{I}_n \otimes \mathbf{g})) \cdot \mathbf{S}_f \\
&= \mathbf{A}\mathbf{R}_1 \cdot \mathbf{S}_f + \mu_1 \mathbf{F}_2 && \text{(Equation (3.7))} \\
&= \mathbf{A}\mathbf{R}_1 \cdot \mathbf{S}_f + \mu_1 \mathbf{A}\mathbf{R}_2 + \mu_1\mu_2(\mathbf{I}_n \otimes \mathbf{g}) \\
&= \mathbf{A}\mathbf{R}_{\text{mult}} + \mu_1\mu_2(\mathbf{I}_n \otimes \mathbf{g}).
\end{aligned}$$

Finally, to see that Equation (3.4) holds for $\mathbf{D}_{\text{mul}}$, first notice that

$$\begin{aligned}(\mathbf{I}_{m\ell} \otimes \mathbf{t}) \cdot \mathbf{D}_1 \cdot \mathbf{S}_f &\approx (\mathbf{R}_1 \otimes \mathbf{g}^t) \cdot (\mathbf{S}_f \otimes 1) && \text{(Equations (3.4); error } n\ell \cdot E_{\mathbf{D}_1}\text{)} \\ &= (\mathbf{R}_1 \cdot \mathbf{S}_f) \otimes \mathbf{g}^t. && (3.9)\end{aligned}$$

In addition,

$$\begin{aligned}(\mathbf{I}_{m\ell} \otimes \mathbf{t}) \cdot (\mathbf{I}_{m\ell} \otimes \mathbf{C}_1) \cdot \mathbf{S}_d &= (\mathbf{I}_{m\ell} \otimes \mathbf{t}\mathbf{C}_1) \cdot \mathbf{S}_d \\ &\approx \mu_1 (\mathbf{I}_{m\ell} \otimes \mathbf{t} \otimes \mathbf{g}) \cdot \mathbf{S}_d && \text{(Equation (3.2); error } n'\ell \cdot E_{\mathbf{C}_1}\text{)} \\ &= \mu_1 (\mathbf{I}_{m\ell} \otimes \mathbf{t}) \cdot \mathbf{D}_2 && \text{(Equation (3.8))} \\ &\approx (\mu_1 \mathbf{R}_2) \otimes \mathbf{g}^t && \text{(Equation (3.4); error } \mu_1 \cdot E_{\mathbf{D}_2}\text{)} \quad (3.10)\end{aligned}$$

Summing Equations (3.9) and (3.10) yields

$$(\mathbf{I}_{m\ell} \otimes \mathbf{t}) \cdot \mathbf{D}_{\text{mul}} \approx \mathbf{R}_{\text{mul}} \otimes \mathbf{g}^t$$

with error $n\ell \cdot E_{\mathbf{D}_1} + n'\ell \cdot E_{\mathbf{C}_1} + \mu_1 \cdot E_{\mathbf{D}_2}$ as desired.

3.3 Instantiating the Parameters

We now bound the worst-case error growth when homomorphically evaluating a depth-d circuit of NAND gates for up to k individual keys. As above, let $n' = nk$. For a ciphertext $(\mathbf{C}, \mathbf{F}, \mathbf{D})$ with commitment randomness $\mathbf{R}$, define the "max error"

$$E^* := \max(E_{\mathbf{C}}, E_{\mathbf{D}}, E \cdot \|\mathbf{R}\|_\infty).$$

By the bounds from the previous subsection, for two ciphertexts with max error at most E^*, their homomorphic NAND has max error at most $(n(k+1)\ell + 1) \cdot E^* = \text{poly}(n, k, \ell) \cdot E^*$. Similarly, when we extend a ciphertext with max error at most E^*, the result has max error at most $(m(\ell+1)+1) \cdot E^* = \text{poly}(n, \ell) \cdot E^*$. Therefore, for any depth-d homomorphic computation on fresh ciphertexts encrypted under k keys, the result has max error at most

$$\text{poly}(n, k, \ell)^{k+d}.$$

The GSW decryption algorithm works correctly on a ciphertext as long as its error is smaller than $q/4$, hence it suffices to choose a modulus q that exceeds the above quantity by a factor of four. Recalling that $\ell = \Theta(\log q) = \tilde{O}(k+d)$, this corresponds to a worst-case approximation factor of $\text{poly}(n, k, d)^{k+d}$ for n-dimensional lattice problems.

4 Small-Ciphertext Construction

In this section we describe a multi-hop, multi-key FHE having smaller ciphertexts and more efficient homomorphic operations than the one in Section 3. Indeed, ciphertexts in this system are simply GSW ciphertexts (with no additional information), which admit the usual homomorphic operations. These efficiency improvements come at the cost of larger public extension keys, as well as a circular-security assumption.

Recall that in the scheme from the previous section, a ciphertext includes a commitment to the message, along with a special encryption of the commitment randomness. By contrast, in the scheme described below, the *extension key* contains a commitment to the *secret key*, along with an encryption (under the secret key)

of the commitment randomness. (Using the commitment randomness to hide the secret key, and using the secret key to hide the commitment randomness, is what leads to a circular-security assumption.) We show how to combine two extension keys to get an encryption, under the concatenation of the secret keys, of the *tensor product* of those keys; this in turn lets us extend a ciphertext encrypted under one of the keys to their concatenation. We now describe the construction.

As in the previous section, the scheme is parameterized by LWE parameters n and q, the standard error distribution χ (which is E-bounded for $E = \Theta(n)$), and $m = \lceil 2n \log q \rceil$. The system is defined as follows.

- Setup: output a uniformly random $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$.
- $\mathsf{Gen}(\mathbf{A})$: do the following, outputting $\mathbf{t}$ as the secret key and $(\mathbf{b}, \mathbf{P}, \mathbf{D})$ as the public extension key.

  1. Choose $\bar{\mathbf{t}} \leftarrow \chi^{n-1}$ and define $\mathbf{t} := (-\bar{\mathbf{t}}, 1) \in \mathbb{Z}^n$. Choose $\mathbf{e} \leftarrow \chi^m$ and define

  $$\begin{aligned} \mathbf{b} &:= \mathbf{t}\mathbf{A} + \mathbf{e} \\ &\approx \mathbf{t}\mathbf{A} \in \mathbb{Z}_q^m. \qquad \text{(error } E\text{)} \end{aligned}$$

  2. Choose a uniformly random $\mathbf{R} \leftarrow \{0,1\}^{m \times n^2 \ell}$ and define

  $$\mathbf{P} := \mathbf{A}\mathbf{R} + (\mathbf{I}_n \otimes \mathbf{t} \otimes \mathbf{g}) \in \mathbb{Z}_q^{n \times n^2 \ell}.$$

  3. As described in Section 2.2, choose an LWE matrix $\bar{\mathbf{D}} \in \mathbb{Z}_q^{nm\ell \times n^2 \ell}$ that satisfies $(\mathbf{I}_{m\ell} \otimes \mathbf{t}) \cdot \bar{\mathbf{D}} \approx \mathbf{0}$ (with error E), and define $\mathbf{D} := \bar{\mathbf{D}} + (\mathbf{R} \otimes \mathbf{g}^t \otimes \mathbf{e}_n^t)$, where $\mathbf{e}_n \in \{0,1\}^n$ denotes the nth standard basis vector. Notice that, because $\mathbf{t} \cdot \mathbf{e}_n^t = 1$, we have

  $$(\mathbf{I}_{m\ell} \otimes \mathbf{t}) \cdot \mathbf{D} \approx \mathbf{R} \otimes \mathbf{g}^t. \qquad \text{(error } E\text{)}$$

- $\mathsf{Enc}(\mathbf{t}, \mu \in \{0,1\})$: This is standard GSW encryption. Specifically, as described in Section 2.2, choose an LWE matrix $\bar{\mathbf{C}} \in \mathbb{Z}_q^{n \times n\ell}$ that satisfies $\mathbf{t}\bar{\mathbf{C}} \approx \mathbf{0}$, and output the ciphertext $\mathbf{C} := \bar{\mathbf{C}} + \mu(\mathbf{I}_n \otimes \mathbf{g})$. Notice that $\mathbf{t}$, $\mathbf{C}$ satisfy the GSW relation

  $$\mathbf{t}\mathbf{C} = \mathbf{t}\bar{\mathbf{C}} + \mu(\mathbf{t} \otimes 1) \cdot (\mathbf{I}_n \otimes \mathbf{g}) \approx \mu(\mathbf{t} \otimes \mathbf{g}). \qquad \text{(error } E_{\mathbf{C}}\text{)}$$

- $\mathsf{Dec}(\mathbf{t}, \mathbf{C})$: this is standard GSW decryption.

We again stress that ciphertexts in the above system are just GSW ciphertexts (with no auxiliary information), so homomorphic addition and multiplication work as usual (and as in Section 3). The IND-CPA security of the system follows immediately from the circular-security assumption that LWE samples for secret $\mathbf{t}$ are indistinguishable from uniform, given $(\mathbf{A}, \mathbf{b}, \mathbf{P}, \mathbf{D})$ as constructed above.

4.1 Extending a Ciphertext to a New Key

We now show how to extend a (potentially multi-key) ciphertext to an additional key, so as to preserve the GSW relation for the concatenation of the secret keys. Specifically, suppose we have a ciphertext $\mathbf{C} \in \mathbb{Z}_q^{n' \times n'\ell}$ that encrypts μ under a key $\mathbf{t} \in \mathbb{Z}^{n'}$, i.e.,

$$\mathbf{t}\mathbf{C} \approx \mu(\mathbf{t} \otimes \mathbf{g}). \qquad \text{(error } E_{\mathbf{C}}\text{)}$$

In this setting, $n' = nk$ for some positive integer $k \geq 1$, and $\mathbf{t} = (\mathbf{t}_1, \ldots, \mathbf{t}_k)$ is the concatenation of k individual secret keys $\mathbf{t}_i \in \mathbb{Z}^n$ for which we know the associated vector $\mathbf{b}_i \approx \mathbf{t}_i\mathbf{A} \in \mathbb{Z}_q^m$ (with error E) from the public extension key. (We will not need the extension key's other components $\mathbf{P}$, $\mathbf{D}$.)

We wish to extend $\mathbf{C}$ to an additional secret key $\mathbf{t}^*$ for which we know the associated matrices $\mathbf{P}^*$, $\mathbf{D}^*$ from the public extension key (we will not need the associated $\mathbf{b}^*$). More precisely, we want to generate a ciphertext $\mathbf{C}'$ that encrypts μ under $\mathbf{t}' = (\mathbf{t}, \mathbf{t}^*) \in \mathbb{Z}^{n(k+1)}$, i.e., we want

$$\mathbf{t}'\mathbf{C}' \approx \mu(\mathbf{t}' \otimes \mathbf{g}) = \mu \begin{pmatrix} \mathbf{t} \otimes \mathbf{g} & \mathbf{t}^* \otimes \mathbf{g} \end{pmatrix}.$$

To do this, we output

$$\mathbf{C}' := \begin{pmatrix} \mathbf{C} & \mathbf{X} \\ & \mathbf{X}^* \end{pmatrix}$$

where $\mathbf{X}' = \left(\begin{smallmatrix} \mathbf{X} \\ \mathbf{X}^* \end{smallmatrix}\right)$ is as defined below. Notice that by construction,

$$\mathbf{t}'\mathbf{C}' \approx \begin{pmatrix} \mu(\mathbf{t} \otimes \mathbf{g}) & \mathbf{t}'\mathbf{X}' \end{pmatrix}. \qquad \text{(error } E_{\mathbf{C}}\text{)}$$

Below we show how to satisfy

$$\mathbf{t}'\mathbf{X}' = \mathbf{t}\mathbf{X} + \mathbf{t}^*\mathbf{X}^* \approx \mu(\mathbf{t}^* \otimes \mathbf{g}) \tag{4.1}$$

with error

$$E_{\mathbf{X}'} = (n^2 \cdot (k\ell + 1)^2 \cdot m + E_{\mathbf{C}}) \cdot E,$$

which yields $\mathbf{t}'\mathbf{C}' \approx \mu(\mathbf{t}' \otimes \mathbf{g})$ with error $\max\{E_{\mathbf{C}}, E_{\mathbf{X}'}\} = E_{\mathbf{X}'}$, as desired.

Constructing $\mathbf{X}'$. We construct $\mathbf{X}'$ in two steps:

1. Using just the $\mathbf{b}_i$ and $\mathbf{P}^*$, $\mathbf{D}^*$ (but not the ciphertext $\mathbf{C}$), we construct $\mathbf{Y}' = \left(\begin{smallmatrix} \mathbf{Y} \\ \mathbf{Y}^* \end{smallmatrix}\right)$ that satisfies

   $$\mathbf{t}'\mathbf{Y}' = \mathbf{t}\mathbf{Y} + \mathbf{t}^*\mathbf{Y}^* \approx (\mathbf{t} \otimes \mathbf{t}^* \otimes \mathbf{g}) \tag{4.2}$$

   with error $E_{\mathbf{Y}'} = (k\ell + 1) \cdot m \cdot E$. This construction is described below.

2. We then obtain $\mathbf{X}'$ by multiplying $\mathbf{Y}'$ by a certain binary matrix that is derived from the ciphertext $\mathbf{C}$. Essentially, this step just replaces $\mathbf{t}$ with $\mu\mathbf{g}$ in the right-hand side of Equation (4.2) (while consuming the existing $\mathbf{g}$).

   Let $\bar{\mathbf{C}} := \mathbf{C} \cdot (\mathbf{e}_n^t \otimes \mathbf{I}_\ell) \in \mathbb{Z}_q^{nk \times \ell}$ consist of the last ℓ columns of $\mathbf{C}$, so that

   $$\mathbf{t}\bar{\mathbf{C}} \approx \mu(\mathbf{t} \otimes \mathbf{g}) \cdot (\mathbf{e}_n^t \otimes \mathbf{I}_\ell) = \mu\mathbf{g}. \qquad \text{(error } E_{\mathbf{C}}\text{)} \tag{4.3}$$

   Define the binary matrix

   $$\mathbf{S} := (\mathbf{I}_{nk} \otimes \mathbf{I}_n \otimes \mathbf{g}^{-1})\left[\bar{\mathbf{C}} \otimes \mathbf{I}_n\right] \in \{0,1\}^{n^2 k\ell \times \ell}, \tag{4.4}$$

   and observe that

   $$\begin{aligned}
   \mathbf{t}'\mathbf{Y}' \cdot \mathbf{S} &\approx (\mathbf{t} \otimes \mathbf{t}^* \otimes \mathbf{g}) \cdot \mathbf{S} && \text{(Equation (4.2); error } n^2 k\ell \cdot E_{\mathbf{Y}'}\text{)} \\
   &= (\mathbf{t} \otimes \mathbf{t}^*) \cdot (\bar{\mathbf{C}} \otimes \mathbf{I}_n) && \text{(Equation (4.4))} \\
   &= (\mathbf{t}\bar{\mathbf{C}}) \otimes \mathbf{t}^* \\
   &\approx \mu(\mathbf{g} \otimes \mathbf{t}^*). && \text{(Equation (4.3), } \|\mathbf{t}^*\|_\infty \leq E\text{, so error } E_{\mathbf{C}} \cdot E\text{)}
   \end{aligned} \tag{4.5}$$

Notice that the right-hand side of Equation (4.5) is exactly the desired right-hand side of Equation (4.1), but permuted (because the arguments of the Kronecker product are swapped). So let $\mathbf{\Pi}$ be the permutation matrix for which $(\mathbf{g} \otimes \mathbf{t}^*)\mathbf{\Pi} = (\mathbf{t}^* \otimes \mathbf{g})$ for any $\mathbf{t}^*$, and define

$$\mathbf{X}' := \mathbf{Y}' \cdot \mathbf{S} \cdot \mathbf{\Pi},$$

which by the above satisfies Equation (4.1), as desired.

Constructing $\mathbf{Y}'$. We now describe the construction of $\mathbf{Y}' = \left(\begin{smallmatrix}\mathbf{Y}\\ \mathbf{Y}^*\end{smallmatrix}\right)$ to satisfy Equation (4.2). To do this we use the public matrices $\mathbf{P}^*, \mathbf{D}^*$ associated with $\mathbf{t}^*$, which by construction satisfy

$$\begin{aligned} \mathbf{P}^* &= \mathbf{A}\mathbf{R}^* + (\mathbf{I}_n \otimes \mathbf{t}^* \otimes \mathbf{g}) \\ (\mathbf{I}_{m\ell} \otimes \mathbf{t}^*) \cdot \mathbf{D}^* &\approx \mathbf{R}^* \otimes \mathbf{g}^t \qquad \text{(error } E\text{)} \end{aligned} \tag{4.6}$$

for some binary matrix $\mathbf{R}^* \in \{0,1\}^{m \times n^2 \ell}$. Recalling that $\mathbf{t} \in \mathbb{Z}^{nk}$ is the concatenation of k individual secret keys $\mathbf{t}_i \in \mathbb{Z}^n$, we also define $\mathbf{b} \in \mathbb{Z}_q^{mk}$ to be the concatenation of the associated $\mathbf{b}_i \approx \mathbf{t}_i\mathbf{A} \in \mathbb{Z}_q^m$ (all with error E), so

$$\mathbf{b} \approx \mathbf{t} \cdot (\mathbf{I}_k \otimes \mathbf{A}). \qquad \text{(error } E\text{)} \tag{4.7}$$

First, we define

$$\mathbf{Y} := \mathbf{I}_k \otimes \mathbf{P}^* = (\mathbf{I}_k \otimes \mathbf{A}\mathbf{R}^*) + (\mathbf{I}_{nk} \otimes \mathbf{t}^* \otimes \mathbf{g}).$$

Observe that

$$\begin{aligned} \mathbf{t}\mathbf{Y} &= \mathbf{t} \cdot (\mathbf{I}_k \otimes \mathbf{A}\mathbf{R}^*) + (\mathbf{t} \otimes 1 \otimes 1) \cdot (\mathbf{I}_{nk} \otimes \mathbf{t}^* \otimes \mathbf{g}) \\ &= \mathbf{t} \cdot (\mathbf{I}_k \otimes \mathbf{A}) \cdot (\mathbf{I}_k \otimes \mathbf{R}^*) + (\mathbf{t} \otimes \mathbf{t}^* \otimes \mathbf{g}) \\ &\approx \mathbf{b} \cdot (\mathbf{I}_k \otimes \mathbf{R}^*) + (\mathbf{t} \otimes \mathbf{t}^* \otimes \mathbf{g}). && \text{(Equation (4.7); error } m \cdot E\text{.)} \end{aligned}$$

Therefore, in order to satisfy Equation (4.2), it suffices to construct $\mathbf{Y}^*$ to satisfy

$$\mathbf{t}^*\mathbf{Y}^* \approx -\mathbf{b} \cdot (\mathbf{I}_k \otimes \mathbf{R}^*).$$

with error $km\ell \cdot E$. To do this, we define

$$\begin{aligned} \mathbf{s} &:= -[\mathbf{b}](\mathbf{I}_k \otimes \mathbf{I}_m \otimes \mathbf{g}^{-t}) \in \{0,1\}^{km\ell} \\ \mathbf{Y}^* &:= (\mathbf{s} \otimes \mathbf{I}_n) \cdot (\mathbf{I}_k \otimes \mathbf{D}^*). \end{aligned} \tag{4.8}$$

Then observe that

$$\begin{aligned} \mathbf{t}^*\mathbf{Y}^* &= (1 \otimes \mathbf{t}^*) \cdot (\mathbf{s} \otimes \mathbf{I}_n) \cdot (\mathbf{I}_k \otimes \mathbf{D}^*) \\ &= (\mathbf{s} \otimes 1) \cdot (\mathbf{I}_{km\ell} \otimes \mathbf{t}^*) \cdot (\mathbf{I}_k \otimes \mathbf{D}^*) \\ &\approx \mathbf{s} \cdot (\mathbf{I}_k \otimes \mathbf{R}^* \otimes \mathbf{g}^t) && \text{(Equation (4.6); error } km\ell \cdot E\text{)} \\ &= -\mathbf{b} \cdot (\mathbf{I}_k \otimes \mathbf{R}^*) && \text{(Equation (4.8))} \end{aligned}$$

as desired. This completes the construction and analysis.

4.2 Instantiating the Parameters

We now bound the worst-case error growth when homomorphically evaluating a depth-d circuit of NAND gates for up to k individual keys. As above, let $n' = nk$. For two ciphertexts with error bounded by E^*, their homomorphic NAND has error bounded by $(n'\ell + 1) \cdot E^* = \mathrm{poly}(n, k, \ell) \cdot E^*$. Similarly, when we extend a ciphertext with error bounded by E^*, the result has error bounded by $(n^2 \cdot (k\ell + 1)^2 \cdot m + E^*) \cdot E = \mathrm{poly}(n, k, \ell) \cdot E^*$. Therefore, for any depth-d homomorphic computation on fresh ciphertexts encrypted under k keys, the result has error bounded by $\mathrm{poly}(n, k, \ell)^{k+d}$. Therefore, it suffices to choose a modulus q that exceeds four times this bound. Recalling that $\ell = \Theta(\log q) = \tilde{O}(k + d)$, this corresponds to a worst-case approximation factor of $\mathrm{poly}(n, k, d)^{k+d}$ for n-dimensional lattice problems.

References

[ACPS09] B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *CRYPTO*, pages 595–618. 2009.

[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 *EUROCRYPT*, pages 483–501. 2012.

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

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

[Gen09] C. Gentry. *A fully homomorphic encryption scheme*. Ph.D. thesis, Stanford University, 2009. `http://crypto.stanford.edu/craig`.

[GHV10] C. Gentry, S. Halevi, and V. Vaikuntanathan. i-hop homomorphic encryption and rerandomizable Yao circuits. In *CRYPTO*, pages 155–172. 2010.

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

[GVW15] S. Gorbunov, V. Vaikuntanathan, and D. Wichs. Leveled fully homomorphic signatures from standard lattices. In *STOC*, pages 469–477. 2015.

[HPS98] J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: A ring-based public key cryptosystem. In *ANTS*, pages 267–288. 1998.

[LPR10] V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. *Journal of the ACM*, 60(6):43:1–43:35, November 2013. Preliminary version in Eurocrypt 2010.

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

[MP12] D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT*, pages 700–718. 2012.

[MW15] P. Mukherjee and D. Wichs. Two round mutliparty computation via multi-key fhe. Cryptology ePrint Archive, Report 2015/345, 2015. To appear, EUROCRYPT 2016.

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

[Reg05] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):1–40, 2009. Preliminary version in STOC 2005.