Cryptographic Program Obfuscation

Sanjam Garg
University of California, Berkeley

February 2016

Chapter 1

Executive Summary

Cryptography is the “black magic,” of theoretical computer science, enabling tasks that often seem paradoxical or simply impossible. For example, consider the goal of enciphering data in a way such that only the designated recipient can make sense of it. Even this seemingly simple task of *encryption* has eluded mankind for centuries. Wars have been lost because of weaknesses in encryption methods, thought to be secure. Modern cryptography has provided us with encryption schemes based on sound mathematical principles giving very strong assurances of security. In fact, now we have a multitude of cryptographic primitives enabling far more ambitious goals than just encryption.

Beyond encryption, cryptographers have long wondered if it possible to *obfuscate* computer programs. More specifically, can we make computer programs “unintelligible” while preserving their functionality? Here, “unintelligibility” requirement means that the obfuscated program should hide the secrets that were embedded inside the original program. This is a very stringent requirement requiring that no automated “deobfuscator” can derive the secrets that we intend to hide inside the program. On the other hand, “preserving its functionality” means that the obfuscated program should be fully executable and have the same input-output behavior as the original program.

Next we explain the “unintelligibility” requirement with the example of how obfuscation can be used to convert a private key encryption scheme to a public key one. This classical application of obfuscation was first envisioned by Diffie and Hellman [DH76] with the goal of realizing the first public key encryption scheme. In a private key encryption scheme it is the same private key that enables both the tasks of encryption and decryption. So given the secret key one can encrypt a plaintext and obtain a ciphertext. The same secret key can then be used to decrypt the generated ciphertext and recover the original plaintext. On the other hand, in the case of a public key encryption, encryption is done using a public key, enabling encryption by anyone while decryption is done using the corresponding private key. Diffie and Hellman noticed that an obfuscation of the program that does the encryption, with the secret key embedded inside it, can be published as the public key. This would enable encryption by all while still preserving decryption to the holder of the secret key. This allows for a generic method for converting a private key encryption scheme into a public key one. In this example, obfuscation attempts to hide the private key inside the program.

As another example consider the goal of secure software patching, something that is typically not considered to be in the domain of cryptography. Consider for instance the security software Norton. Let’s say that one day Norton’s team of software security engineers, finds a bug in their software and releases a patch to fix this bug. An attacker could reverse-engineer this newly released patch and figure out what the original bug was. Widespread deployment of software patches takes a lot of time, certainly a few days and sometimes even weeks. Knowledge of this bug allows an attacker to attack such unpatched machines on the Internet. This actually turns to be a real security problem that exists when software patches are released today. Obfuscation offers a very elegant solution to this problem. If instead of releasing the patch in its unobfuscated form, Norton was to release the patch in its obfuscated form; then that would keep the patch "unitelligible." Here the security bug is the secret which the obfuscation will now be protecting. Of course, once large scale deployment of the obfuscated patch has completed, Norton could move to a more efficient unobfuscated version of the patch.

Defining obfuscation rigorously turns out to be somewhat involved. For example, the requirement of maintaining functionality implicitly places some limitations on what obfuscation can accomplish. In particular, consider a program that on a specified input prints its own code. No obfuscation method that preserves functionality can successfully hide the code of this program.

The most natural way to define unintelligibility of obfuscation is to require that giving the obfuscated program is as good as giving *black-box* oracle access to the program [BGI$^+$01]. More specifically, whatever an attacker could learn from the obfuscated program could have been learned using just oracle access to the program implementing that function. However, this notion of obfuscation, has been shown to be impossible albeit only for some unnatural class of circuits. An alternative weaker notion of obfuscation is *indistinguishability obfuscation* [BGI$^+$01, GR07, GGH$^+$13b]. This notion only requires computational indistinguishability of obfuscations of two circuits computing the exact same functionality. In particular, if two circuits C_1 and C_2 are such that $C_1(x) = C_2(x)$ for all x then this definition requires that the obfuscation of C_1, namely $\mathcal{O}(C_1)$, is computationally indistinguishable from $\mathcal{O}(C_2)$.

Given the impossibility results of Barak et al. [BGI$^+$01], cryptographers were very skeptical about the existence of general obfuscation results. However, recent works [GGH$^+$13b] provide plausible candidate constructions based on multilinear maps [BS02, GGH13a]. These works provide the indistinguishability obfuscation notion. At first sight, this might seem week. Goldwasser and Rothblum [GR07], provide strong philosophical argument supporting the meaningfulness of this notion. In particular they show that (efficiently computable) indistinguishability obfuscators achieve the notion of Best-Possible Obfuscation: Informally, a best-possible obfuscator guarantees that its output hides as much about the input circuit as any other circuit (of a certain size).

Current obfuscation methods are prohibitively inefficient. Furthermore, our understanding of the security of these constructions is limited. Next we provide an overview of the techniques used to construct multilinear maps, obfuscation and the resulting applications. Details will be provided in the main body of this report.

1.1 Technical Insights - Multilinear Maps

In his breakthrough result, Gentry [Gen09] constructed a *fully-homomorphic encryption* scheme that enabled arbitrary computation on encrypted data without allowing for decryption. However for many applications, the ability to perform arbitrary computation on encrypted data along with the ability to check if two ciphertexts encrypt the same message is essential. In his scheme, Gentry relied on "noise" to hide messages. The presence of noise, which helps hide messages without restricting arbitrary computation on them, seems to be in conflict with the goal of equality checking. Recent candidate constructions [GGH13a, CLT13, GGH15] overcome this obstacle by introducing techniques that enable equality testing even in the presence of noise. Next we present an overview of how this is achieved in the GGH construction [GGH13a].

The GGH constructions work in polynomial rings and use principal ideals in these rings (and their associated lattices). In a nutshell, an instance of the GGH construction has a secret short ring element $\mathbf{g} \in R$, generating a principal ideal $\mathcal{I} = \langle \mathbf{g} \rangle \subset R$. In addition, it has an integer parameter q and another secret $\mathbf{z} \in R/qR$, which is chosen at random (and hence is not small).

Think of a term like g^x in a discrete-log system as an "encoding" of the "plaintext exponent" x. In GGH, the role of the "plaintext exponents" is played by the elements in $R/\mathcal{I}$ (i.e. cosets of $\mathcal{I}$), which is "encoded" via division by $\mathbf{z}$ in R_q. In a few more details, GGH system provides many levels of encoding, where a level-i encoding of the coset $\boldsymbol{e}_{\mathcal{I}} = \boldsymbol{e} + \mathcal{I}$ is an element of the form $\boldsymbol{c}/\mathbf{z}^i \bmod q$ where $\boldsymbol{c} \in \boldsymbol{e}_{\mathcal{I}}$ is short. It is easy to see that such encodings can be both added and multiplied, so long as the numerators remain short. This enables the goal of allowing for computation on the hidden "plaintext exponents." Additionally, GGH scheme publishes a "zero testing parameter" that enables to test if two elements encode the same coset at a given level, without violating security (e.g., it should still be hard to compute x from an encoding of x at higher levels). Namely, GGH add to the public parameters an element of the form $\mathbf{p}_{zt} = \boldsymbol{h} \cdot \boldsymbol{z}^{\kappa}/\mathbf{g} \bmod q$ for a not-too-large $\boldsymbol{h}$, where κ is the level of multilinearity. Multiplying an encoding of zero (at the κ^{th} level) by $\mathbf{p}_{zt}$ (mod q) yields a small element, while multiplying an encoding of a non-zero by $\mathbf{p}_{zt}$ (mod q) yields a large element. Hence one can distinguish zero from non-zero, and by subtraction one can distinguish two encodings of the same element from encodings of two different elements.

Building on the GGH multilinear maps [CLT13, CLT15, GGH15] alternate constructions of multilinear maps have been obtained. The security of these constructions also relies on new assumptions. The suggested candidate constructions of multilinear map constructions also supported randomization of encodings. Several attacks have been shown in this setting [CHL$^+$15, CLR15, CLLT15, CFL$^+$16]. However, typical obfuscation candidate constructions, besides [GLSW15], do not reply on encodings of zero. Therefore, obfuscation constructions remain unaffected by these attacks and so we do not include discussion about these zeroing attacks in this report. We remark that very recent attacks [JHC16, MA16] are applicable for the setting where zero's are not provided. These attacks can be prevented using the parameters carefully. Finally, a very recent attack [MSZ16] provides an attack that works directly on certain obfuscation candidates specifically for the GGH construction. We note that this attack isn't actually known to break the known candidates but points to something that we should worry about. Furthermore, we need to develop defences that would avoid these attacks in a provable sense. We don't consider this attack in this report.

1.2 Technical Insights - Obfuscation

As stated, multilinear maps enable arbitrary computing on encrypted data along with ability to check if two ciphertexts encrypt the same message. Obfuscation is more general and requires revealing specific functions of encrypted data already which *program obfuscation* [BGI$^+$01, BGI$^+$12]. Positive results for obfuscation, without multilinear maps, have been limited to simple classes of programs such as point functions [Can97, CMR98, LPS04, Wee05, CD08, BC10], testing hyperplane membership [CRV10] and a few other simple programs [HRsV07, HMLS07, Had10, CCV12]. On the other hand, multilinear maps based construction work for a arbitrary programs [GGH$^+$13b]. This new construction, the focus of this report, provides a realization of the indistinguishability obfuscation notion. Many follow up works have improved these candidate construction in several ways [BR14b, BGK$^+$14, AGIS14, MSW14, BV15, AJ15, BMSZ15, AJS15]

Technically speaking the obfuscation construction is obtained by encrypting the program in such a manner such that computing the encrypted program on an input is same as evaluating an input specific polynomial on the encrypted values and checking if it evaluates to zero or not. In this setting the polynomial for each input is different and this is what enables computation for any input choice. Security itself is based on the fact that any other (bounded degree) polynomial that can be computed on the encrypted data evaluates to something random and hence doesn't reveal anything.

Note on VBB obfuscation. Even though *Virtual black box* obfuscation [BGI$^+$01] (VBB in short) is impossible in general it various flavors of it have been realized for special functions such as conjunctions [BR13, BR14a, BVWW16] and for general circuits in idealized models [BR14b, BGK$^+$14]. Our inability to provide more general results can be explained by the negative results of [BGI$^+$01], who showed that there exist families of "unobfuscatable" functions for which the VBB definition is impossible to achieve *in the plain model*. However this result does not apply to the setting of generic multilinear attacks, in which case the VBB notion can actually be realized [BR14b, BGK$^+$14]. These works provide evidence that no *algebraic* attacks (that respect multilinear maps) against these candidate constructions leak anything beyond what could be leaked in a black-box manner and provide heuristic evidence that these obfuscation mechanisms offer strong security for "natural" functions.

1.3 Technical Insights - Applications

Obfuscation radically enhances our tool set and opens a floodgate of applications. One-round multi-party key-exchange is a classical example. Diffie and Hellman in their seminal paper [DH76] provided the first construction of a one-round two-party key-exchange protocol which was then generalized to the three party setting by Joux [Jou00] using Weil and Tate pairings. Obfuscation can be used realize a result that works for arbitrary number of parties [BZ14, GPSZ16]. Another very interesting application of obfuscation has been in demonstrating the hardness of finding Nash Equilibrium [BPR15, GPS15]. Next we describe at length the various applications of obfuscation to the problem of enabling flexible encryption.

How flexible can we make access to encrypted data? Enabling encryption by arbitrary parties motivated the invention of public key encryption [DH76, RSA78]. However, enabling fine-grained decryption capabilities has remained an elusive goal![Sha84, SW05, GPSW06]. Shamir [Sha84] proposed the problem of non-interactively associating identities with encrypted data, and later Sahai and Waters [SW05] asked if an encrypter at the time of encryption can non-interactively embed any arbitrary decryption policy into his ciphertext. All primitive described above enabled encrypters with the ability to specify who can decrypt. However at the same time these tools do not provide for a mechanism to specific what a decrypter can learn. A decrypter learns either the entire message or nothing about it. More specifically, in functional encryption [BSW11, O'N10], ciphertexts encrypt inputs x and keys are issued for functions f. The striking feature of this system is that given an encryption of x, the key corresponding to f can be used to obtain $f(x)$ but nothing else about x. Furthermore, *any arbitrary* collusion of key holders relative to many functions f_i does not yield any more information about x beyond what is "naturally revealed" by each of them individually (i.e. $f_i(x)$ for all i). Using obfuscation, [GGH$^+$13b] give a construction of functional encryption for general circuits.[1]

[1] We note that the [GGH$^+$13c] construction gets a weaker indistinguishability notion of security for functional encryption. However this can be upgraded to natural simulation-based definitions of security using the work of De Caro et al. [CIJ$^+$13].

Witness Encryption. Encryption in all its myriad flavors has always been imagined with some known recipient in mind. But, what if the intended recipient of the message is not known and may never be known to the encrypter? For example, consider the task of encrypting to someone who knows a solution to a crossword puzzle that appeared in the *The New York Times.* Or, in general, a solution to some NP search problem which he might know or might acquire over a period of time. The encrypter on the other hand may even be unaware of the existence of a solution.

[GGSW13] proposed the concept of *witness encryption* which captures this intuition and realized it based on our noisy multilinear maps. It was later shown in [GGH$^+$13b] that it can be based on indistinguishability obfuscation as well. Witness Encryption is closely related to the notion of computational secret sharing for NP-complete access structures, first posed by Rudich in 1989 [Rud89] (see [Bei11]). As observed by Rudich, this primitive already suffices for converting private-key cryptosystems to public-key ones. Witness encryption has found applications elsewhere as well. Most prominently, Goldwasser et al. [GKP$^+$13] used (a variant of) witness encryption for constructing a variant of attribute-based encryption scheme for polynomial-time Turing machines, where the sizes of secret keys depend only on the size of the Turing machine (rather than its runtime). Furthermore in these constructions, the decryption algorithm has an input-specific runtime rather than worst-case runtime (at the price of revealing this runtime).

Obfuscation has many other applications and we refer the reader to [SW14, GGHR14] for the first works on the topic.

1.4 Roadmap

In Sections 2.1 and 2.2 we recall some of the preliminary definitions and concepts needed for explaining multilinear maps, as needed for constructing obfuscation. The construction for multilinear maps in provided in Section 2.3. These three sections have been largely taken verbatim from the authors doctoral thesis. Next we provide a construction of obfuscation for $\mathbf{NC}^1$ as in [BGK$^+$14] in Section 3.2 and provide extension to general circuits in Section 3.3 as in [GGH$^+$13b].

Chapter 2

Multilinear Maps

2.1 Preliminaries I : Lattices

We denote set of complex number by $\mathbb{C}$, real numbers by $\mathbb{R}$, the rationals by $\mathbb{Q}$ and the integers by $\mathbb{Z}$. For a positive integer n, $[n]$ denotes the set $\{1, \dots, n\}$. By convention, vectors are assumed to be in column form and are written using bold lower-case letters, e.g. $\boldsymbol{x}$. The ith component of $\boldsymbol{x}$ will be denoted by x_i. We will use $\boldsymbol{x}^T$ to denotes the transpose of $\boldsymbol{x}$. For a vector $\boldsymbol{x}$ in $\mathbb{R}^n$ or $\mathbb{C}^n$ and $p \in [1, \infty]$, we define the ℓ_p norm as $\|\boldsymbol{x}\|_p = \left(\sum_{i \in [n]} |x_i|^p\right)^{1/p}$ where $p < \infty$, and $\|\boldsymbol{x}\|_\infty = \max_{i \in [n]} |x_i|$ where $p = \infty$. Whenever p is not specified, $\|\boldsymbol{x}\|$ is assumed to represent the ℓ_2 norm (also referred to as the Euclidean norm).

Matrices are written as bold capital letters, e.g. $\boldsymbol{X}$, and the ith column vector of a matrix $\boldsymbol{X}$ is denoted $\boldsymbol{x}_i$. Finally we will denote the transpose and the inverse (if it exists) of a matrix $\boldsymbol{X}$ with $\boldsymbol{X}^T$ and $\boldsymbol{X}^{-1}$ respectively.

The natural security parameter throughout the report is λ, and all other quantities are implicitly assumed to be functions of λ. We use standard big-O notation to classify the growth of functions, and say that $f(\lambda) = \tilde{O}(g(\lambda))$ if $f(\lambda) = O(g(\lambda) \cdot log^c \lambda)$ for some fixed constant c. We let $\mathsf{poly}(\lambda)$ denote an unspecified function $f(\lambda) = O(\lambda^c)$ for some constant c. A *negligible* function, denoted generically by $\mathsf{negl}(\lambda)$, is an $f(\lambda)$ such that $f(\lambda) = o(\lambda^{-c})$ for every fixed constant c. We say that a function is *overwhelming* if it is $1 - \mathsf{negl}(\lambda)$.

The *statistical distance* between two distributions X and Y over a domain D is defined to be $\frac{1}{2}\sum_{d \in D} |\Pr[X = d] - \Pr[Y = d]|$. We say that two ensembles of distributions $\{X_\lambda\}$ and $\{Y_\lambda\}$ are *statistically indistinguishable* if for every λ the statistical distance between X_λ and Y_λ is negligible in λ.

Two ensembles of distributions $\{X_\lambda\}$ and $\{Y_\lambda\}$ are *computationally indistinguishable* if for every probabilistic poly-time (in λ) machine $\mathcal{A}$, $|\Pr[\mathcal{A}(1^\lambda, X_\lambda) = 1] - \Pr[\mathcal{A}(1^\lambda, Y_\lambda) = 1]|$ is negligible in λ. The definition is extended to non-uniform families of poly-sized circuits in the standard way.

2.1.1 Lattices

A lattice Λ is an additive discrete sub-group of $\mathbb{R}^n$, i.e., it is a subset $\Lambda \subset \mathbb{R}^n$ satisfying the following properties:

(subgroup) λ is closed under addition and subtraction,

(discrete) there is an $\epsilon > 0$ such that any two distinct lattice points $\boldsymbol{x} \neq \boldsymbol{y} \in \Lambda$ are at distance at least $\|\boldsymbol{x} - \boldsymbol{y}\| \geq \epsilon$.

Let $\boldsymbol{B} = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k\} \subset \mathbb{R}^n$ consist of k linearly independent vectors in $\mathbb{R}^n$. The lattice generated by the $\boldsymbol{B}$ is the set

$$\mathcal{L}(\boldsymbol{B}) = \{\boldsymbol{B}\boldsymbol{z} = \sum_{i=1}^{k} z_i \boldsymbol{b}_i : \boldsymbol{z} \in \mathbb{Z}^k\},$$

of all the integer linear combinations of the columns of $\boldsymbol{B}$. The matrix $\boldsymbol{B}$ is called a *basis* for the lattice $\mathcal{L}(\boldsymbol{B})$. The integers n and k are called the *dimension* and *rank* of the lattice. If $n = k$ then $\mathcal{L}(\boldsymbol{B})$ is called a *full-rank* lattice. We will only be concerned with full-rank lattices, hence unless otherwise mentioned we will assume that the lattice considered is full-rank.

The *minimum distance* $\lambda_1(\Lambda)$ of a lattice Λ is the length (in the Euclidean ℓ_2 norm, unless otherwise indicated) of its shortest nonzero vector: $\lambda_1(\Lambda) = \min_{\boldsymbol{x} \neq \boldsymbol{0}, \boldsymbol{x} \in \Lambda} \|\boldsymbol{x}\|$. More generally, the *ith successive minimum* $\lambda_i(\Lambda)$ is the smallest radius r such that Λ contains i linearly independent vectors of norm at most r. We write λ_1^∞ to denote the minimum distance measured in the ℓ_∞ norm (which as mentioned earlier, is defined as $\|\boldsymbol{x}\|_\infty = \max |x_i|$).

For lattices $\Lambda' \subseteq \Lambda$, the quotient group Λ/Λ' (also written as $\Lambda \mod \Lambda'$) is well-defined as the additive group of distinct *cosets* $\boldsymbol{v} + \Lambda'$ for $\boldsymbol{v} \in \Lambda$, with addition of cosets defined in the usual way.

2.1.2 Gaussians on Lattices

Review of Gaussian measure over lattices presented here follows the development by prior works [Reg04, AR05, MR07, GPV08, AGHS12]. For any real $s > 0$, define the (spherical) *Gaussian function* $\rho_s : \mathbb{R}^n \to (0, 1]$ with[1] parameter s as:

$$\forall \boldsymbol{x} \in \mathbb{R}^n, \rho_s(\boldsymbol{x}) = \exp(-\pi\langle \boldsymbol{x}, \boldsymbol{x}\rangle / s^2) = \exp(-\pi \|\boldsymbol{x}\|^2 / s^2).$$

For any real $s > 0$, and n-dimensional lattice Λ, define the (spherical) *discrete Gaussian distribution* over Λ as:

$$\forall \boldsymbol{x} \in \Lambda, D_{\Lambda,s}(\boldsymbol{x}) = \frac{\rho_s(\boldsymbol{x})}{\rho_s(\Lambda)}.$$

Smoothing Parameter. Micciancio and Regev [MR07] introduced a lattice quantity called the *smoothing parameter*, and related it other lattice parameters.

Definition 2.1.1 (Smoothing Parameter, [MR07, Definition 3.1])**.** *For an n-dimensional lattice Λ, and positive real $\epsilon > 0$, we define its* smoothing parameter *denoted $\eta_\epsilon(\Lambda)$, to be the smallest s such that $\rho_{1/s}(\Lambda^* \setminus \{\boldsymbol{0}\}) \leq \epsilon$.*

Intuitively, for a small enough ϵ, the number $\eta_\epsilon(\Lambda)$ is sufficiently larger than a fundamental parallelepiped of Λ so that sampling from the corresponding Gaussian "wipes out the internal structure" of Λ. The following Lemma 2.1.3 and Corollary 2.1.4 formally provide this claim. The bounds on $\eta_\epsilon(\Lambda)$ are specified by Lemma 2.1.2. Finally Lemma 2.1.5 provides bounds on the length of a vector sampled from a Gaussian.

Lemma 2.1.2 ([MR07, Lemma 3.3])**.** *For any n-dimensional lattice Λ and positive real $\epsilon > 0$, we have that*

$$\eta_\epsilon(\Lambda) \leq \sqrt{\frac{\ln(2n(1 + 1/\epsilon))}{\pi}} \cdot \lambda_n(\Lambda).$$

[1] The Gaussian function can be defined more generally as being centered around a specific vector $\boldsymbol{c}$ instead of $\boldsymbol{0}$ as done here. The simpler definition considered here suffices for our purposes.

The following lemma explains the name “smoothing parameter.”

Lemma 2.1.3 ([MR07, Lemma 4.1])**.** *For any lattice Λ, $\epsilon > 0$, $s \geq \eta_\epsilon(\Lambda)$, and $\boldsymbol{c} \in \mathbb{R}^n$, the statistical distance between $D_s + \boldsymbol{c} \mod \Lambda$ and the uniform distribution modulo Λ is at most $\epsilon/2$.*

Corollary 2.1.4 ([GPV08, Corollary 2.8])**.** *Let Λ, Λ' be n-dimensional lattices, with $\Lambda' \subseteq \Lambda$. Then for any $\epsilon \in (0, \frac{1}{2})$, any $s \geq \eta_\epsilon(\Lambda')$, the distribution of $(D_{\Lambda,s} \mod \Lambda')$ is within a statistical distance at most 2ϵ of uniform over $(\Lambda \mod \Lambda')$.*

Lemma 2.1.5 ([MR07, Lemma 4.4] and [BF11b, Proposition 4.7])**.** *For any n-dimensional lattice Λ, and $s \geq \eta_\epsilon(\Lambda)$ for some negligible ϵ, then for any constant $\delta > 0$ we have*

$$\Pr_{\boldsymbol{x} \leftarrow D_{\Lambda,s}} \left[(1-\delta)s\sqrt{\frac{n}{2\pi}} \leq \|\boldsymbol{x}\| \leq (1+\delta)s\sqrt{\frac{n}{2\pi}} \right] \geq 1 - \mathsf{negl}(n).$$

2.2 Preliminaries II : Algebraic Number Theory Background

Algebraic number theory is the study of *number fields.* Here we review the background essential for understanding our encoding scheme. We consider the special case of *cyclotomic* number fields as a special example of particular interest. Much of our description here follows [LPR10], and we refer the reader to [Jan96, Ste04, Oss08, Wes99] for detailed background reading.

2.2.1 Number Fields and Ring of Integers

An algebraic number field (or simply number field) K is a finite (and hence algebraic) field extension of the field of rational numbers $\mathbb{Q}$. In this section we will recall definition of some of these elementary notions.

Definition 2.2.1 (Algebraic Number and Algebraic Integer)**.** *We say that $\zeta \in \mathbb{C}$ is an* algebraic number *if it is a root of a polynomial $f(x) \in \mathbb{Q}[x]$. Furthermore, we say that that ζ is an* algebraic integer *if additionally $f(x)$ is a monic (a polynomial whose leading coefficient is 1) polynomial in $\mathbb{Z}[x]$.*

Definition 2.2.2 (Minimal Polynomial)**.** *The* minimal polynomial *of ζ is the monic polynomial $f(x) \in \mathbb{Q}[x]$ of least positive degree such that $f(\zeta) = 0$.*

The *conjugates* of ζ are defined by all the roots of its minimal polynomial.

Proposition 2.2.3 ([Ste04, Lemma 5.1.3])**.** *If ζ is an algebraic integer, then the minimal polynomial of ζ is in $\mathbb{Z}[x]$.*

Proposition 2.2.4 ([Ste04, Proposition 5.1.5])**.** *The set of all algebraic integers form a ring, i.e., the sum and product of two algebraic integers is again an algebraic integer.*

Now we are ready to define the notion of a number field and its ring of integers.

Definition 2.2.5 (Number Field and Ring of Integers)**.** *A* number field *is a field extension $K = \mathbb{Q}(\zeta)$ obtained by adjoining an algebraic number ζ to the field of rationals $\mathbb{Q}$. The* ring of integers *of a number field K is the ring*

$$\mathcal{O}_K = \{x \in K : x \textit{ is an algebraic integer.}\}$$

Let the minimal polynomial $f(x)$ of ζ have degree n. Then because $f(\zeta) = 0$, there is a natural isomorphism between $\mathbb{Q}[x] \mod f(x)$ and K, given by $x \mapsto \zeta$, and the number field K can be seen as an n-dimensional vector space over $\mathbb{Q}$ with basis $\{1, \zeta, \ldots, \zeta^{n-1}\}$. This is called the *power basis* of K.

The case of Cyclotomic Number Fields. Let $\zeta_m = e^{2\pi\sqrt{-1}/m} \in \mathbb{C}$ denote a *primitive* m-th root of unity. (Recall that an mth root of unity is said to be a *primitive* root if it is not a kth root for some $0 < k < m$.)

Definition 2.2.6 (Cyclotomic Polynomial)**.** *The m-th* cyclotomic polynomial*, denote by $\Phi_m(x)$, is defined as the product*

$$\Phi_m(x) = \prod_{k \in \mathbb{Z}_m^*} (x - \zeta_m^k).$$

Observe that the values ζ^k run over all the primitive m^{th} roots of unity in $\mathbb{C}$, thus $\Phi_m(x)$ has degree $n = \varphi(m)$, where $\varphi(m)$ denotes the *Euler's totient* or *phi function*. Recall that if m is a positive integer, then $\varphi(m)$ is the number of integers in the set $\{1, 2, \ldots, m\}$ that are relatively prime to m.

It is easy to see that $\Phi_m(x)$ is monic. It is also known (a nontrivial result due to Gauss) that $\Phi_m(x)$ is in $\mathbb{Z}[x]$ and is irreducible over $\mathbb{Q}$. Therefore ζ_m is an algebraic integer with the minimal polynomial $\Phi_m(x)$.

The cyclotomic polynomial $\Phi_m(x)$ may be computed by (exactly) dividing $x^n - 1$ by the cyclotomic polynomials of the proper divisors of n previously computed recursively (setting, $\Phi_1(x) = x - 1$) by the same method:

$$\Phi_m(x) = \frac{x^m - 1}{\prod_{\substack{d|m \\ d<m}} \Phi_d(x)}.$$

Two useful facts about cyclotomic polynomials are that $\Phi_m(x) = \frac{x^m-1}{x-1} = x^{m-1} + \ldots + x + 1$ for prime m, and $\Phi_m(x) = \Phi_{m_0}(x^{m/m_0})$ where m_0 is the radical of m, i.e., the product of all primes diving m. For instance, $\Phi_8(c) = x^4 + 1$ and $\Phi_9(x) = x^6 + x^3 + 1$. We will be most interested in the case when $m \geq 2$ is a power of 2 in which case $\Phi_m(x) = x^{m/2} + 1$. (However, not all cyclotomic polynomials have 0-1, or even small coefficients: e.g., $\Phi_6(x) = x^2 - x + 1$, $\Phi_{3 \cdot 5 \cdot 7}$ has a -2 coefficient, and $\Phi_{3 \cdot 5 \cdot 7 \cdot 11 \cdot 13}(x)$ has coefficients with magnitudes as large as 22.)

Definition 2.2.7. *The mth* cyclotomic field $\mathbb{Q}(\zeta_m)$ *(with $m > 2$) is obtained by adjoining ζ_m to $\mathbb{Q}$.*

Proposition 2.2.8 ([Jan96, p 48, Proposition 4.3])**.** *The ring of integers in $\mathbb{Q}(\zeta_m)$ is $\mathbb{Z}(\zeta_m)$. This ring $\mathbb{Z}(\zeta_m)$ is called the* cyclotomic ring.

2.2.2 Embeddings and Geometry

In this section we will recall various geometric interpretations of a number field and most importantly define different notion of norm essential for our study.

Canonical Embedding. A number field $K = \mathbb{Q}(\zeta)$ of degree[2] n has [Wes99, p 9, Proposition 2.1] exactly n field homomorphisms $\sigma_i = K \hookrightarrow \mathbb{C}$ that fix every element of $\mathbb{Q}$. Concretely, these embeddings map ζ to each of its conjugates; it can be verified that these are the only field homomorphisms from K to $\mathbb{C}$ because ζ's conjugates are the only roots of ζ's minimal polynomial $f(x)$. An embedding whose image lies in $\mathbb{R}$ (corresponding to a real root of $f(x)$) is called a *real*

[2] Recall that a number field $K = Q(\zeta)$ is isomorphic to $\mathbb{Q}[x]/f(x)$ where $f(x)$ is the minimal polynomial of ζ. The degree of K defined to be the value $[K : \mathbb{Q}]$, is same as [Ste04, p 28] the degree of the polynomial $f(x)$. (More generally, if $K \subset L$ are number fields, we let $[L : K]$ denote the dimension of L viewed as a K-vector space.)

embedding; otherwise (for a complex root of $f(x)$) it is called a *complex embedding.* Because complex roots of $f(x)$ come in conjugate pairs, so too do the complex embeddings. The number of real embeddings is denoted s_1 and the number of *pairs* of complex embeddings is denoted by s_2, so we have $n = s_1 + 2s_2$. The pair (s_1, s_2) is called the *signature* of K. By convention, we let $\{\sigma_j\}_{j\in[s_1]}$ be the real embeddings, and order the complex embeddings so that $\sigma_{s_1+s_2+j} = \overline{\sigma_{s_1+j}}$ for $j \in [s_2]$. The *canonical embedding* $\sigma : K \to \mathbb{R}^{s_1} \times \mathbb{C}^{2s_2}$ is defined as

$$\sigma(x) = (\sigma_1(x), \ldots, \sigma_n(x)).$$

The canonical embedding σ is a field homomorphism from K to $\mathbb{R}^{s_1} \times \mathbb{C}^{2s_2}$, where multiplication and addition in $\mathbb{R}^{s_1} \times \mathbb{C}^{2s_2}$ are component-wise (since σ is a ring homomorphism). Due to the pairing of the complex embeddings, σ maps into the following space $H \subseteq \mathbb{R}^{s_1} \times \mathbb{C}^{2s_2} \subset \mathbb{C}^n$:

$$H = \{(x_1, \ldots, x_n) \in \mathbb{R}^{s_1} \times \mathbb{C}^{2s_2} : x_{s_1+s_2+j} = \overline{x_{s_1+j}}, \forall j \in [s_2]\}.$$

By identifying elements of K with their canonical embeddings in H, we can speak of geometric *canonical norms* on K. Specifically, we define the ℓ_p canonical norm of x, denoted as $\|x\|_p^{can}$ as $\|\sigma(x)\|_p = \left(\sum_{i\in[n]} |\sigma_i(x)|^p\right)^{\frac{1}{p}}$ for $p < \infty$, and as $\max_{i\in[n]} |\sigma_i(x)|$ for $p = \infty$. (As always we assume the ℓ_2 norm when p is omitted.)

Field Norm. The (field) *norm* of an element $a \in K$ is defined as $\mathsf{N}(a) = \mathsf{N}_{K/\mathbb{Q}}(a) = \prod_{i\in[n]} \sigma_i(a)$.[3] Note that the [Wes99, p 43, proof of Lemma 3.2] norm of an algebraic integer is in $\mathbb{Z}$.

Coefficient Embedding. There is also a *coefficient embedding* $\tau : K \to \mathbb{Q}^n$. As mentioned earlier, since $f(\zeta) = 0$, there is an isomorphism between $\mathbb{Q}[x] \mod f(x)$ and K given by $x \to \zeta$. So, K can be represented as a n-dimensional vector space over $\mathbb{Q}$ using the *power basis* $\{1, \zeta, \ldots, \zeta^{n-1}\}$, and τ maps an element of K to its associated coefficient vector. When identifying an element $a \in K$ as a coefficient vector, i.e., $\tau(a)$ we denote it as a boldface vector $\mathbf{a}$. Note that the addition of vectors is done component-wise, while the multiplication is done as polynomials modulo $f(x)$. We define the *coefficient norm* of a as the norm of the vector $\mathbf{a}$. Specifically, we define the ℓ_p coefficient norm of a, denoted as $\|a\|_p$ or $\|\mathbf{a}\|_p$ as $\left(\sum_{i\in[n]} a_i^p\right)^{\frac{1}{p}}$ for $p < \infty$, and as $\max_{i\in[n]} |a_i|$ for $p = \infty$. (As always we assume the ℓ_2 norm when p is omitted.)

Relationship between Coefficient and Canonical Embeddings. The conversion of an element in $K = \mathbb{Q}[\zeta_m]$ $(n = \phi(m))$ from its coefficient representation to the canonical one can be seen as the multiplication of the coefficients of the polynomial by a specific Vandermonde matrix. More specifically, if $\mathbf{a}$ is an element of K in the coefficient representation then $V_m \cdot \mathbf{a}$ is exactly the canonical representation where $V_m \in \mathbb{C}^{n\times n}$ such that its i^{th} row is the vector $(1, \zeta_m^{j_i}, \zeta_m^{2j_i}, \ldots, \zeta_m^{(n-1)j_i})$ for all $j_i \in \mathbb{Z}_m^*$. The matrix V_m when m is a power of 2 is special in the sense that the matrix $\frac{1}{n}V_m$ is unitary. This means that conversions between the canonical embedding and the coefficient representation corresponds to just a rigid rotation and a scaling.

[3] More generally, the *relative norm* $\mathsf{N}_{K/L}(a)$ of an element $a \in K$ over a subfield $L \subset K$ is $\prod_{\sigma_i\in S} \sigma_i(a)$, where S consists of the K-embeddings σ_i that fix every element in L.

Multiplicative Expansion Factor. We define the *multiplicative expansion factor* γ_{Mult} to denote (as in [Gen09, p. 71]) the maximal value of $\frac{\|\mathbf{a}\times\mathbf{b}\|}{\|\mathbf{a}\|\cdot\|\mathbf{b}\|}$ for any $\boldsymbol{a}, \boldsymbol{b} \in K$. (See [LM06] for a different definition of the expansion factor for multiplication.) The dependence of γ_{Mult} value on the underlying field K is understood.

Next we will argue (also see [Gen09, Lemma 7.4.3] and [GH10, Section 2.2]) that for the field $K = \mathbb{Q}[x]/(x^n + 1)$, γ_{Mult} can be upper bounded by $\sqrt{n}$.

Lemma 2.2.9. *Let $K = \mathbb{Q}[x]/(x^n + 1)$, for any positive integer n. $\forall \boldsymbol{a}, \boldsymbol{b} \in K$ and $\boldsymbol{c} = \boldsymbol{a} \times \boldsymbol{b}$ we have that*

$$\|\boldsymbol{c}\| \leq \sqrt{n} \cdot \|\boldsymbol{a}\| \cdot \|\boldsymbol{b}\|.$$

Proof. Consider the ith coefficient c_i of $\boldsymbol{c}$. First observe that for each i, c_i is obtained as a dot product of $\boldsymbol{a}$ and some reordering of entries of $\boldsymbol{b}$ (additionally the signs of some entries can also be reversed). Therefore we have $c_i \leq \|\boldsymbol{a}\| \cdot \|\boldsymbol{b}\|$. This allows us to conclude that $\|\boldsymbol{c}\| \leq \sqrt{n} \cdot \|\boldsymbol{a}\| \cdot \|\boldsymbol{b}\|$. □

Example. Continuing with our example of the mth cyclotomic number field where $K = \mathbb{Q}(\zeta_m)$ for $m > 2$, there are $2s_2 = n = \varphi(m)$ complex canonical embeddings (and no real ones), which are given by $\sigma_i(\zeta_m) = \zeta_m^i$ for $i \in \mathbb{Z}_m^*$. (It is convenient to index the embeddings by elements of $\mathbb{Z}_m^*$ instead of $[n]$.) For an element $x = \zeta^j \in K$ in the power basis of K, all the embeddings of x have magnitude 1, and hence $\|x\|_2^{can} = \sqrt{n}$ and $\|x\|_\infty^{can} = 1$. Also considering the coefficient embedding $\|x\|_2 = 1$.

2.2.3 Ideals in the Ring of Integers

The ring of integers $\mathcal{O}_K$, of a number field K of degree n, is a free $\mathbb{Z}$-module (see [Wes99, p 39, Theorem 2.22]) of rank n, i.e., the set of all $\mathbb{Z}$-linear combinations of some *integral basis* $\{b_1, \ldots, b_n\} \subset \mathcal{O}_K$. Such a set is called an *integral basis*, and it is also a $\mathbb{Q}$-basis for K. As usual, there are infinitely many such bases when $n > 1$.

Continuing with our example of the mth cyclotomic number field $K = \mathbb{Q}(\zeta_m)$ of degree $n = \varphi(m)$, the power basis $\{1, \zeta_m, \ldots, \zeta_m^{n-1}\}$ of K also happens to be an integral basis of the *cyclotomic ring* $\mathcal{O}_K = \mathbb{Z}[\zeta_m]$. (In general, it is unusual for the power basis of a number field to generate the entire ring of integers.)

Definition 2.2.10 (Ideal)**.** *An* (integral) ideal *$\mathcal{I} \subseteq \mathcal{O}_K$ is a nontrivial (i.e., nonempty and nonzero[4]) additive subgroup that is closed under multiplication by $\mathcal{O}_K$ – that is, $r \cdot g \in \mathcal{I}$ for any $r \in \mathcal{O}_K$ and $g \in \mathcal{I}$. A* fractional ideal *$\mathcal{I} \subset K$ is a set such that $d \cdot \mathcal{I}$ is an integral ideal for some $d \in \mathcal{O}_K$. The* inverse *$\mathcal{I}^{-1}$ of an ideal $\mathcal{I}$ is the set $\{a \in K : a \cdot \mathcal{I} \subseteq \mathcal{O}_K\}$.*

An ideal $\mathcal{I}$ in $\mathcal{O}_K$ is finitely generated as the set of all K-linear combinations of some *generators* $g_1, g_2, \ldots \in \mathcal{O}_K$, denoted $\mathcal{I} = \langle g_1, g_2, \ldots \rangle$. (In fact, it is know that two generators [Ste04, Proposition 9.1.7] always suffice.)

Definition 2.2.11. *An ideal $\mathcal{I}$ is* principal *if $\mathcal{I} = \langle g \rangle$ for $g \in \mathcal{O}_K$ – that is, if one generator suffices.*

More useful to us is the fact [Oss08, Proposition 1.6.1] that an ideal (integral or fractional) is also a free $\mathbb{Z}$-module of rank n, i.e., it is generated as the set of all $\mathbb{Z}$-linear combinations of some basis $\{b_1, \ldots, b_n\} \subset \mathcal{O}_K$.

[4] Some texts also define the trivial set $\{0\}$ as an ideal, but in this work it is more convenient to exclude it.

Definition 2.2.12. *Let $\mathcal{I}, \mathcal{J}$ be ideal of a ring R. Their* sum *is the ideal*

$$\mathcal{I} + \mathcal{J} = \{a + b : a \in \mathcal{I}, b \in \mathcal{J}\}$$

and their product $\mathcal{I}\mathcal{J}$ *is ideal generated by all products of elements in $\mathcal{I}$ with elements in $\mathcal{J}$, or*

$$\mathcal{I}\mathcal{J} = \langle a \cdot b : a \in \mathcal{I}, b \in \mathcal{J}\rangle.$$

Two ideals $\mathcal{I}, \mathcal{J} \subseteq \mathcal{O}_K$ are said to be *coprime* (or *relatively prime*) if $\mathcal{I} + \mathcal{J} = \mathcal{O}_K$.

2.2.4 Prime Ideals - Unique Factorization and Distributions

In this section we will define the notion of prime ideals and recall some of their properties. A prime ideal shares many important properties of a prime number in $\mathbb{Z}$.

Definition 2.2.13. *An ideal $\mathfrak{p} \subsetneq \mathcal{O}_K$ is* prime *if whenever $a, b \in \mathcal{O}_K$ and $ab \in \mathfrak{p}$ then either $a \in \mathfrak{p}$ or $b \in \mathfrak{p}$.*

Unique Factorization. As per unique-prime-factorization theorem, we have that every integer greater than 1 is either prime itself or is the product of prime numbers. Similar in any ring of integers $\mathcal{O}_K$ of the number field K has unique factorization of ideals into prime ideals.

Proposition 2.2.14 (Unique Factorization of Ideals [Ste04, Theorem 6.1.9])**.** *Suppose $\mathcal{I}$ is an integral ideal of $\mathcal{O}_K$. Then $\mathcal{I}$ can be written as a product*

$$\mathcal{I} = \mathfrak{p}_1 \dots \mathfrak{p}_n$$

of prime ideals of $\mathcal{O}_K$, and this representation is unique up to order.

Ideal Norm and some of its properties. Now we will define the norm of an ideal and mention some of the properties about the norms of prime ideals.

Definition 2.2.15. *If $\mathcal{I}$ is an ideal of a ring of integers $\mathcal{O}_K$, we define the* norm *of $\mathcal{I}$ to be*

$$\mathsf{N}(\mathcal{I}) = |\mathcal{O}_K/\mathcal{I}|$$

where $|\mathcal{O}_K/\mathcal{I}|$ dentes the size of the quotient ring $\mathcal{O}_K/\mathcal{I}$.

It is know that [Wes99, p 60, Lemma 2.2] $\mathsf{N}(\mathcal{I}\mathcal{J}) = \mathsf{N}(\mathcal{I})\mathsf{N}(\mathcal{J})$.

In $\mathcal{O}_K$, an [Ste04, Proposition 6.1.4] ideal $\mathfrak{p}$ is prime if and only if it is *maximal*, i.e., if the only proper superideal of $\mathfrak{p}$ is $\mathcal{O}_K$ itself, which implies that the quotient ring $\mathcal{O}_K/\mathfrak{p}$ is a finite field of order $\mathsf{N}(\mathfrak{p})$.

Proposition 2.2.16 ([Oss08, Corollary 1.6.9])**.** *For a in a ring of integers $\mathcal{O}_K$, let $\mathfrak{p} = \langle a \rangle$ be the principal ideal generated by a, then we have that $\mathsf{N}(\mathcal{I}) = |\mathsf{N}(a)|$.*

Suppose $\mathfrak{p}$ is an ideal of a ring of integers $\mathcal{O}_K$, and $\mathsf{N}(\mathfrak{p}) = p$ for some prime integer $p \in \mathbb{Z}$. Then we have that [Oss08, Lemma 1.6.7] $\mathfrak{p}$ is prime in $\mathcal{O}_K$. Note that, many prime ideals do not have prime norms. In fact [Oss08, Lemma 4.6.1] if $\mathfrak{p}$ is a prime ideal in a ring of integers $\mathcal{O}_K$, then $\mathsf{N}(\mathfrak{p}) = p^n$ for some prime $p \in \mathbb{Z}$ and $n \in \mathbb{N}$.

Distribution of Prime ideals. The distribution of prime ideals in number fields is quite analogous to the distribution of primes in the integers. Just as the prime number theorem states that the number of primes less than x is approximately $x/\ln x$, we have Landau's prime ideal theorem.

Theorem 2.2.17 (Landau's prime number theorem [BS96, Theorem 8.7.2])**.** *Let K be an algebraic number field of degree n. Let $\pi_K(x)$ denote the number of prime ideals whose norm is $\leq x$. Let $\xi(x) = (\ln x)^{3/5}(\ln\ln x)^{-1/5}$. There is a $c > 0$ (depending on K) such that*

$$\pi_K(x) = Li(x) + O(xe^{-c\xi(x)}) \sim \frac{x}{\ln x},$$

where $Li(x) = \int_2^t \frac{dt}{\ln t}$.

Furthermore the prime ideals in the above theorem are dominated by the ideals of norm a prime number. Assuming the Generalized Riemann Hypothesis (GRH) [BS96, Conjecture 8.7.3], a stronger statement [BS96, Theorem 8.7.4] can be made but the above mentioned unconditional statement suffices for our purposes.

In our constructions we will need results on the distribution of prime ideals that are also principal. From prime number theorem for arithmetic progressions, we know that the number of primes less that or equal to x and congruent to $a \mod n$ (where a and n are co-prime), is $x/(\phi(n)\ln x)$. Similarly one of the consequences of Chebotarëv's density[5] theorem (see for example [Ste10, Proof of Lemma 4]) is that the among all the prime ideals in a number field K, $\frac{1}{h}$ of them are principal, where h is the class number of K.

We refer the reader to [Lan90, p 77] for a general analytic formula for computing the class number of any number field K. The class number[6] of the n-th cyclotomic field K, factors as h^+ times h^-, where h^+ is the class number of the maximal real subfield of K. The Brauer-Siegel theorem (see [Was97, Theorem 4.20]) implies that $\log(h^-)$ grows roughly as $\frac{1}{4}\phi(n)\log n$ as $n \to \infty$. However, h^+ tends to be rather small. For n a power of 2, it is conjectured that $h^+ = 1$. This is true for $n = 2^k$ with $k \leq 7$, and also for $k = 8$ if we assume GRH. This provides for theoretical evidence that principal prime ideals exist. However since the class number is already exponential this does not suffice for our purposes.

Nevertheless restricting the Landau's prime number theorem to principal ideals we can heuristically expect that with noticeable probability a random principal ideal will have a prime norm.

Conjecture 2.2.18. *Let K be the n-th cyclotomic field for n a power of 2. For every $\sigma = \mathsf{poly}(n)$ there is a constant $c > 1$ such that for sufficiently large n we have that*

$$\Pr_{f \leftarrow D_{Z^n,\sigma}}[\mathsf{N}(f) \geq 2^{O(n)} \text{ and is prime}] \geq \frac{1}{n^c}.$$

Smart and Vercauteren [SV10] and Boneh and Freeman [BF11a] follow a similar heuristic in their applications. Experimental results supporting this heuristic have been provided by Smart and Vercauteren [SV10].

[5] Just like Landau's prime number theorem is a generalization of the prime number theorem, we have the Chebotarëv's density theorem [BS96, Theorem 8.7.9] with generalizes the prime number theorem for arithmetic progressions [BS96, Theorem 8.4.2] to number fields. Chebotarëv's density theorem is a very technical result building on field theory and we do not delve into stating it formally. We refer the reader to [SL96] for a very down to earth introduction to Chebotarëv's Density Theorem.

[6] We would like to thank Alice Silverberg and Lawrence Washington for pointing [SW13] these facts about class number of cyclotomic fields to us.

2.2.5 Ideal Lattices

Recall that a number field $K = Q(\zeta)$ is isomorphic to $\mathbb{Q}[x]/f(x)$ where $f(x)$ is the minimal polynomial of ζ. Also recall that any ideal $\mathcal{I}$ of $\mathcal{O}_K$ is a free $\mathbb{Z}$-module, i.e., it is generated as the set of all $\mathbb{Z}$-linear combinations of some basis $B = \{b_1, \ldots, b_n\} \subset \mathcal{O}_K$. Therefore under the coefficient embedding τ, the ideal $\mathcal{I}$ of $\mathcal{O}_K$ yields a rank-n lattice $\tau(\mathcal{I})$ having basis $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n\}$, where each $\boldsymbol{b}_i = \tau(b_i)$. Obviously, addition is done component-wise in the coefficients, and multiplication is polynomial multiplication modulo the polynomial $f(x)$. We call $\mathcal{I}$ an *ideal lattice* to stress its dual interpretation as both an ideal and a lattice. When visualizing it as a lattice we speak of, e.g., the minimum distance $\lambda_1(\mathcal{I})$ of an ideal, etc.

As pointed out earlier the mth cyclotomic ring with $n = \varphi(m)$ happens to be exactly $\mathbb{Z}[\zeta_m]$ which corresponds to the lattice $\mathbb{Z}^n$.

Proposition 2.2.19 ([LPR12, p 22])**.** *For any ideal $\mathcal{I}$ of the mth cyclotomic ring (with $n = \varphi(m)$) we have $\lambda_n(\mathcal{I}) = \lambda_1(\mathcal{I})$.*

We will sketch the argument here. Consider the mth cyclotomic field such that $n = \varphi(m)$. Observe that multiplying a shortest nonzero element $\boldsymbol{v} \in \mathcal{I}$ by $1, \zeta, \ldots, \zeta^{n-1}$ gives n linearly independent elements of the same length. This allows us to conclude the above proposition.

Invertibility of ring elements. Let R denote the $2n^{th}$ cyclotomic ring and let R_q denote R/qR for a prime q. We note that R_q is also a ring and not all elements in it are invertible. Let $R_q^\times$ denote the set of elements in R_q that are invertible. We next provide a lemma of Stehlé and Steinfeld that points out that a (large enough) random element is R_q is also in $R_q^\times$ with large probability.

Lemma 2.2.20 ([SS11, Lemma 4.1])**.** *Let $n \geq 8$ be a power of 2 such that $x^n + 1$ splits into n linear factors modulo $q \geq 5$. Let $\sigma \geq \sqrt{n \ln(2n(1 + 1/\delta))/\pi} \cdot q^{1/n}$, for an arbitrary $\delta \in (0, 1/2)$. Then*

$$\Pr_{f \leftarrow D_{Z^n,\sigma}} [f \mod q \notin R_q^\times] \leq n(1/q + 2\delta).$$

2.3 GGH Multilinear Maps

In this section we describe the GGH multilinear maps for the "symmetric setting" Later we explain how to handle the "asymmetric setting," which will be needed for our obfuscation construction. We provide only those details that are essential for understanding the obfuscation construction.

An instance of the GGH construction is parameterized by the security parameter λ and the required multi-linearity level $\kappa \leq$poly(λ). Based on these parameters, choose the $2n$th cyclotomic ring $R = \mathbb{Z}[x]/(x^n + 1)$ where n is a power of 2 (n is set large enough to ensure security), and a modulus q that defines $R_q = R/qR$ (with q large enough to support functionality). The specific constraints that these parameters must satisfy are discussed in Section 2.3.2, an approximate setting to keep in mind is $n = \tilde{O}(\kappa\lambda^2)$, and $q = 2^{\kappa\lambda}$.

2.3.1 The Encoding Scheme

We start by giving the intuition behind the scheme. An instance of the GGH scheme relative to the parameters above encodes elements of a quotient ring $QR = R/\mathcal{I}$, where $\mathcal{I}$ is a principal prime ideal $\mathcal{I} = \langle \mathbf{g} \rangle \subset R$, generated by a "short" vector $\mathbf{g}$. Namely, the "ring elements" that are encoded in our scheme are cosets of the form $\boldsymbol{e} + \mathcal{I}$ for some vector $\boldsymbol{e}$. The short generator $\mathbf{g}$ itself is kept secret, and no "good" description of $\mathcal{I}$ is made public in our scheme. In addition, GGH system depends on another secret element $\mathbf{z}$, which is chosen at random in R_q (and hence is not short).

A level-zero ("plaintext") encoding of a coset $\boldsymbol{e} + \mathcal{I} \in R/\mathcal{I}$ is just a short vector in that coset (which must exist, since the generator $\mathbf{g}$ is short and therefore the basic cell of $\mathcal{I}$ is quite small). For higher-level encodings, a level-i encoding of the same coset is a vector of the form $\boldsymbol{c}/\mathbf{z}^i \in R_q$ with $\boldsymbol{c} \in \boldsymbol{e} + \mathcal{I}$ short. Specifically, for $i \in \{0, 1, \ldots, \kappa\}$ the set of all level-i encodings is $S_i = \{\boldsymbol{c}/\mathbf{z}^i \in R_q : \|\boldsymbol{c}\| < q^{1/8}\}$, and the set of level-$i$ encodings of the "plaintext element" $\boldsymbol{e} + \mathcal{I}$ is $S_i^{(\boldsymbol{e}+\mathcal{I})} = \{\boldsymbol{c}/\mathbf{z}^i \in R_q : \boldsymbol{c} \in \boldsymbol{e} + \mathcal{I},\ \|\boldsymbol{c}\| < q^{1/8}\ \}$. Throughout the construction we use the size of the numerator as the "noise level" in the encoding. Namely, with each level-i encoding $\boldsymbol{c}/\mathbf{z}^i$ we produce also an upper bound on $\|\boldsymbol{c}\|$.

Instance generation: $(\mathsf{params}, \mathbf{p}_{zt}) \leftarrow \mathsf{InstGen}(1^\lambda, 1^\kappa)$. GGH instance-generation procedure chooses at random the ideal-generator $\mathbf{g}$ and denominator $\mathbf{z}$, as well as several other vectors that are used in the other procedures and are described later in the section. The denominator $\mathbf{z}$ is chosen uniformly at random in R_q, and hence is not "small" with overwhelming probability. Using Lemma 2.2.20 we have that $\mathbf{z}$ is invertible in R_q with overwhelming probability.

Simply draw $\mathbf{g}$ from a discrete Gaussian over $\mathbb{Z}^n$, say $\mathbf{g} \leftarrow D_{\mathbb{Z}^n,\sigma}$ with $\sigma = \sqrt{\lambda n}$ repeatedly till:

(i) $\|\mathbf{g}\| \le \sigma\sqrt{n}$ and $\mathbf{g}$ is invertible in R_q.

(ii) $\|\mathbf{g}^{-1}\| \le n^{c+1.5}$ (in K) for an appropriate constant c. (Recall that we denote $K = \mathbb{Q}[x]/(x^n + 1)$. The reason that we need $\mathbf{g}^{-1} \in K$ to be short is explained when we describe the zero-testing procedure.)

(iii) $\mathsf{N}(\mathbf{g})$ is a prime $\ge 2^{O(n)}$.

From Lemma 2.3.1 we can conclude that the above described rejection sampling procedure succeeds in polynomially many trials. Condition (iii) from above, Proposition 2.2.16 and the discussion there after imply that $\mathcal{I} = \langle \mathbf{g} \rangle$ is a principal prime ideal.

An element $\mathbf{p}_{zt}$ that is used as a zero-testing parameter is also generated. This generation process is described later. The instance-generation procedure outputs $\mathsf{params} = (n, q, \mathbf{g}, \mathbf{z})$ and $\mathbf{p}_{zt}$.

Lemma 2.3.1. *If $\mathbf{g} \leftarrow D_{\mathbb{Z}^n,\sigma}$, then assuming Conjecture 2.2.18 there exists a constant c such that* (i), (ii) *and* (iii) *from above are simultaneously satisfied with a noticeable probability.*

Proof. We will proceed by obtaining bounds on probabilities that each of the above conditions (i), (ii) and (iii) individually holds. Subsequently the lemma follows by a union bound argument.

(i) It follows directly from Lemma 2.1.5 and Lemma 2.2.20 that condition (i) is satisfied with overwhelming probability.

(ii) Now we argue that with good probability $\mathbf{g}^{-1}$ in the field of fractions K is also rather short. We will argue this by looking at $\mathbf{g}$ in terms of the canonical embedding. As pointed in Section 2.2.2, the canonical embedding representation can be obtained by multiplying the coefficient representation with the matrix V_{2n}. And this transformation for a power of 2 cyclotomic corresponds to just a rigid rotation and a scaling (thus the spherical Gaussian distribution is not affected by the transformation). Therefore we have that sampling $\mathbf{g}$ from $D_{\mathbb{Z}^n,\sigma}$ and considering the canonical embedding is the same as sampling directly the canonical representation for an appropriately scaled gaussian parameter σ', which in our case is at least $\omega(1)$. This implies that roughly with probability $1 - o(1/n^{c+1})$, evaluating $\mathbf{g}$ at any complex $2n$'th root of unity $\zeta \in \mathbb{C}$ yields $\mathbf{g}(\zeta)$ which is greater than $1/n^{c+1}$.

Next by taking a union bound, with probability $1 - o(1/n^c)$ we have $\mathbf{g}^{-1}(\zeta) = 1/\mathbf{g}(\zeta) < n^{c+1}$ for all the primitive $2n$'th roots of unity ζ, which means that $\|\mathbf{g}^{-1}\|_\infty^{can} < n^{c+1}$. This implies an upper bound of $\|\mathbf{g}^{-1}\|_\infty < n^{c+1}$ as well (because for every $\boldsymbol{a} \in K$ we have that $\|\boldsymbol{a}\|_\infty \leq \|\boldsymbol{a}\|_\infty^{can}$; see for example [DPSZ11, Theorem 7 and Discussion on p. 39] for a detailed proof). Hence a bound of $\|\mathbf{g}^{-1}\| < n^{c+1.5}$.

(iii) Conjecture 2.2.18 implies that there exists a constant c such that condition (iii) is satisfied with probability at least $\frac{1}{n^c}$.

Putting the above bounds together and taking a union bound implies the claimed lemma. □

Sampling level-zero encodings: $\boldsymbol{d} \leftarrow \mathsf{samp}(\mathsf{params})$. To sample a level-zero encoding of a random coset, just draw a random short element in R, $\boldsymbol{d} \leftarrow D_{\mathbb{Z}^n,\sigma'}$, where $\sigma' = \sigma n\sqrt{\lambda}$ (for σ that was used to sample $\mathbf{g}$). In Lemma 2.3.2 we argue that the sampled value $\boldsymbol{d}$ corresponds to a random coset of $\mathcal{I}$. Finally note that by Lemma 2.1.5 the size of this level-zero encoding is bounded by $\sigma'\sqrt{n}$ (and we use this as our noise-bound for this encoding).

Lemma 2.3.2. *Let $\mathcal{I} = \langle \mathbf{g} \rangle$ and $\sigma' \geq \sqrt{\lambda n}\|\mathbf{g}\|$, then we have that the distribution $\boldsymbol{d} \mod \mathcal{I}$ where $\boldsymbol{d} \leftarrow D_{\mathbb{Z}^n,\sigma'}$ is close to uniform over $\mathbb{Z}^n \mod \mathcal{I}$, up to negligible distance.*

Proof. We can safely assume that $\lambda_1(\mathcal{I}) \leq \|\mathbf{g}\|$. Next according to Proposition 2.2.19 we have that $\lambda_n(\mathcal{I}) = \lambda_1(\mathcal{I})$. This along with Lemma 2.1.2 allows us to conclude that with overwhelming probability

$$\begin{aligned} \eta_{2^{-\lambda}}(\mathcal{I}) &\leq \sqrt{\frac{\ln(2n(1+1/\epsilon))}{\pi}} \cdot \|\mathbf{g}\| \\ &\leq \sqrt{\frac{\ln(2n(1+1/\epsilon))}{\pi}} \cdot \|\mathbf{g}\| \\ &\leq \sqrt{\lambda n}\|\mathbf{g}\| \end{aligned}$$

Finally since we have that $\sigma' \geq \eta_{2^{-\lambda}}(\mathcal{I})$, therefore by Corollary 2.1.4 we can conclude that the induced distribution over the cosets of $\mathcal{I}$ is close to uniform, up to a negligible distance. □

Encodings at higher levels: $\boldsymbol{u}_i \leftarrow \mathsf{enc}(\mathsf{params}, i, \boldsymbol{d})$. To allow encoding of cosets at higher levels, draw $\boldsymbol{d} \leftarrow D_{\mathbb{Z}^n,\sigma'}$, for some parameter σ', then compute the encoding as $[\boldsymbol{d}/\mathbf{z}^i]_q$ (A more secure version of sampling is described in [GGH13a]. We use the notation $[\cdot]_q$ to denote operations in R_q.)

Adding and multiplying encodings. It is easy to see that the encoding as above is additively homomorphic, in the sense that adding encodings yields an encoding of the sum. This follows since if we have many short $\boldsymbol{c}_j$'s then their sum is still short, $\|\sum_j \boldsymbol{c}_j\| \ll q$, and therefore the sum $\boldsymbol{c} = \sum_j \boldsymbol{c}_j = [\sum_j \boldsymbol{c}_j]_q \in R_q$ belong to the coset $\sum_j(\boldsymbol{c}_j + \mathcal{I})$. Hence, if we denote $\boldsymbol{u}_j = \boldsymbol{c}_j/\mathbf{z} \in R_q$ then each $\boldsymbol{u}_j$ is an encoding of the coset $\boldsymbol{c}_j + \mathcal{I}$, and the sum $[\sum_j \boldsymbol{u}_j]_q$ is of the form $\boldsymbol{c}/\mathbf{z}$ where $\boldsymbol{c}$ is still a short element in the sum of the cosets.

Moreover, since $\mathcal{I}$ is an ideal then multiplying upto κ encodings can be interpreted as an encoding of the product, by raising the denominator to the appropriate power. Namely, for $\boldsymbol{u}_j = \boldsymbol{c}_j/\mathbf{z} \in R_q$ as above, we have

$$\boldsymbol{u} = \left[\prod_{j=1}^{\kappa} \boldsymbol{u}_j\right]_q = \left[\frac{\prod_j \boldsymbol{c}_j}{\mathbf{z}^\kappa}\right]_q.$$

As long as the $\boldsymbol{c}_j$'s are small enough to begin with, we still have $\| \prod_j \boldsymbol{c}_j \| \ll q$, which means that $[\prod_j \boldsymbol{c}_j]_q = \prod_j \boldsymbol{c}_j$ (where the product $\prod_j \boldsymbol{c}_j$ is computed in R), hence $[\prod_j \boldsymbol{c}_j]_q$ belongs to the product coset $\prod_j (\boldsymbol{c}_j + \mathcal{I})$.

Thus, if each $\boldsymbol{u}_j$ is a level-1 encoding of the coset $\boldsymbol{c}_j + \mathcal{I}$ with short-enough numerator, then their product is a level-κ encoding of the product coset. We note that just like level-1 encoding, level-κ encoding (and in fact any of the intermediate level encoding) also offers additive homomorphism.

Zero testing: $\mathsf{isZero}(\mathsf{params}, \mathbf{p}_{zt}, \boldsymbol{u}_\kappa) \overset{?}{=} 0/1$**.** Since the encoding is additively homomorphic, GGH can test equality between encodings by subtracting them and comparing to zero. To enable zero-testing, GGH generate the zero-testing parameter as follows: Draw a "somewhat small" ring element $\boldsymbol{h} \leftarrow D_{\mathbb{Z}^n, \sqrt{q}}$, such that $\boldsymbol{h} \notin \mathcal{I}$ and set the zero-testing parameter as $\mathbf{p}_{zt} = [\boldsymbol{h}\mathbf{z}^\kappa / \mathbf{g}]_q$. (A more secure version of sampling is described in [GGH13a].) To test if a level-κ encoding $\boldsymbol{u} = [\boldsymbol{c}/\mathbf{z}^\kappa]_q$ is an encoding of zero, we just multiply it in R_q by $\mathbf{p}_{zt}$ and check whether the resulting element $\boldsymbol{w} = [\mathbf{p}_{zt} \cdot \boldsymbol{u}]_q$ is short (e.g., shorter than $q^{3/4}$). Namely, we use the test

$$\mathsf{isZero}(\mathsf{params}, \mathbf{p}_{zt}, \boldsymbol{u}) \;=\; \begin{cases} 1 & \text{if } \|[\mathbf{p}_{zt}\boldsymbol{u}]_q\|_\infty < q^{3/4} \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

In Lemma 2.3.3 we will argue that encodings of zero (such that the numerator is less than $q^{1/8}$) always pass the zero test. Next in Lemma 2.3.5 we argue that encodings of non-zero cosets pass the zero test only with a negligible probability.

Lemma 2.3.3. *For any $\boldsymbol{u} = [\boldsymbol{c}/\mathbf{z}^\kappa]_q$ such that $\|\boldsymbol{c}\| < q^{1/8}$ and $\boldsymbol{c} \in \mathcal{I} = \langle \mathbf{g} \rangle$, such that $\|\mathbf{g}^{-1}\| < \frac{q^{1/8}}{n^{3/2}}$ (in K) we have that $\|[\mathbf{p}_{zt}\boldsymbol{u}]_q\|_\infty < q^{3/4}$ where $\boldsymbol{h} \leftarrow D_{\mathbb{Z}^n, \sqrt{q}}$, and $\mathbf{p}_{zt} = [\boldsymbol{h}\mathbf{z}^\kappa / \mathbf{g}]_q$.*

Proof. To see why this works, we note that

$$\boldsymbol{w} \;=\; \mathbf{p}_{zt} \cdot \boldsymbol{u} \;=\; \frac{\boldsymbol{h}\mathbf{z}^\kappa}{\mathbf{g}} \cdot \frac{\boldsymbol{c}}{\mathbf{z}^\kappa} \;=\; \boldsymbol{h} \cdot \boldsymbol{c}/\mathbf{g} \;\; \text{(all the operations in } R_q\text{)}.$$

If $\boldsymbol{u}$ is an encoding of zero then $\boldsymbol{c}$ is a short vector in $\mathcal{I}$ (containing elements $\mathbf{g}\boldsymbol{r}$ for $\boldsymbol{r} \in R$), which means that it is divisible by $\mathbf{g}$ in R. Hence the element $\boldsymbol{c}/\mathbf{g} \in R$ is the same as the element $\boldsymbol{c} \cdot \mathbf{g}^{-1} \in K$. Next we have that $\boldsymbol{c} \cdot \mathbf{g}^{-1}$ is at most $\|\boldsymbol{c}\| \cdot \|\mathbf{g}^{-1}\| \cdot \gamma_{\mathsf{Mult}}$ (recall that using Lemma 2.2.9 γ_{Mult} can be bounded $\sqrt{n}$). Next we have that $\|\boldsymbol{w}\| \le \|\boldsymbol{h}\| \cdot \|\boldsymbol{c}\| \cdot \|\mathbf{g}^{-1}\| \cdot \gamma^2_{\mathsf{Mult}}$, which for our choice of parameter is $q^{1/2} \cdot \sqrt{n} \cdot q^{1/8} \cdot \|\mathbf{g}^{-1}\| \cdot n < q^{3/4}$ (Note that by Lemma 2.1.5 we have that $\|\boldsymbol{h}\| \le q^{1/2} \cdot \sqrt{n}$ with overwhelming probability). This immediately also gives an upper bound on the ℓ_∞ norm of $\boldsymbol{w}$. □

If $\boldsymbol{u}$ is an encoding of a non-zero coset, then $\boldsymbol{c}$ is a short vector in some coset of $\mathcal{I}$. In this case we have $\boldsymbol{w} = [\boldsymbol{c} \cdot \boldsymbol{h}/\mathbf{g}]_q$, where $\boldsymbol{c}, \mathbf{g}$ are small (and $\boldsymbol{h}$ is "somewhat small"). Intuitively, since $[\boldsymbol{h}/\mathbf{g}]_q$ is large with high probability then for a "random enough" $\boldsymbol{c}$ we expect the size of $\boldsymbol{w}$ to be large. More formally, we argue below (Lemma 2.3.4) that when choosing a uniformly random coset of $\mathcal{I} = \langle \mathbf{g} \rangle$, there are *no short elements* $\boldsymbol{c}$ in that coset such that $[\boldsymbol{c} \cdot \boldsymbol{h}/\mathbf{g}]_q$ is small. This will allow up to conclude Lemma 2.3.5.

Lemma 2.3.4. *Let $\boldsymbol{w} = [\boldsymbol{c} \cdot \boldsymbol{h}/\mathbf{g}]_q$ and suppose $\|\mathbf{g} \cdot \boldsymbol{w}\|$ and $\|\boldsymbol{c} \cdot \boldsymbol{h}\|$ are each at most $q/2$. Suppose $\langle \mathbf{g} \rangle$ is a prime ideal. Then, either $\mathbf{c}$ or $\boldsymbol{h}$ is in the ideal $\langle \mathbf{g} \rangle$.*

Proof. Since $\mathbf{g} \cdot \boldsymbol{w} = \boldsymbol{c} \cdot \boldsymbol{h} \bmod q$, and since $\|\mathbf{g} \cdot \boldsymbol{w}\|$ and $\|\boldsymbol{c} \cdot \boldsymbol{h}\|$ are each at most $q/2$, we have $\mathbf{g} \cdot \boldsymbol{w} = \boldsymbol{c} \cdot \boldsymbol{h}$ exactly. We also have an equality of ideals $\langle \mathbf{g} \rangle \cdot \langle \boldsymbol{w} \rangle = \langle \boldsymbol{c} \rangle \cdot \langle \boldsymbol{h} \rangle$, and, since $\langle \mathbf{g} \rangle$ is a prime ideal and our cyclotomic ring is a unique factorization domain (see Proposition 2.2.14), we have that $\langle \mathbf{g} \rangle$ divides either $\langle \boldsymbol{c} \rangle$ or $\langle \boldsymbol{h} \rangle$ (or both). The result follows. □

Lemma 2.3.5. *Let $q = n^{\omega(1)}$, and $\langle \mathbf{g} \rangle$ be a prime ideal such that $\|\mathbf{g}\| = \mathsf{poly}(n)$. Sample $\boldsymbol{h} \leftarrow D_{\mathbb{Z}^n,\sqrt{q}}$ such that $\boldsymbol{h} \notin \langle \mathbf{g} \rangle$. Then, there is no $\epsilon > 0$ and $\mathbf{c} \notin \mathcal{I}$ such that $\|\mathbf{c}\| < q^{1/8}$ and $\|[\mathbf{c} \cdot \boldsymbol{h}/\mathbf{g}]_q\| < q^{1-\epsilon}$.*

Proof. We are give than $\|c\| < q^{1/8}$ and have $\|h\| < \sqrt{q \cdot n}$ (with overwhelming probability using Lemma 2.1.5). Hence, using Lemma 2.2.9 we have that $\|\boldsymbol{c} \cdot \boldsymbol{h}\| < q^{1/8+1/2} \cdot n < q/2$. Also for the sake of contradiction assume that that $\boldsymbol{w} = [\boldsymbol{c} \cdot \boldsymbol{h}/\mathbf{g}]_q$ is such that $\|\boldsymbol{w}\| < q^{1-\epsilon}$. Then again we have that $\|\boldsymbol{w} \cdot \mathbf{g}\| < q^{1-\epsilon} \cdot \|\mathbf{g}\|\sqrt{n} < q/2$ as $\|\mathbf{g}\| = \mathsf{poly}(n)$ and $q = n^{\omega(1)}$. Now using Lemma 2.3.4, we have that either $\boldsymbol{c}$ or $\boldsymbol{h}$ is in the ideal $\langle \mathbf{g} \rangle$, which is a contradiction. □

2.3.2 Setting the parameters

In this section we provide the parameters for the basic setting that should be set so that all the constraints required by the scheme are met. A overview is presented in Table 2.3.2.

Parameter	Constraints	Value Set
σ	By Lemma 2.3.1, $\|\mathbf{g}\| \leq \sigma\sqrt{n}$, $\|\mathbf{g}^{-1}\| \leq n^{c+1.5}$.	$\sqrt{n\lambda}$
σ'	By Lemma 2.3.2, $\sigma' \geq \sqrt{n\lambda} \cdot \|\mathbf{g}\|$.	$\lambda n^{3/2}$
q	Multiplication of κ encoding should have small numerator. By Lemma 2.3.5, $q > n^{\omega(1)}$. By Lemma 2.3.3, $\|\mathbf{g}^{-1}\| < \frac{q^{1/8}}{n^{3/2}}$.	$q \geq 2^{8\kappa\lambda} n^{O(\kappa)}$

Table 2.1: Parameters for our graded encoding scheme.

- The basic Gaussian parameter σ that we use to draw the ideal generator, $\mathbf{g} \leftarrow D_{\mathbb{Z}^n,\sigma}$, needs to be set to satisfy $\sigma \geq \eta_{2^{-\lambda}}(\mathbb{Z}^n)$, which means that we have $\sigma = \sqrt{\lambda n}$. Then as argued in Lemma 2.3.1 we have that the size of $\mathbf{g}$ is bounded with overwhelming probability by $\|\mathbf{g}\| \leq \sigma\sqrt{n} = n\sqrt{\lambda}$.
- Once we have the ideal lattice $\mathcal{I} = \langle \mathbf{g} \rangle$, the Gaussian parameter σ' by Lemma 2.3.2 we should have $\sigma' \geq \|\mathbf{g}\|\sqrt{\lambda n}$. Given the bound from above bound on the size of $\mathbf{g}$, it is sufficient to set $\sigma' = \lambda n^{3/2}$, which means that the size of level-zero elements is bounded with overwhelming probability by λn^2.
- A level-κ encoding is obtained by multiplying κ level-one encodings (which will always be re-randomized). Hence it is of the form $[\boldsymbol{c}/\mathbf{z}^\kappa]_q$ with $\boldsymbol{c}$ of size bounded with high probability by $\|\boldsymbol{c}\| \leq (2^\lambda \cdot \mathrm{poly}(n))^\kappa = 2^{\kappa\lambda} \cdot n^{O(\kappa)}$. To use Lemma 2.3.5 for level-κ encodings, we need $\|\boldsymbol{c}\| \leq q^{1/8}$, so it is sufficient to set $q \geq 2^{8\kappa\lambda} \cdot n^{O(\kappa)}$. With this choice the constraints from Lemma 2.3.5 ($q > n^{\omega(1)}$) and Lemma 2.3.3 ($\|\mathbf{g}^{-1}\| < \frac{q^{1/8}}{n^{3/2}}$) are easily satisfied.
- Finally, in order to get λ-level security against lattice attacks, we roughly need to set the dimension n large enough so that $q < 2^{n/\lambda}$, which means that $n > \tilde{O}(\kappa\lambda^2)$.

2.3.3 Asymmetric Variant

We now describe the asymmetric variant of multilinear maps. In this variant we still choose just one ideal generator $\mathbf{g}$, but several different denominators $\mathbf{z}_j \overset{r}{\leftarrow} R_q$, $j = 1, \ldots, \tau$. Then, a vector of the form $\boldsymbol{c}/\mathbf{z}_j \in R_q$ with $\boldsymbol{c}$ short is a level-one encoding of the coset $\boldsymbol{c}+\mathcal{I}$ relative to the "j'th dimension". In this case we use vectors rather than integers to represent the different levels, where for an index $\boldsymbol{w} = \langle w_1, \ldots, w_\tau \rangle \in \mathbb{N}^\tau$ and a coset $\boldsymbol{c'}+\mathcal{I}$, the encodings of $\boldsymbol{c'}+\mathcal{I}$ relative to the index $\boldsymbol{w}$ are

$$S_{\boldsymbol{w}}^{(\mathbf{c}'+\mathcal{I})} \;=\; \left\{ \boldsymbol{c}/\boldsymbol{z}^* \;:\; \boldsymbol{c} \in \boldsymbol{c}' + \mathcal{I},\ \|\boldsymbol{c}\| < q^{1/8},\ \boldsymbol{z}^* = \prod_{i=1}^{\tau} \mathbf{z}_i^{w_i} \right\}.$$

To enable encoding in this asymmetric variant, we provide the public parameters $\mathbf{y}_j = [\mathbf{a}_j/\mathbf{z}_j]_q$ and $\{\mathbf{x}_{i,j} = [\mathbf{b}_{i,j}/\mathbf{z}_j]_q\}_i$ for all $j = 1, 2, \ldots, \kappa$, with short $\mathbf{a}_i \in 1+\mathcal{I}$ and $\mathbf{b}_{i,j} \in \mathcal{I}$. To enable zero-test relative to index $\langle v_1, \ldots, v_\tau \rangle \in \mathbb{N}^\tau$ we provide the zero-test parameter $\mathbf{p}_{zt} = (\boldsymbol{h} \cdot \prod_{i=1}^{\tau} \mathbf{z}_i^{v_i})/\mathbf{g} \in R_q$. The parameters for this variant will have to be set in order to provide functionality up to $\sum_i v_i$ levels. In particular, we will need $q > 2^{8\kappa\lambda^{\sum_i v_i} n^{O(\kappa)}}$ and $n > \tilde{O}(\kappa\lambda^{1+\sum_i v_i})$.

Chapter 3

Obfuscation

3.1 Preliminaries

In this section we define the notion of "virtual black-box" obfuscation and indistinguishability obfuscation. Additionally we recall the definition of branching programs and describe a "*dual-input*" variant of branching programs used in our construction.

3.1.1 "Virtual Black-Box" Obfuscation

We roughly the "Virtual Black-Box" Obfuscation (or, VBB for short) definition requires that having access to the obfuscated circuit is as good as having only oracle access to the circuit.

Definition 3.1.1 ("Virtual Black-Box" Obfuscation)**.** *We say that a uniform PPT machine $\mathcal{O}$ is a "Virtual Black-Box" Obfuscator for $\{\mathcal{C}_\ell\}_{\ell\in\mathbb{N}}$, if the following conditions are satisfied:*

- *Functionality: For every $\ell \in \mathbb{N}$, every $C \in \mathcal{C}_\ell$, every input x to C:*

$$\Pr[(\mathcal{O}(C))(x) \neq C(x)] \leq \mathit{negl}(|C|) \ ,$$

  where the probability is over the coins of $\mathcal{O}$.

- *Polynomial Slowdown: there exist a polynomial p such that for every $\ell \in \mathbb{N}$ and every $C \in \mathcal{C}_\ell$, we have that $|\mathcal{O}(C)| \leq p(|C|)$.*

- *Virtual Black-Box: for every PPT predicate $\mathcal{A}$ there exist a PPT simulator $\mathcal{S}$, and a negligible function μ such that for all PPT distinguishers D, for every $\ell \in \mathbb{N}$ and every $C \in \mathcal{C}_\ell$:*

$$\left|\Pr[D(\mathcal{A}(\mathcal{O}(C))) = 1] - \Pr[D(\mathcal{S}^C(1^{|C|})) = 1]\right| \leq \mu(|C|) \ ,$$

  where the probabilities are over the coins of D,$\mathcal{A}$,$\mathcal{S}$, and $\mathcal{O}$.

3.1.2 Indistinguishability Obfuscators

Given that the VBB definition is impossible in general we also consider the weaker indistinguishability based definition.

Definition 3.1.2 (Indistinguishability Obfuscator ($i\mathcal{O}$))**.** *A uniform PPT machine $i\mathcal{O}$ is called an* indistinguishability obfuscator *for a circuit class $\{\mathcal{C}_\lambda\}$ if the following conditions are satisfied:*

- *For all security parameters $\lambda \in \mathbb{N}$, for all $C \in \mathcal{C}_\lambda$, for all inputs x, we have that*

$$\Pr[C'(x) = C(x) : C' \leftarrow i\mathcal{O}(\lambda, C)] = 1$$

- *For any (not necessarily uniform) PPT distinguisher D, there exists a negligible function α such that the following holds: For all security parameters $\lambda \in \mathbb{N}$, for all pairs of circuits $C_0, C_1 \in \mathcal{C}_\lambda$, we have that if $C_0(x) = C_1(x)$ for all inputs x, then*

$$\Big| \Pr\big[D(i\mathcal{O}(\lambda, C_0)) = 1\big] - \Pr\big[D(i\mathcal{O}(\lambda, C_1)) = 1\big] \Big| \leq \alpha(\lambda)$$

Definition 3.1.3 (Indistinguishability Obfuscator for NC^1)**.** *A uniform PPT machine $i\mathcal{O}$ is called an* indistinguishability obfuscator *for NC^1 if for all constants $c \in \mathbb{N}$, the following holds: Let $\mathcal{C}_\lambda$ be the class of circuits of depth at most $c \log \lambda$ and size at most λ. Then $i\mathcal{O}(c, \cdot, \cdot)$ is an indistinguishability obfuscator for the class $\{\mathcal{C}_\lambda\}$.*

Definition 3.1.4 (Indistinguishability Obfuscator for $P/poly$)**.** *A uniform PPT machine $i\mathcal{O}$ is called an* indistinguishability obfuscator *for $P/poly$ if the following holds: Let $\mathcal{C}_\lambda$ be the class of circuits of size at most λ. Then $i\mathcal{O}$ is an indistinguishability obfuscator for the class $\{\mathcal{C}_\lambda\}$.*

3.1.3 Branching Programs

A branching program consists of a sequence of steps, where each step is defined by a pair of permutations. In each step the the program examines one input bit, and depending on its value the program chooses one of the permutations. The program outputs 1 if and only if the multiplications of the permutations chosen in all steps is the identity permutation.

Definition 3.1.5 (Oblivious Matrix Branching Program)**.** *A* branching program *of width w and length n for ℓ-bit inputs is given by a permutation matrix $P_{\mathsf{reject}} \in \{0,1\}^{w \times w}$ such that $P_{\mathsf{reject}} \neq I_{w \times w}$, and by a sequence:*

$$\mathsf{BP} = \big(\mathsf{inp}(i), B_{i,0}, B_{i,1}\big)_{i=1}^{n} \ ,$$

where each $B_{i,b}$ is a permutation matrix in $\{0,1\}^{w \times w}$, and $\mathsf{inp}(i) \in [\ell]$ is the input bit position examined in step i. The output of the branching program on input $x \in \{0,1\}^\ell$ is as follows:

$$BP(x) \overset{\text{def}}{=} \begin{cases} 1 & \textit{if } \prod_{i=1}^{n} B_{i,x_{\mathsf{inp}(i)}} = I_{w \times w} \\ 0 & \textit{if } \prod_{i=1}^{n} B_{i,x_{\mathsf{inp}(i)}} = P_{\mathsf{reject}} \\ \perp & \textit{otherwise} \end{cases}$$

The branching program is said to be oblivious *if $\mathsf{inp} : [n] \to [\ell]$ is a fixed function, independent of the function being evaluated.*

Theorem 3.1.6 ([Bar86])**.** *For any depth-d fan-in-2 boolean circuit C, there exists an oblivious branching program of width 5 and length at most 4^d that computes the same function as the circuit C.*

We present an obfuscation construction that uses a variant of branching programs that we call *dual-input* branching programs. Instead of reading one input bit in every step, a dual-input branching program inspects a pair of input bits and chooses a permutation based on the values of both bits.

Definition 3.1.7 (Dual-Input Branching Program)**.** *A* Oblivious dual-input *branching program of width w and length n for ℓ-bit inputs is given by a permutation matrix $P_{\mathsf{reject}} \in \{0,1\}^{w\times w}$ such that $P_{\mathsf{reject}} \neq I_{w\times w}$, and by a sequence*

$$\mathsf{BP} = \left(\mathsf{inp}_1(i), \mathsf{inp}_2(i), \{B_{i,b_1,b_2}\}_{b_1,b_2\in\{0,1\}}\right)_{i=1}^{n},$$

where each B_{i,b_1,b_2} is a permutation matrix in $\{0,1\}^{w\times w}$, and $\mathsf{inp}_1(i), \mathsf{inp}_2(i) \in [\ell]$ are the positions of the input bits inspected in step i. The output of the branching program on input $x \in \{0,1\}^\ell$ is as follows:

$$\mathsf{BP}(x) \overset{\text{def}}{=} \begin{cases} 1 & \textit{if } \prod_{i=1}^{n} B_{i,x_{\mathsf{inp}_1(i)},x_{\mathsf{inp}_2(i)}} = I_{w\times w} \\ 0 & \textit{if } \prod_{i=1}^{n} B_{i,x_{\mathsf{inp}_1(i)},x_{\mathsf{inp}_2(i)}} = P_{\mathsf{reject}} \\ \bot & \textit{otherwise} \end{cases}$$

As before, the dual-input branching program is said to be oblivious *if both $\mathsf{inp}_1 : [n] \to [\ell]$ and $\mathsf{inp}_2 : [n] \to [\ell]$ are fixed functions, independent of the function being evaluated.*

Note that any branching program can be simulated by a dual-input branching program with the same width and length, since the dual-input branching program can always "ignore" one input bit in each pair.

3.1.4 Straddling Set System

In this section, we define the notion of a *straddling set system*, and prove combinatorial properties regarding this set system. This set system will be an ingredient in the obfusctaion construction, and the specific combinatorial properties that we establish will be essential for the security of the construction.

Definition 3.1.8. *A* straddling set system *with n entries is a collection of sets $\mathbb{S}_n = \{S_{i,b}, : i \in [n], b \in \{0,1\}\}$ over a universe U, such that*

$$\cup_{i\in[n]} S_{i,0} = \cup_{i\in[n]} S_{i,1} = U$$

and for every distinct non-empty sets $C, D \subseteq \mathbb{S}_n$ we have that if:

1. *(Disjoint Sets:) C contains only disjoint sets. D contains only disjoint sets.*
2. *(Collision:) $\cup_{S\in C} S = \cup_{S\in D} S$*

Then, it must be that $\exists\ b \in \{0,1\}$:

$$C = \{S_{j,b}\}_{j\in[n]} \ \ , \ \ D = \{S_{j,(1-b)}\}_{j\in[n]} \ \ .$$

Therefore, in a straddling set system, the only exact covers of the universe U are $\{S_{j,0}\}_{j\in[n]}$ and $\{S_{j,1}\}_{j\in[n]}$.

Construction. Let $\mathbb{S}_n = \{S_{i,b}, : i \in [n], b \in \{0,1\}\}$, over the universe $U = \{1, 2, \ldots, 2n-1\}$, where:

$S_{1,0} = \{1\}$, $S_{2,0} = \{2,3\}$, $S_{3,0} = \{4,5\}$, $\ldots$, $S_{i,0} = \{2i-2, 2i-1\}$, $\ldots$, $S_{n,0} = \{2n-2, 2n-1\}$; and,

$S_{1,1} = \{1,2\}$, $S_{2,1} = \{3,4\}$, $\ldots$, $S_{i,1} = \{2i-1, 2i\}$, $\ldots$, $S_{n-1,1} = \{2n-3, 2n-2\}$, $S_{n,1} = \{2n-1\}$.

The proof that Construction 3.1.4 satisfies the definition of a straddling set system is straightforward and is skipped.

3.2 Obfuscation Candidate for $\mathbf{NC}^1$

In this section we describe the "virtual black-box" obfuscator $\mathcal{O}$ for $\mathbf{NC}^1$. Next we show how one this obfuscation can be bootstrapped to obtain an obfuscation for arbitrary circuits. The security of this candidate is argued in [BGK+14].

Input. The obfuscator $\mathcal{O}$ takes as input a circuit and transforms it into an oblivious dual-input branching program BP of width w and length n for ℓ-bit inputs:

$$\mathsf{BP} = \left(\mathsf{inp}_1(i), \mathsf{inp}_2(i), \{B_{i,b_1,b_2}\}_{b_1,b_2 \in \{0,1\}}\right)_{i=1}^{n}.$$

Recall that each B_{i,b_1,b_2} is a permutation matrix in $\{0,1\}^{w \times w}$, and $\mathsf{inp}_1(i), \mathsf{inp}_2(i) \in [\ell]$ are the positions of the input bits inspected in step i. Without loss of generality, we make the following assumptions on the structure of the brunching program BP:

- In every step BP inspects two different input bits; that is, for every step $i \in [n]$, we have $\mathsf{inp}_1(i) \neq \mathsf{inp}_2(i)$.
- Every pair of different input bits are inspected in some step of BP; that is, for every $j_1, j_2 \in [\ell]$ such that $j_1 \neq j_2$ there exists a step $i \in [n]$ such that $(\mathsf{inp}_1(i), \mathsf{inp}_2(i)) = (j_1, j_2)$.
- Every bit of the input is inspected by BP exactly ℓ' times. More precisely, for input bit $j \in [\ell]$, we denote by $\mathsf{ind}(j)$ the set of steps that inspect the j'th bit:

  $$\mathsf{ind}(j) = \{i \in [n] : \mathsf{inp}_1(i) = j\} \cup \{i \in [n] : \mathsf{inp}_2(i) = j\} \ \ .$$

  We assume that for every input bit $j \in [\ell]$, $|\mathsf{ind}(j)| = \ell'$. Note that in every step, the j'th input bit can be inspected at most once.

Randomizing. Next, the Obfuscator $\mathcal{O}$ "randomizes" the branching program BP as follows. First, $\mathcal{O}$ samples a prime p of length $\Theta(n)$. Then, $\mathcal{O}$ samples random and independent elements as follows:

- Non-zero scalars $\{\alpha_{i,b_1,b_2} \in \mathbb{Z}_p : i \in [n], b_1, b_2 \in \{0,1\}\}$.
- Pair of vectors $\mathbf{s}, \mathbf{t} \in \mathbb{Z}_p^w$.
- $n+1$ random full-rank matrices $R_0, R_1, \ldots, R_n \in \mathbb{Z}_p^{w \times w}$.

Finally, $\mathcal{O}$ computes the pair of vectors:

$$\tilde{\mathbf{s}} = \mathbf{s}^t \cdot R_0^{-1}, \quad \tilde{\mathbf{t}} = R_n \cdot \mathbf{t} \ \ ,$$

and for every $i \in [n]$ and $b_1, b_2 \in \{0,1\}$, $\mathcal{O}$ computes the matrix:

$$\tilde{B}_{i,b_1,b_2} = R_{i-1} \cdot B_{i,b_1,b_2} \cdot R_i^{-1}.$$

Initialization. For every $j \in [\ell]$, let $\mathbb{S}^j$ be a straddling set system with ℓ' entries over a set U_j, such that the sets $U_1, \ldots, U_\ell$ are disjoint. Let $U = \bigcup_{j \in [\ell]} U_j$, and let B_s and B_t be sets such that U, B_s, B_t are disjoint. We associate the set system $\mathbb{S}^j$ with the j'th input bit. We index the elements of $\mathbb{S}^j$ by the steps of the branching program BP that inspect the j'th input. Namely,

$$\mathbb{S}^j = \left\{ S^j_{k,b} : k \in \mathsf{ind}(j), b \in \{0,1\} \right\} .$$

For every step $i \in [n]$ and bits $b_1, b_2 \in \{0,1\}$ we denote by $S(i, b_1, b_2)$ the union of pairs of sets that are indexed by i:

$$S(i, b_1, b_2) = S^{\mathsf{inp}_1(i)}_{i,b_1} \cup S^{\mathsf{inp}_2(i)}_{i,b_2} \ .$$

Note that by the way we defined the set $\mathsf{ind}(j)$ for input bit $j \in [\ell]$, and by the way the elements of $\mathbb{S}^j$ are indexed, indeed, $S^{\mathsf{inp}_1(i)}_{i,b_1} \in \mathbb{S}^{\mathsf{inp}_1(i)}$ and $S^{\mathsf{inp}_2(i)}_{i,b_2} \in \mathbb{S}^{\mathsf{inp}_2(i)}$.

$\mathcal{O}$ generates the parameters $\mathbf{g}, \mathbf{z}, \mathbf{p}_{zt}$ for multilinear mops. Next it interprets each of the above terms in $\mathbb{Z}_p$ as a small element from the corresponding coset on $\mathcal{I} = \langle \mathbf{g} \rangle$. We use the asymmetric setting of multilinear maps with the universe set $U \cup B_s \cup B_t$. Here we need to encode the following values:

$$\begin{aligned}
&(\mathbf{s} \cdot \mathbf{t}, B_s \cup B_t), \\
&\left\{ (\tilde{\mathbf{s}}[j], B_s), (\tilde{\mathbf{t}}[j], B_t) \right\}_{j \in [w]} \\
&\{ (\alpha_{i,b_1,b_2}, S(i, b_1, b_2)) \}_{i \in [n], b_1, b_2 \in \{0,1\}} \\
&\left\{ (\alpha_{i,b_1,b_2} \cdot \tilde{B}_{i,b_1,b_2}[j,k], S(i, b_1, b_2)) \right\}_{i \in [n], b_1, b_2 \in \{0,1\}, j,k \in [w]}
\end{aligned}$$

We denote the encoding of the element α at level S by $[\alpha]_S$. For a matrix M, $[M]_S$ denotes a matrix of encodings such that $[M]_S[j,k]$ is the encoding of the element $(M[j,k], S)$. Using this notation, $\mathcal{O}$ outputs the obfuscation as:

$$[\tilde{\mathbf{s}}]_{B_s} , \quad [\tilde{\mathbf{t}}]_{B_t} , \quad [\mathbf{s} \cdot \mathbf{t}]_{B_s \cup B_t} , \quad \left\{ [\alpha_{i,b_1,b_2}]_{S(i,b_1,b_2)} , \quad \left[\alpha_{i,b_1,b_2} \cdot \tilde{B}_{i,b_1,b_2} \right]_{S(i,b_1,b_2)} \right\}_{i \in [n], b_1, b_2 \in \{0,1\}} .$$

Output. The obfuscator $\mathcal{O}$ outputs the above described encodings and the $\mathbf{p}_{zt}$ parameter.

Evaluation. Given two encodings $[\alpha]_S$ and $[\beta]_S$, we can obtain the sum as $[\alpha]_S + [\beta]_S$. Similarly, given two encodings $[\alpha_1]_{S_1}$ and $[\alpha_2]_{S_2}$ such that $S_1 \cap S_2 = \emptyset$, we obtain the product as $[\alpha_1]_{S_1} \cdot [\alpha_2]_{S_2}$. Given two matrices of encodings $[M_1]_{S_1}$, $[M_2]_{S_2}$, we define their matrix multiplication in the natural way, and denote it by $[M_1]_{S_1} \cdot [M_2]_{S_2}$.

For input $x \in \{0,1\}^\ell$ to $\mathcal{O}(\mathsf{BP})$, and for every $i \in [n]$ let $(b^i_1, b^i_2) = (x_{\mathsf{inp}_1(i)}, x_{\mathsf{inp}_2(i)})$. On input x, $\mathcal{O}(\mathsf{BP})$ obtains the following encodings:

$$h = [\tilde{\mathbf{s}}]_{B_s} \cdot \prod_{i=1}^{n} \left[\alpha_{i,b^i_1,b^i_2} \cdot \tilde{B}_{i,b^i_1,b^i_2} \right]_{S(i,b^i_1,b^i_2)} \cdot [\tilde{\mathbf{t}}]_{B_t} , \quad h' = [\mathbf{s} \cdot \mathbf{t}]_{B_s \cup B_t} \cdot \prod_{i=1}^{n} \left[\alpha_{i,b^i_1,b^i_2} \right]_{S(i,b^i_1,b^i_2)}$$

$\mathcal{O}(\mathsf{BP})$ uses $\mathbf{p}_{zt}$ to check if $h' - h$ is not an encoding of 0. If the zero test outputs 1 then $\mathcal{O}(\mathsf{BP})$ outputs 1, and otherwise $\mathcal{O}(\mathsf{BP})$ outputs 0.

Correctness. By construction we have that $h' - h$ is an encoding of 0 if and only if:

$$\begin{aligned}
0 &= \tilde{\mathbf{s}} \cdot \prod_{i=1}^{n} \alpha_{i,b_1^i,b_2^i} \cdot \tilde{B}_{i,b_1^i,b_2^i} \cdot \tilde{\mathbf{t}} - \mathbf{s} \cdot \mathbf{t} \cdot \prod_{i=1}^{n} \alpha_{i,b_1^i,b_2^i} \\
&= \left(\tilde{\mathbf{s}} \cdot \prod_{i=1}^{n} \tilde{B}_{i,b_1^i,b_2^i} \cdot \tilde{\mathbf{t}} - \mathbf{s} \cdot \mathbf{t} \right) \cdot \prod_{i=1}^{n} \alpha_{i,b_1^i,b_2^i} \\
&= \left(\mathbf{s}^t \cdot R_0^{-1} \cdot \prod_{i=1}^{n} \left(R_{i-1} \cdot B_{i,b_1,b_2} \cdot R_i^{-1} \right) \cdot R_n^{-1} \cdot \mathbf{t} - \mathbf{s} \cdot \mathbf{t} \right) \cdot \prod_{i=1}^{n} \alpha_{i,b_1^i,b_2^i} \\
&= \mathbf{s}^t \cdot \left(\prod_{i=1}^{n} B_{i,b_1,b_2} - I_{w \times w} \right) \cdot \mathbf{t} \cdot \prod_{i=1}^{n} \alpha_{i,b_1^i,b_2^i}
\end{aligned}$$

From the definition of the branching program we have:

$$\mathsf{BP}(x) = 1 \Leftrightarrow \prod_{i=1}^{n} B_{i,b_1^i,b_2^i} = I_{w \times w}$$

Thus, if $\mathsf{BP}(x) = 1$ then $\mathcal{O}(\mathsf{BP})$ outputs 1 with probability 1. If $\mathsf{BP}(x) = 0$ then $\mathcal{O}(\mathsf{BP})$ outputs 1 with probability at most $1/p = \mathrm{negl}(n)$ over the choice of $\mathbf{s}$ and $\mathbf{t}$.

It is left to show that when multiplying two matrices of of encodings $[M_1]_{S_1} \cdot [M_2]_{S_2}$, none of the addition or multiplication calls fail as long as $S_1 \cap S_2 = \emptyset$. Therefore, to show that none of the addition or multiplication fail, it is enough to show that following sets are disjoint:

$$B_s, B_t, S(1, b_1^1, b_2^1), \ldots, S(n, b_1^n, b_2^n) \ .$$

Their disjointness follows from the fact that $U_1, \ldots, U_\ell, B_s, B_t$ are disjoint, together with definition of $S(i, b_1^i, b_2^i)$ and with the fact that for every set system $\mathbb{S}^j$, for every distinct $i, i' \in \mathsf{ind}(j)$, and for every $b \in \{0,1\}$, we have that $S_{i,b}^j \cap S_{i',b}^j = \emptyset$.

Finally zero testing is possible as the encoding $h - h'$ corresponds to the entire universe. Namely, we have that:

$$\left(\bigcup_{i=1}^{n} S(i, b_1^i, b_2^i) \right) \cup B_s \cup B_t = U \cup B_s \cup B_t \ ,$$

which follows from the following equalities:

$$\bigcup_{i=1}^{n} S(i, b_1^i, b_2^i) = \bigcup_{i=1}^{n} S_{i,b_1^i}^{\mathsf{inp}_1(i)} \cup S_{i,b_2^i}^{\mathsf{inp}_2(i)} = \bigcup_{j=1}^{\ell} \bigcup_{k \in \mathsf{ind}(j)} S_{k,x_i}^{j} = \bigcup_{j=1}^{\ell} U_j = U \ .$$

We refer the reader to [BGK$^+$14] for security analysis of this construction.

3.3 Amplifying to Poly-sized Circuit Indistinguishability Obfuscation

In this section we present Poly-sized Circuit Indistinguishability Obfuscation from an indistinguishability obfuscator, $i\mathcal{O}_{\mathbf{NC}^1}$, for circuits in $\mathbf{NC}^1$. In addition to $i\mathcal{O}$, our construction makes use a Leveled Homomorphic Encryption. We let $(\mathsf{Setup}_{FHE}, \mathsf{Encrypt}_{FHE}, \mathsf{Decrypt}_{FHE}, \mathsf{Eval}_{FHE})$ be a leveled homomorphic encryption scheme. Furthermore, we assume the decryption algorithm $\mathsf{Decrypt}_{FHE}$ can be realized by a family of circuits in $\mathbf{NC}^1$ and that the system has *perfect correctness.*

3.3.1 Our Construction

Consider a family of circuit classes $\{\mathcal{C}_\lambda\}$ for $\lambda \in \mathbb{N}$ where the input size, n, and the maximum circuit size are polynomial functions of λ. Let $\{U_\lambda\}$ be a poly-sized universal circuit family for these circuit classes, where $U_\lambda(C, m) = C(m)$ for all $C \in \{\mathcal{C}_\lambda\}$ and $m \in \{0,1\}^n$. Furthermore, all circuits $C \in \{\mathcal{C}_\lambda\}$ can be encoded as an $\ell = \ell(\lambda)$ (a polynomial functions of λ) bit string as input to U. We note that this may require some padding since circuits in the family are not required to be all of the same size.

We show how to build an $i\mathcal{O}$ for such a circuit class given an indistinguishability obfuscator, $i\mathcal{O}_{\mathbf{NC}^1}$, for circuits in $\mathbf{NC}^1$.

Our construction is described by an obfuscation algorithm and an evaluation algorithm.[1]

- $\mathsf{Obfuscate}(1^\lambda, C \in \mathcal{C}_\lambda)$: The $\mathsf{Obfuscate}$ algorithm takes the security parameter λ and a circuit C and computes

  1. Generate $(\mathsf{PK}^1_{FHE}, \mathsf{SK}^1_{FHE}) \leftarrow \mathsf{Setup}_{FHE}(1^\lambda)$ and $(\mathsf{PK}^2_{FHE}, \mathsf{SK}^2_{FHE}) \leftarrow \mathsf{Setup}_{FHE}(1^\lambda)$. The number of levels in leveled HE scheme should be set to be the depth of U_λ.
  2. Encrypt $g_1 = \mathsf{Encrypt}_{FHE}(\mathsf{PK}^1_{FHE}, C)$ and $g_2 = \mathsf{Encrypt}_{FHE}(\mathsf{PK}^2_{FHE}, C)$. Here we assume that C is encoded in a canonical form as an ℓ bit string for use by the universal circuit $U_\lambda(\cdot,\cdot)$
  3. Generate an NC^1 obfuscation for program $\mathrm{P1}^{(\mathsf{SK}^1_{FHE}, g_1, g_2)}$ as $P = i\mathcal{O}_{\mathbf{NC}^1}(\mathrm{P1}^{(\mathsf{SK}^1_{FHE}, g_1, g_2)})$. (See Figure 3.1.)
  4. The obfuscation components are output as: $\sigma = (P, \mathsf{PK}^1_{FHE}, \mathsf{PK}^2_{FHE}, g_1, g_2)$.

- $\mathsf{Evaluate}(\sigma = (P, \mathsf{PK}^1_{FHE}, \mathsf{PK}^2_{FHE}, g_1, g_2), m)$: The $\mathsf{Evaluate}$ algorithm takes in the obfuscation output σ and program input m and computes the following.

  1. Compute $e_1 = \mathsf{Eval}_{FHE}(\mathsf{PK}^1_{FHE}, U_\lambda(\cdot, m), g_1)$ and $e_2 = \mathsf{Eval}_{FHE}(\mathsf{PK}^2_{FHE}, U_\lambda(\cdot, m), g_2)$. [2]
  2. Compute a low-depth proof ϕ that e_1 and e_2 were computed correctly. (Note that verification of any computation can be done in $\mathbf{NC}^1$.) We note that this statement for which proof is being given is in P
  3. Run $P(m, e_1, e_2, \phi)$ and output the result.

Correctness To verify correctness we first check that the size of the circuits evaluating $\mathrm{P1}^{(\mathsf{SK}^1_{FHE}, g_1, g_2)}$ are in $\mathbf{NC}^1$. Step 1 of the program is in $\mathbf{NC}^1$ since it involves verification of a low depth proof. The second step of the program is also in $\mathbf{NC}^1$ since we use an HE scheme with decryption in $\mathbf{NC}^1$. Since both steps are in $\mathbf{NC}^1$ therefore the entire circuit is in $\mathbf{NC}^1$.

The correctness of HE implies that the evaluated ciphertext e_1 will be an encryption of the value $U(C, m) = C(m)$. Next the correctness of the $i\mathcal{O}_{\mathbf{NC}^1}$ implies that P implements the same function as $P1$. These two facts together imply correctness of the obfuscation scheme.

[1] Technically, we could make do with just a single obfuscation algorithm that outputs a circuit description as is the convention given in Section 3.1. However, for the exposition of this construction we have the obfuscation algorithm output cryptographic material that is used by the evaluation algorithm.

[2] The circuit $U_\lambda(\cdot, m)$ is the universal circuit with m hardwired in as an input. This in hardwired circuit takes in an ℓ bit circuit description C as its input and evaluates to $U(C, m)$.

P1

Given input (m, e_1, e_2, ϕ), $\mathrm{P1}^{(\mathsf{SK}^1_{FHE}, g_1, g_2)}$ proceeds as follows:

1. Check if ϕ is a valid low-depth proof for the NP-statement:

$$e_1 = \mathsf{Eval}_{FHE}(\mathtt{PK}^1_{FHE}, U_\lambda(\cdot, m), g_1) \quad \bigwedge \quad e_2 = \mathsf{Eval}_{FHE}(\mathtt{PK}^2_{FHE}, U_\lambda(\cdot, m), g_2).$$

2. If the check fails output 0; otherwise, output $\mathsf{Decrypt}_{FHE}(e_1, \mathsf{SK}^1_{FHE})$.

Figure 3.1:

P2

Given input (m, e_1, e_2, ϕ), $\mathrm{P2}^{(\mathsf{SK}^2_{FHE}, g_1, g_2)}$ proceeds as follows:

1. Check if ϕ is a valid low-depth proof for the NP-statement:

$$e_1 = \mathsf{Eval}_{FHE}(\mathtt{PK}^1_{FHE}, U_\lambda(\cdot, m), g_1) \quad \bigwedge \quad e_2 = \mathsf{Eval}_{FHE}(\mathtt{PK}^2_{FHE}, U_\lambda(\cdot, m), g_2).$$

2. If the check fails output 0; otherwise, output $\mathsf{Decrypt}_{FHE}(e_2, \mathsf{SK}^2_{FHE})$.

Figure 3.2:

3.3.2 Proof of Security

We prove that for all $C_0, C_1 \in \mathcal{C}_\lambda$ such that they are functionally equivalent, there can be no poly-time indistinguishability attacker that can distinguish between the obfuscations of C_0 and C_1.

We organize our proof into a sequence of hybrids. In the first hybrid the challenger obfuscates C_0. We then gradually change the obfuscation in multiple hybrid steps into an obfuscation of C_1. We show that each successive hybrid experiment is indistinguishable from the last, thus showing our obfuscator to have indistinguishability security. The proof hybrid steps themselves primarily weave back and forth in between changing the underlying ciphertexts and the programs that are used in a two-key proof type manner. The changes program that we will use is described in Figure 3.2.

Sequence of Hybrids

- Hyb_0: This hybrid corresponds to a honest execution of the Indistinguishability Obfuscation game where C_0 is obfuscated.

- Hyb_1: Same as hybrid Hyb_0 except we generate $g_1 = \mathsf{Encrypt}_{FHE}(\mathtt{PK}^1_{FHE}, C_0)$, $g_2 = \mathsf{Encrypt}_{FHE}(\mathtt{PK}^2_{FHE}, C_1)$. Now g_1 and g_2 encrypt different circuits.

  Computational indistinguishability between Hyb_0 and Hyb_1 follows from the semantic security of the encryption scheme.

- Hyb_2: We still generate $g_1 = \mathsf{Encrypt}_{FHE}(\mathtt{PK}^1_{FHE}, C_0)$ and $g_2 = \mathsf{Encrypt}_{FHE}(\mathtt{PK}^2_{FHE}, C_1)$ as in Hyb_1. Now P is created as $P = i\mathcal{O}_{\mathbf{NC}^1}(\mathrm{P2}^{(\mathsf{SK}^2_{FHE}, g_1, g_2)})$.

  Computational indistinguishability between Hyb_1 and Hyb_2 follows from indistinguishability obfuscation of $\mathbf{NC}^1$ obfuscator.

- Hyb_3: We now generate $g_1 = \mathsf{Encrypt}_{FHE}(\mathtt{PK}^1_{FHE}, C_1)$ and $g_2 = \mathsf{Encrypt}_{FHE}(\mathtt{PK}^2_{FHE}, C_1)$. The obfuscated program P is still created as $P = i\mathcal{O}_{\mathbf{NC}^1}(\mathrm{P2}^{(\mathsf{SK}^2_{FHE}, g_1, g_2)})$.

Computational indistinguishability between Hyb_2 and Hyb_3 follows from the semantic security of the encryption scheme.

- Hyb_4: We still generate $g_1 = \mathsf{Encrypt}_{FHE}(\mathsf{PK}^1_{FHE}, C_1)$ and $g_2 = \mathsf{Encrypt}_{FHE}(\mathsf{PK}^2_{FHE}, C_1)$ as in Hyb_3. Now P is created as $P = i\mathcal{O}_{\mathbf{NC}^1}(\mathrm{P1}^{(\mathsf{SK}^1_{FHE}, g_1, g_2)})$. Note that this hybrid corresponds to obfuscation of C_1.

  Computational indistinguishability between Hyb_3 and Hyb_4 follows from indistinguishability obfuscation of $\mathbf{NC}^1$ obfuscator.

Bibliography

[AGHS12] Shweta Agrawal, Craig Gentry, Shai Halevi, and Amit Sahai. Sampling discrete gaussians efficiently and obliviously. Cryptology ePrint Archive, Report 2012/714, 2012. http://eprint.iacr.org/.

[AGIS14] Prabhanjan Vijendra Ananth, Divya Gupta, Yuval Ishai, and Amit Sahai. Optimizing obfuscation: Avoiding Barrington's theorem. In Gail-Joon Ahn, Moti Yung, and Ninghui Li, editors, *ACM CCS 14: 21st Conference on Computer and Communications Security*, pages 646–658, Scottsdale, AZ, USA, November 3–7, 2014. ACM Press.

[AJ15] Prabhanjan Ananth and Abhishek Jain. Indistinguishability obfuscation from compact functional encryption. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 308–326, Santa Barbara, CA, USA, August 16–20, 2015. Springer, Heidelberg, Germany.

[AJS15] Prabhanjan Ananth, Abhishek Jain, and Amit Sahai. Indistinguishability obfuscation with constant size overhead. Cryptology ePrint Archive, Report 2015/1023, 2015. http://eprint.iacr.org/2015/1023.

[AR05] Dorit Aharonov and Oded Regev. Lattice problems in np cap conp. *J. ACM*, 52(5):749–765, 2005.

[Bar86] D A Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in nc_1. In *STOC*, 1986.

[BC10] Nir Bitansky and Ran Canetti. On strong simulation and composable point obfuscation. In Tal Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 520–537, Santa Barbara, CA, USA, August 15–19, 2010. Springer, Heidelberg, Germany.

[Bei11] Amos Beimel. Secret-sharing schemes: A survey. In Yeow Meng Chee, Zhenbo Guo, San Ling, Fengjing Shao, Yuansheng Tang, Huaxiong Wang, and Chaoping Xing, editors, *Coding and Cryptology - Third International Workshop, IWCC 2011*, volume 6639 of *Lecture Notes in Computer Science*, pages 11–46, Qingdao, China, May 30-June 3 2011. Springer.

[BF11a] Dan Boneh and David Mandell Freeman. Homomorphic signatures for polynomial functions. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 149–168, Tallinn, Estonia, May 15–19, 2011. Springer, Heidelberg, Germany.

[BF11b] Dan Boneh and David Mandell Freeman. Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *PKC 2011: 14th International Conference on Theory and Practice of Public Key Cryptography*, volume 6571 of *Lecture Notes in Computer Science*, pages 1–16, Taormina, Italy, March 6–9, 2011. Springer, Heidelberg, Germany.

[BGI$^+$01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 1–18, Santa Barbara, CA, USA, August 19–23, 2001. Springer, Heidelberg, Germany.

[BGI$^+$12] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. *J. ACM*, 59(2):6, 2012.

[BGK$^+$14] Boaz Barak, Sanjam Garg, Yael Tauman Kalai, Omer Paneth, and Amit Sahai. Protecting obfuscation against algebraic attacks. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 221–238, Copenhagen, Denmark, May 11–15, 2014. Springer, Heidelberg, Germany.

[BMSZ15] Saikrishna Badrinarayanan, Eric Miles, Amit Sahai, and Mark Zhandry. Post-zeroizing obfuscation: The case of evasive circuits. Cryptology ePrint Archive, Report 2015/167, 2015. `http://eprint.iacr.org/2015/167`.

[BPR15] Nir Bitansky, Omer Paneth, and Alon Rosen. On the cryptographic hardness of finding a Nash equilibrium. In Venkatesan Guruswami, editor, *56th Annual Symposium on Foundations of Computer Science*, pages 1480–1498, Berkeley, CA, USA, October 17–20, 2015. IEEE Computer Society Press.

[BR13] Zvika Brakerski and Guy N. Rothblum. Obfuscating conjunctions. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 416–434, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Heidelberg, Germany.

[BR14a] Zvika Brakerski and Guy N. Rothblum. Black-box obfuscation for d-CNFs. In Moni Naor, editor, *ITCS 2014: 5th Innovations in Theoretical Computer Science*, pages 235–250, Princeton, NJ, USA, January 12–14, 2014. Association for Computing Machinery.

[BR14b] Zvika Brakerski and Guy N. Rothblum. Virtual black-box obfuscation for all circuits via generic graded encoding. In Yehuda Lindell, editor, *TCC 2014: 11th Theory of Cryptography Conference*, volume 8349 of *Lecture Notes in Computer Science*, pages 1–25, San Diego, CA, USA, February 24–26, 2014. Springer, Heidelberg, Germany.

[BS96] Eric Bach and Jeffrey Shallit. *Algorithmic Number Theory, Volume I: Efficient Algorithms.* MIT Press, 1996.

[BS02] Dan Boneh and Alice Silverberg. Applications of multilinear forms to cryptography. Cryptology ePrint Archive, Report 2002/080, 2002. `http://eprint.iacr.org/2002/080`.

[BSW11] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: definitions and challenges. In *TCC*, pages 253–273, 2011.

[BV15] Nir Bitansky and Vinod Vaikuntanathan. Indistinguishability obfuscation from functional encryption. In Venkatesan Guruswami, editor, *56th Annual Symposium on Foundations of Computer Science*, pages 171–190, Berkeley, CA, USA, October 17–20, 2015. IEEE Computer Society Press.

[BVWW16] Zvika Brakerski, Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Obfuscating conjunctions under entropic ring LWE. In Madhu Sudan, editor, *ITCS 2016: 7th Innovations in Theoretical Computer Science*, pages 147–156, Cambridge, MA, USA, January 14–16, 2016. Association for Computing Machinery.

[BZ14] Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology – CRYPTO 2014, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 480–499, Santa Barbara, CA, USA, August 17–21, 2014. Springer, Heidelberg, Germany.

[Can97] Ran Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In Burton S. Kaliski Jr., editor, *Advances in Cryptology – CRYPTO'97*, volume 1294 of *Lecture Notes in Computer Science*, pages 455–469, Santa Barbara, CA, USA, August 17–21, 1997. Springer, Heidelberg, Germany.

[CCV12] Nishanth Chandran, Melissa Chase, and Vinod Vaikuntanathan. Functional re-encryption and collusion-resistant obfuscation. In Ronald Cramer, editor, *TCC 2012: 9th Theory of Cryptography Conference*, volume 7194 of *Lecture Notes in Computer Science*, pages 404–421, Taormina, Sicily, Italy, March 19–21, 2012. Springer, Heidelberg, Germany.

[CD08] Ran Canetti and Ronny Ramzi Dakdouk. Obfuscating point functions with multibit output. In Nigel P. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 489–508, Istanbul, Turkey, April 13–17, 2008. Springer, Heidelberg, Germany.

[CFL$^+$16] Jung Hee Cheon, Pierre-Alain Fouque, Changmin Lee, Brice Minaud, and Hansol Ryu. Cryptanalysis of the new clt multilinear map over the integers. Cryptology ePrint Archive, Report 2016/135, 2016. `http://eprint.iacr.org/`.

[CHL$^+$15] Jung Hee Cheon, Kyoohyung Han, Changmin Lee, Hansol Ryu, and Damien Stehlé. Cryptanalysis of the multilinear map over the integers. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 3–12, Sofia, Bulgaria, April 26–30, 2015. Springer, Heidelberg, Germany.

[CIJ$^+$13] Angelo De Caro, Vincenzo Iovino, Abhishek Jain, Adam O'Neill, Omer Paneth, and Giuseppe Persiano. On the achievability of simulation-based security for functional encryption. In *CRYPTO*, 2013.

[CLLT15] Jean-Sebastien Coron, Moon Sung Lee, Tancrede Lepoint, and Mehdi Tibouchi. Cryptanalysis of GGH15 multilinear maps. Cryptology ePrint Archive, Report 2015/1037, 2015. `http://eprint.iacr.org/2015/1037`.

[CLR15] Jung Hee Cheon, Changmin Lee, and Hansol Ryu. Cryptanalysis of the new CLT multilinear maps. Cryptology ePrint Archive, Report 2015/934, 2015. `http://eprint.iacr.org/2015/934`.

[CLT13] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Practical multilinear maps over the integers. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 476–493, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Heidelberg, Germany.

[CLT15] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. New multilinear maps over the integers. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 267–286, Santa Barbara, CA, USA, August 16–20, 2015. Springer, Heidelberg, Germany.

[CMR98] Ran Canetti, Daniele Micciancio, and Omer Reingold. Perfectly one-way probabilistic hash functions (preliminary version). In *30th Annual ACM Symposium on Theory of Computing*, pages 131–140, Dallas, Texas, USA, May 23–26, 1998. ACM Press.

[CRV10] Ran Canetti, Guy N. Rothblum, and Mayank Varia. Obfuscation of hyperplane membership. In Daniele Micciancio, editor, *TCC 2010: 7th Theory of Cryptography Conference*, volume 5978 of *Lecture Notes in Computer Science*, pages 72–89, Zurich, Switzerland, February 9–11, 2010. Springer, Heidelberg, Germany.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[DPSZ11] I. Damgard, V. Pastro, N.P. Smart, and S. Zakarias. Multiparty computation from somewhat homomorphic encryption. Cryptology ePrint Archive, Report 2011/535, 2011. `http://eprint.iacr.org/`.

[Gen09] Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. `crypto.stanford.edu/craig`.

[GGH13a] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 1–17, Athens, Greece, May 26–30, 2013. Springer, Heidelberg, Germany.

[GGH$^+$13b] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th Annual Symposium on Foundations of Computer Science*, pages 40–49, Berkeley, CA, USA, October 26–29, 2013. IEEE Computer Society Press.

[GGH$^+$13c] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. Cryptology ePrint Archive, Report 2013/451, 2013. `http://eprint.iacr.org/`.

[GGH15] Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015: 12th Theory of Cryptography Conference, Part II*, volume 9015 of *Lecture Notes in Computer Science*, pages 498–527, Warsaw, Poland, March 23–25, 2015. Springer, Heidelberg, Germany.

[GGHR14] Sanjam Garg, Craig Gentry, Shai Halevi, and Mariana Raykova. Two-round secure MPC from indistinguishability obfuscation. In Yehuda Lindell, editor, *TCC 2014: 11th Theory of Cryptography Conference*, volume 8349 of *Lecture Notes in Computer Science*, pages 74–94, San Diego, CA, USA, February 24–26, 2014. Springer, Heidelberg, Germany.

[GGSW13] Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th Annual ACM Symposium on Theory of Computing*, pages 467–476, Palo Alto, CA, USA, June 1–4, 2013. ACM Press.

[GH10] Craig Gentry and Shai Halevi. Implementing gentry's fully-homomorphic encryption scheme. Cryptology ePrint Archive, Report 2010/520, 2010. `http://eprint.iacr.org/`.

[GKP$^+$13] Shafi Goldwasser, Yael Tauman Kalai, Raluca A. Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. How to run turing machines on encrypted data. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 536–553, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Heidelberg, Germany.

[GLSW15] Craig Gentry, Allison Bishop Lewko, Amit Sahai, and Brent Waters. Indistinguishability obfuscation from the multilinear subgroup elimination assumption. In Venkatesan Guruswami, editor, *56th Annual Symposium on Foundations of Computer Science*, pages 151–170, Berkeley, CA, USA, October 17–20, 2015. IEEE Computer Society Press.

[GPS15] Sanjam Garg, Omkant Pandey, and Akshayaram Srinivasan. On the exact cryptographic hardness of finding a nash equilibrium. Cryptology ePrint Archive, Report 2015/1078, 2015. `http://eprint.iacr.org/2015/1078`.

[GPSW06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 06: 13th Conference on Computer and Communications Security*, pages 89–98, Alexandria, Virginia, USA, October 30 – November 3, 2006. ACM Press. Available as Cryptology ePrint Archive Report 2006/309.

[GPSZ16] Sanjam Garg, Omkant Pandey, Akshayaram Srinivasan, and Mark Zhandry. Breaking the sub-exponential barrier in obfustopia. Cryptology ePrint Archive, Report 2016/102, 2016. `http://eprint.iacr.org/2016/102`.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th Annual ACM Symposium on Theory of Computing*, pages 197–206, Victoria, British Columbia, Canada, May 17–20, 2008. ACM Press.

[GR07] Shafi Goldwasser and Guy N. Rothblum. On best-possible obfuscation. In Salil P. Vadhan, editor, *TCC 2007: 4th Theory of Cryptography Conference*, volume 4392 of *Lecture Notes in Computer Science*, pages 194–213, Amsterdam, The Netherlands, February 21–24, 2007. Springer, Heidelberg, Germany.

[Had10] Satoshi Hada. Secure obfuscation for encrypted signatures. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 92–112, French Riviera, May 30 – June 3, 2010. Springer, Heidelberg, Germany.

[HMLS07] Dennis Hofheinz, John Malone-Lee, and Martijn Stam. Obfuscation for cryptographic purposes. In Salil P. Vadhan, editor, *TCC 2007: 4th Theory of Cryptography Conference*, volume 4392 of *Lecture Notes in Computer Science*, pages 214–232, Amsterdam, The Netherlands, February 21–24, 2007. Springer, Heidelberg, Germany.

[HRsV07] Susan Hohenberger, Guy N. Rothblum, abhi shelat, and Vinod Vaikuntanathan. Securely obfuscating re-encryption. In Salil P. Vadhan, editor, *TCC 2007: 4th Theory of Cryptography Conference*, volume 4392 of *Lecture Notes in Computer Science*, pages 233–252, Amsterdam, The Netherlands, February 21–24, 2007. Springer, Heidelberg, Germany.

[Jan96] Gerald J. Janusz. *Algebraic Number Fields.* American Mathematical Society, 1996.

[JHC16] Changmin Lee Jung Hee Cheon, Jinhyuck Jeong. An algorithm for ntru problems and cryptanalysis of the ggh multilinear map without an encoding of zero. Cryptology ePrint Archive, Report 2016/139, 2016. `http://eprint.iacr.org/`.

[Jou00] Antoine Joux. A one round protocol for tripartite diffie-hellman. In *Algorithmic Number Theory - ANTS'00*, volume 1838 of *Lecture Notes in Computer Science*, pages 385–394. Springer, 2000.

[Lan90] S. Lang. *Cyclotomic Fields I and II: With and Appendix by Karl Rudin.* Graduate Texts in Mathematics. Springer-Verlag, 1990.

[LM06] Vadim Lyubashevsky and Daniele Micciancio. Generalized compact Knapsacks are collision resistant. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *ICALP 2006: 33rd International Colloquium on Automata, Languages and Programming, Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 144–155, Venice, Italy, July 10–14, 2006. Springer, Heidelberg, Germany.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 1–23, French Riviera, May 30 – June 3, 2010. Springer, Heidelberg, Germany.

[LPR12] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. Cryptology ePrint Archive, Report 2012/230, 2012. `http://eprint.iacr.org/`.

[LPS04] Ben Lynn, Manoj Prabhakaran, and Amit Sahai. Positive results and techniques for obfuscation. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 20–39, Interlaken, Switzerland, May 2–6, 2004. Springer, Heidelberg, Germany.

[MA16] Lo Ducas Martin Albrecht, Shi Bai. A subfield lattice attack on overstretched ntru assumptions: Cryptanalysis of some fhe and graded encoding schemes. Cryptology ePrint Archive, Report 2016/127, 2016. `http://eprint.iacr.org/`.

[MR07] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Computing*, 37(1):267–302, 2007.

[MSW14] Eric Miles, Amit Sahai, and Mor Weiss. Protecting obfuscation against arithmetic attacks. Cryptology ePrint Archive, Report 2014/878, 2014. `http://eprint.iacr.org/2014/878`.

[MSZ16] Eric Miles, Amit Sahai, and Mark Zhandry. Annihilation attacks for multilinear maps: Cryptanalysis of indistinguishability obfuscation over ggh13. Cryptology ePrint Archive, Report 2016/147, 2016. `http://eprint.iacr.org/`.

[O'N10] Adam O'Neill. Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556, 2010. `http://eprint.iacr.org/`.

[Oss08] Brian Osserman. *Algebraic Number Theory*. Lecture Notes, 2008. `https://www.math.ucdavis.edu/~osserman/classes/numthy/numthybook.pdf`.

[Reg04] Oded Regev. New lattice-based cryptographic constructions. *J. ACM*, 51(6):899–942, 2004.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signature and public-key cryptosystems. *Communications of the Association for Computing Machinery*, 21(2):120–126, 1978.

[Rud89] Steven Rudich. Unpublished, 1989.

[Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, *Advances in Cryptology – CRYPTO'84*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53, Santa Barbara, CA, USA, August 19–23, 1984. Springer, Heidelberg, Germany.

[SL96] Peter Stevenhagen and Hendrik W Lenstra. Chebotarëv and his density theorem. *The Mathematical Intelligencer*, 18(2):26–37, 1996.

[SS11] Damien Stehlé and Ron Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 27–47, Tallinn, Estonia, May 15–19, 2011. Springer, Heidelberg, Germany.

[Ste04] William Stein. *A Brief Introduction to Classical and Adelic Algebraic Number Theory*. 2004. `http://modular.math.washington.edu/129/ant/ant.pdf`.

[Ste10] C. L. Stewart. On divisors of lucas and lehmer numbers. 2010.

[SV10] Nigel P. Smart and Frederik Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In Phong Q. Nguyen and David Pointcheval, editors, *PKC 2010: 13th International Conference on Theory and Practice of Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 420–443, Paris, France, May 26–28, 2010. Springer, Heidelberg, Germany.

[SW05] Amit Sahai and Brent R. Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473, Aarhus, Denmark, May 22–26, 2005. Springer, Heidelberg, Germany.

[SW13] Alice Silverberg and Lawrence Washingoton, 2013. Personal Communication.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In David B. Shmoys, editor, *46th Annual ACM Symposium on Theory of Computing*, pages 475–484, New York, NY, USA, May 31 – June 3, 2014. ACM Press.

[Was97] L.C. Washington. *Introduction to Cyclotomic Fields*. Graduate Texts in Mathematics. Springer-Verlag, 1997.

[Wee05] Hoeteck Wee. On obfuscating point functions. In Harold N. Gabow and Ronald Fagin, editors, *37th Annual ACM Symposium on Theory of Computing*, pages 523–532, Baltimore, Maryland, USA, May 22–24, 2005. ACM Press.

[Wes99] Tom Weston. *Algebraic Number Theory*. Course Notes, 1999. `https://www.math.umass.edu/~weston/cn/notes.pdf`.