Noninteractive Zero Knowledge for NP from (Plain) Learning With Errors

Chris Peikert[*] Sina Shiehian[†]

February 15, 2019

Abstract

We finally close the long-standing problem of constructing a noninteractive zero-knowledge (NIZK) proof system for any NP language with security based on the *plain Learning With Errors* (LWE) problem, and thereby on worst-case lattice problems. Our proof system instantiates the framework recently developed by Canetti *et al.* [EUROCRYPT'18], Holmgren and Lombardi [FOCS'18], and Canetti *et al.* [STOC'19] for soundly applying the Fiat–Shamir transform using a hash function family that is *correlation intractable* for a suitable class of relations. Previously, such hash families were based either on "exotic" assumptions (e.g., indistinguishability obfuscation or optimal hardness of certain LWE variants) or, more recently, on the existence of circularly secure fully homomorphic encryption (FHE). However, none of these assumptions are known to be implied by plain LWE or worst-case hardness.

Our main technical contribution is a hash family that is correlation intractable for arbitrary size-S circuits, for any polynomially bounded S, based on plain LWE (with small polynomial approximation factors). The construction combines two novel ingredients: a correlation-intractable hash family for *log-depth* circuits based on LWE (or even the potentially harder Short Integer Solution problem), and a "bootstrapping" transform that uses (leveled) FHE to promote correlation intractability for the FHE decryption circuit to *arbitrary* (bounded) circuits. Our construction can be instantiated in two possible "modes," yielding a NIZK that is either *computationally* sound and *statistically* zero knowledge in the common *random* string model, or vice-versa in the common *reference* string model.

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

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

1 Introduction

A *zero-knowledge* (ZK) proof system [GMR85] is a protocol by which a prover can convince a verifier that a particular statement is true, while revealing nothing more than that fact. Such a system is *noninteractive* [BDMP88] (NIZK) if both parties have access to some common string (e.g., a public source of randomness), and the prover just sends a single message to the verifier. In the three decades since the introduction of NIZK, several works have constructed such protocols for *arbitrary NP languages* based on various cryptographic structures (such as quadratic residuosity, bilinear pairings, and code obfuscation) [BFM88, GO94, FLS90, GOS06, SW14], and used them in a variety of important cryptographic settings, like encryption that withstands chosen-ciphertext attacks [BFM88, NY90], digital signatures [BG89], ZAPs [DN00] cryptocurrencies [BCG$^+$14], and low-interaction protocols in general.

In recent years, cryptography based on *lattices* has seen enormous growth. Among its attractions are apparent resistance to quantum attacks, advanced functionality like fully homomorphic encryption (FHE) [Gen09], and strong theoretical guarantees like security under *worst-case* hardness assumptions, usually via the well-known Short Integer Solution (SIS) [Ajt96] and Learning With Errors (LWE) problems [Reg05]. Yet while (non-)interactive zero-knowledge protocols for *specific* lattice problems have been known for some time [MV03, PV08, CDLP13], the goal of obtaining NIZK for *general NP languages* based on standard, worst-case lattice assumptions (which was explicitly posed in [PV08]) has frustratingly remained out of reach. The past year has seen impressive progress toward this goal [KW18, CCRR18, CCH$^+$19], but the current constructions either satisfy a relaxed notion of NIZK or are based on assumptions that are not yet known to be implied by worst-case hardness.

More specifically, a fascinating recent line of research [KRR17, CCRR18, HL18, CCH$^+$19] develops a framework for instantiating the Fiat–Shamir transform [FS86], which removes interaction from a public-coin protocol by replacing each random verifier message with a hash of the transcript so far. In particular, these works show that if the hash function satisfies a property called *correlation intractability* [CGH98], then the Fiat–Shamir transform can be applied soundly to many interactive protocols, including some zero-knowledge ones. Roughly speaking, a hash family H is correlation intractable for a relation R if, given a hash key k, it is hard to find an input-output pair $(x, H_k(x)) \in R$. In the context of Fiat–Shamir, this ensures that a cheating prover cannot find a message that hashes to a verifier message that admits an accepting transcript.

The works [CCRR18, HL18, CCH$^+$19] construct correlation-intractable hash functions for various *sparse* relations, and use them to soundly instantiate the Fiat–Shamir transform, obtaining NIZK proofs for all of NP (among other results). Of particular interest is the beautiful work of [CCH$^+$19], which shows that for this purpose, it suffices to have correlation intractability for arbitrary (bounded) *polynomial-time computations*, i.e., for the special class of *efficiently searchable* relations. These are relations where each input has at most a single output (witness) that is computable within some desired polynomial time bound.

The hash families constructed in [CCRR18, CCH$^+$19] are proved to be correlation intractable under various lattice-related assumptions. However, these assumptions are somehow non-standard, involving either "optimal hardness" (e.g., of LWE with uniform error in an interval) against polynomial-time attacks [CCRR18, CCH$^+$19], or the existence of circularly secure FHE [CCH$^+$19]. Although the latter assumption seems tantalizingly close to plain LWE (and remains the only known way of obtaining FHE that supports *unbounded*, as opposed to just *leveled*, homomorphic computations), none of these assumptions are known to be supported by the hardness of LWE, nor the conjectured worst-case hardness of lattice problems.

1.1 Contributions

Our main result is a noninteractive zero-knowledge proof system for any NP language, based on the *plain LWE* problem with *(small) polynomial* approximation factors. This finally closes (following much recent progress) the central open problem of basing NIZK for NP on worst-case lattice assumptions. Our system instantiates the NIZK framework recently developed in [CCRR18, CCH$^+$19], but with a new primary ingredient: a correlation-intractable hash family for arbitrary size-S circuits (i.e., relations searchable in size S), for any desired $S = \mathrm{poly}(\lambda)$, based on plain LWE with small polynomial factors.

Just like the correlation-intractable hash family constructed in [CCH$^+$19], ours also can be instantiated in two "intractability modes," *computational* and *statistical*, by constructing the hash key in one of two computationally indistinguishable ways. In the statistical mode, input-output pairs that satisfy the relation simply *do not exist* (so obviously one cannot be found); in the computational mode, the hash key is *uniformly random* and security can be based merely on *SIS*, a potentially harder problem for which we have even stronger hardness theorems than for LWE. In either case, these are the first known construction of CI hash families for "rich" functions from plain LWE/SIS, or any worst-case lattice assumption. As shown in [CCH$^+$19], the choice of intractability mode determines the precise properties of the NIZK system: computational mode yields a *statistically* zero knowledge, (*adaptively*) *computationally* sound (i.e., argument) system in the common *random* string model, while statistical mode yields a *computationally* zero knowledge, *statistically sound* (i.e., proof) system in the common *reference* string model.

Our correlation-intractable hash family for bounded circuits is obtained by combining two new ingredients that are interesting in their own right:

1. a correlation-intractable hash family for bounded circuits based on plain SIS/LWE, where in particular for *log-depth* circuits the associated approximation factor is a (small) *polynomial*; and

2. a "bootstrapping" transform that uses (leveled) fully homomorphic encryption to promote CI for the FHE *decryption* circuit to CI for *arbitrary bounded* circuits. (This transformation is inspired by other bootstrapping techniques for code obfuscation [GGH$^+$13], and is in some sense dual to Gentry's bootstrapping technique for FHE [Gen09].)

In particular, a suitable FHE scheme having log-depth decryption can be instantiated based on plain LWE with small polynomial factors [BV14, AP14], which yields our ultimate LWE-based CI hash family.

1.2 Discussion and Open Problems

We make a few additional remarks about our constructions, and list some open problems for further research. Just as in [CCH$^+$19], our CI hash family also suffices for proving that the parallelized quadratic residuosity protocol of [GMR85] is *not* zero knowledge (assuming that QR is not in BPP), but now under plain SIS/LWE assumptions instead of circularly secure FHE.

We emphasize that our CI hash family is *non-compact*: the size of the hash key grows with the size S of the circuits for which it is correlation intractable. This property is shared by all other prior constructions except those based on highly "exotic" assumptions like indistinguishability obfuscation or optimal key-dependent message security, e.g., [KRR17, CCRR18, CCH$^+$19]. A non-compact construction based on standard assumptions would be very interesting, and presumably quite powerful.

Our SIS-based CI hash family works for circuits of any depth, but is only supported by polynomial SIS factors for *log-depth* circuits. Dealing with deeper circuits while retaining polynomial approximation factors requires us to use our bootstrapping theorem with (leveled) FHE, which brings in the LWE assumption. (In addition, the NIZK construction also uses LWE for lossy encryption.) It is an interesting open problem to get a CI hash family for super-logarithmic depth based on just SIS with polynomial factors.

1.3 Techniques

Here we summarize the main ideas and techniques underlying our constructions.

1.3.1 Bootstrapping Correlation Intractability

In Section 4 we give a generic transform that uses (leveled) fully homomorphic encryption to convert a correlation-intractable hash family for "simple" relations related to the FHE decryption function, into one for complex relations induced by circuits of any size S. For simplicity, here we focus on correlation intractability for *functions* f, i.e., for searchable relations $R_f = \{(x, f(x))\}$, but everything easily generalizes to more general relations.

Let $\mathsf{FHE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ denote a (symmetric-key) fully homomorphic encryption scheme.[1] Let $\mathsf{CIH} = (\mathsf{Gen}, \mathsf{Hash})$ denote a hash family that is correlation intractable for the class $\{\mathsf{Dec}_{sk}(\cdot)\}$ of FHE decryption functions, taken over all valid "hard-wired" secret keys. We define a new hash family $\mathsf{CIH}' = (\mathsf{Gen}', \mathsf{Hash}')$ for circuits of size S as follows:

- $\mathsf{Gen}'(1^\lambda)$ generates a CIH key $k \leftarrow \mathsf{CIH}.\mathsf{Gen}(1^\lambda)$, an FHE key pair $(sk, ek) \leftarrow \mathsf{FHE}.\mathsf{Gen}(1^\lambda)$, and a "dummy" ciphertext $c \leftarrow \mathsf{Enc}(sk, 0^S)$. It outputs the hash key $k' = (k, ek, c)$.
- $\mathsf{Hash}'(k' = (k, ek, c), x)$ outputs
$$\mathsf{Hash}(k, \mathsf{Eval}(ek, U_x, c)),$$
where $U_x(\cdot) = U(\cdot, x)$ is a universal circuit for size-S circuits with x "hard-coded" in.

In words, Hash' homomorphically evaluates an encrypted (dummy) circuit on the input x, then hashes the resulting ciphertext using the underlying Hash algorithm.

We now sketch why CIH' is correlation intractable for any function f having circuit size S. As a thought experiment, imagine replacing the "dummy" ciphertext with $c \leftarrow \mathsf{Enc}(sk, f)$. By the security of the FHE scheme, this does not noticeably change the probability that the adversary, given the key $k' = (k, ek, c)$, can find an input x that violates correlation intractability of $\mathsf{Hash}'(k', \cdot)$ for f, i.e.,

$$\mathsf{Hash}'(k', x) = \mathsf{Hash}(k, \underbrace{\mathsf{Eval}(ek, U_x, c)}_{c_x}) = f(x).$$

Suppose for the purpose of contradiction that the adversary is able to find such an x. Then because c_x is an FHE encryption of $f(x)$ by construction, we have $\mathsf{Hash}(k, c_x) = f(x) = \mathsf{Dec}_{sk}(c_x)$. Therefore, we have found an input c_x that violates the correlation intractability of $\mathsf{Hash}(k, \cdot)$ for the function Dec_{sk}, which is the desired contradiction.[2]

[1] For simplicity, here we assume that FHE supports unbounded, not just leveled, homomorphic evaluation. Adapting the construction to leveled FHE is straightforward because Eval is used only on circuits of bounded depth.

[2] The astute reader might notice that the specific function Dec_{sk} is not fixed in advance, but is instead chosen at random by the reduction. This is addressed in the non-uniform setting by "fixing coins" for $\mathsf{FHE}.\mathsf{Gen}$ that maximize the probability that the Hash'-attacker finds a suitable x, or in the uniform setting by adopting a security definition that allows the adversary to declare a (valid) target function before receiving the hash key.

1.3.2 Correlation Intractability from SIS/LWE

In Section 3 we construct a public-coin, correlation-intractable hash family for arbitrary functions of bounded circuit size based on plain SIS, with a complementary *statistically* intractable mode based on LWE. Our construction works for arbitrary functions, and the circuit size, depth, and output length induce corresponding SIS/LWE parameters. More specifically, the dimension n grows linearly in the output length, and the approximation factor (and hence modulus q) grows exponentially with the depth and polynomially with the size. Due to our bootstrapping transformation, the main parameterization of interest is *log-depth* circuits, for which the approximation factors can be made (small) polynomials.

Our construction is based upon the fully homomorphic *commitment* scheme implicit in GSW homomorphic encryption [GSW13], and made explicit in subsequent work on fully homomorphic signatures [GVW15]. The construction works essentially as follows:

- A hash key is a commitment $k = \widehat{D}$ to a "dummy" circuit D of the desired output length L and size S.
- To evaluate the hash function at an input x, we first homomorphically evaluate a commitment $\widehat{D(x)}$ of $D(x)$. We then homomorphically apply a certain fixed, public linear function G from $\{0,1\}^L$ to the SIS/LWE range $\mathbb{Z}_q^n$ to get a "quasi commitment" $c_x = \widehat{G(D(x))}$ that is itself in $\mathbb{Z}_q^n$. Finally, we output $\text{bin}(c_x)$, the binary representation (in $\{0,1\}^L$) of c_x.

The linear function G just needs to satisfy $G(\text{bin}(\mathbf{u})) = \mathbf{u}$ for all $\mathbf{u} \in \mathbb{Z}_q^n$. (This implies that G is surjective, so the circuit output length must be at least $n \log q$.) For example, G can map each of n groups of $\ell = \lceil \lg q \rceil$ bits to the mod-q integer they represent in binary.[3]

The construction is structurally similar to the one from [CCH$^+$19]; the main difference is that after computing $\widehat{D(x)}$, instead of homomorphically evaluating (the complement of) a secret decryption circuit on the result, we "quasi-homomorphically" evaluate the public linear function G. Indeed, the former is what leads [CCH$^+$19] to need circularly secure FHE. (And because we no longer need a decryption circuit, fully homomorphic commitment is enough for our purposes.) Most importantly, the quasi-commitment c_x does not just hide a single *bit* (i.e., the output of decryption), but an entire *vector* $G(D(x)) \in \mathbb{Z}_q^n$ *that is the same size as* c_x *itself.* In the security proof this allows us to reason about the hash output directly, instead of how it behaves under decryption (as in [CCH$^+$19]).

Security is argued as follows, where for the moment we focus on the proof from SIS. Suppose that an adversary is able to violate correlation intractability for some function f of size S and output length L, i.e., given a hash key it finds an input x that hashes to $f(x)$. By the (statistical) security of the commitment scheme, the adversary has essentially the same probability of succeeding if the hash key is a commitment $\widehat{f}$ to f. When it does succeed we have $\text{bin}(\widehat{G(f(x))}) = f(x)$, and so by applying G to both sides we get

$$\widehat{G(f(x))} = G(f(x)) \in \mathbb{Z}_q^n. \tag{1.1}$$

To see why this yields an SIS solution, we need to understand the particular form of the commitments in a little more detail. All commitments are with respect to a random SIS matrix $\mathbf{A}$ over $\mathbb{Z}_q$. The commitment scheme has the property that, given the randomness used to form the original commitment $\widehat{f}$, it is possible to efficiently compute randomness that is consistent with the homomorphically evaluated commitment $\widehat{D(x)}$, and likewise for the quasi-commitment $c_x = \widehat{G(f(x))}$. Concretely, this derived randomness is a *short* integer vector $\mathbf{r}$ such that

$$\widehat{G(f(x))} = \mathbf{A}\mathbf{r} + G(f(x)) \pmod{q}.$$

[3] Those familiar with the literature will recognize this as the linear transform induced by the "gadget" matrix $\mathbf{G}$.

But because $\widehat{G(f(x))} = G(f(x))$ by Equation (1.1), it follows that $\mathbf{Ar} = \mathbf{0} \in \mathbb{Z}_q^n$. Therefore, the short vector $\mathbf{r}$ is a solution to the SIS problem for the random instance $\mathbf{A}$, as desired. (We also need to ensure that $\mathbf{r}$ is nonzero; this can be done with standard techniques.)

To get *statistical* correlation intractability based on LWE, we need to slightly tweak the construction, defining the hash function to evaluate a quasi-commitment $c_x = \widehat{G(D(x))} + \lfloor q/2 \rceil \mathbf{u}_n$, where $\mathbf{u}_n$ is the nth standard basis vector.[4] For a particular f of interest, we again replace the commitment to D with one to f. Then, to get a hash key for which an x that hashes to $f(x)$ simply *does not exist*, we switch $\mathbf{A}$ to be an LWE matrix whose bottom row $\mathbf{b}^t$ is a noisy linear combination of the others, i.e., $\mathbf{b}^t = \mathbf{s}^t\mathbf{A}' + \mathbf{e}^t$ where $\mathbf{A}'$ consists of the top $n-1$ rows of $\mathbf{A}$ and $\mathbf{e}$ is a "short" error vector; by the LWE assumption, this change is indistinguishable to the attacker. Much like above, a hypothetical input x which hashes to $f(x)$ now yields $\mathbf{Ar} = -\lfloor q/2 \rceil \mathbf{u}_n$, which implies that $\mathbf{A}'\mathbf{r} = \mathbf{0}$ and hence

$$-\lfloor q/2 \rceil = \mathbf{b}^t \cdot \mathbf{r} = (\mathbf{s}^t\mathbf{A}' + \mathbf{e}^t)\mathbf{r} = \mathbf{s}^t(\mathbf{A}'\mathbf{r}) + \mathbf{e}^t \cdot \mathbf{r} = \langle \mathbf{e}, \mathbf{r} \rangle \pmod q.$$

But because both $\mathbf{e}$ and $\mathbf{r}$ are relatively short, by taking q to be large enough this equation simply cannot hold, hence no such x exists.

Acknowledgments. We thanks Alex Lombardi and Daniel Wichs for useful comments on our constructions.

2 Preliminaries

We denote column vectors by lower-case bold letters, e.g., $\mathbf{a}$. We denote matrices by upper-case bold letters, e.g., $\mathbf{A}$. For integral vectors and matrices (i.e., those over $\mathbb{Z}$), we use the notation $|\mathbf{r}|, |\mathbf{R}|$ to denote the maximum absolute value over all the entries.

The Kronecker product $\mathbf{A} \otimes \mathbf{B}$ of two matrices (or vectors) $\mathbf{A}$ and $\mathbf{B}$ is obtained by replacing each entry $a_{i,j}$ of $\mathbf{A}$ with the block $a_{i,j}\mathbf{B}$. This obeys the *mixed-product* property: $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD})$ for any matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ with compatible dimensions.

2.1 Noninteractive Zero Knowledge

Definition 2.1. Let R be a relation. A *noninteractive proof system* for R is a tuple of PPT algorithms $(\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ having the following interfaces (where $1^n, 1^\lambda$ are implicit inputs to Prove, Verify):

- $\mathsf{Setup}(1^n, 1^\lambda)$, given a statement length n and a security parameter λ, outputs a string σ.
- $\mathsf{Prove}(\sigma, x, w)$, given a string σ and a statement-witness pair $(x, w) \in R$, outputs a proof π.
- $\mathsf{Verify}(\sigma, x, \pi)$, given a string σ, a statement x, and a proof π, either accepts or rejects.

Definition 2.2. Let $\Pi = (\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ be a noninteractive proof system for a relation R, and let L be the language defined by R. In this work we focus on systems that satisfy some subset of the following properties:

1. *Completeness*: for every $(x, w) \in R$ and every $\lambda \in \mathbb{N}$, $\mathsf{Verify}(\sigma, x, \pi)$ accepts with probability 1, over the choice of $\sigma \leftarrow \mathsf{Setup}(1^{|x|}, 1^\lambda)$ and $\pi \leftarrow \mathsf{Prover}(\sigma, x, w)$.

2. *Common random string*: $\mathsf{Setup}(1^n, 1^\lambda)$ simply outputs a uniformly random string.

[4] With this change, the SIS-based proof still goes through, but for $n-1$ dimensions, by ignoring the last row of $\mathbf{A}$.

3. *Statistical soundness*: there exists a negligible function $\nu(\lambda)$ such that for any $n \in \mathbb{N}$,

$$\Pr_{\sigma \leftarrow \mathsf{Setup}(1^n, 1^\lambda)}[\exists (x, \pi^*) \text{ s.t. } \mathsf{Verify}(\sigma, x, \pi^*) \text{ accepts} \wedge x \notin L] \leq \nu(\lambda).$$

4. *Adaptive soundness*: for every non-uniform polynomial-size "cheating" prover $P^* = \{P^*_\lambda\}$ there exists a negligible function $\nu(\lambda)$ such that for any $n \in \mathbb{N}$,

$$\Pr_{\substack{\sigma \leftarrow \mathsf{Setup}(1^n, 1^\lambda) \\ (x, \pi^*) = P^*_\lambda(\sigma)}}[\mathsf{Verify}(\sigma, x, \pi^*) \text{ accepts} \wedge x \notin L] \leq \nu(\lambda).$$

5. *Statistical zero knowledge*: there exists a PPT simulator $\mathcal{S}$ such that for every $(x, w) \in R$ the following two distribution ensembles are statistically indistinguishable:

$$\{\mathcal{S}(1^\lambda, x)\}_\lambda \overset{s}{\approx} \{(\sigma, \mathsf{Prover}(\sigma, x, w)) : \sigma \leftarrow \mathsf{Setup}(1^{|x|}, 1^\lambda)\}_\lambda \ .$$

6. *Adaptive (computational) zero knowledge*: there exists a PPT simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ such that for every non-uniform polynomial-size "cheating" verifier $V^* = (V^*_1, V^*_2)$, for every $n \in \mathbb{N}$ the probabilities

$$\Pr[V^*_2(\sigma, x, \pi, \zeta) = 1 \wedge (x \in L)]$$

in the following two experiments differ only by $\mathrm{negl}(\lambda)$:

- in the "real" experiment, $\sigma \leftarrow \mathsf{Setup}(1^n, 1^\lambda), (x, w, \zeta) \leftarrow V^*_1(\sigma), \pi \leftarrow \mathsf{Prove}(\sigma, x, w)$;
- in the "simulation" experiment, $(\sigma, \tau) \leftarrow \mathcal{S}_1(1^\lambda), (x, w, \zeta) \leftarrow V^*_1(\sigma), \pi \leftarrow \mathcal{S}_2(\sigma, x, \tau)$.

2.2 Correlation Intractability

As in [CCH$^+$19] we define efficiently searchable relations and recall the definitions of correlation intractability, in their computational and statistical versions.

Definition 2.3. We say that a relation $R \subseteq \mathcal{X} \times \mathcal{Y}$ is *searchable in size* S if there exists a function $f : \mathcal{X} \to \mathcal{Y}$ that is implementable as a boolean circuit of size S, such that if $(x, y) \in R$ then $y = f(x)$. (In other words, $f(x)$ is the unique witness for x, if such a witness exists.)

Definition 2.4. Let $\mathcal{R} = \{\mathcal{R}_\lambda\}$ be a relation class, i.e., a set of relations for each λ. A hash function family $(\mathsf{Gen}, \mathsf{Hash})$ is *correlation intractable (CI) for* $\mathcal{R}$ if for every non-uniform polynomial-size adversary $\mathcal{A} = \{\mathcal{A}_\lambda\}$ there exists a negligible function $\nu(\lambda)$ such that for every $R \in \mathcal{R}_\lambda$

$$\Pr_{\substack{k \leftarrow \mathsf{Gen}(1^\lambda) \\ x = \mathcal{A}_\lambda(k)}}[(x, \mathsf{Hash}(k, x)) \in R] \leq \nu(\lambda) \ .$$

Definition 2.5. Let $\mathcal{R} = \{\mathcal{R}_\lambda\}$ be a relation class. A hash function family $(\mathsf{Gen}, \mathsf{Hash})$ with a fake-key generation algorithm $\mathsf{StatGen}$ is *somewhere statistically correlation intractable for* $\mathcal{R}$ if

1. $\mathsf{StatGen}(1^\lambda, z)$, where z is an auxiliary input, outputs a key k,
2. there exists a negligible function $\nu(\lambda)$ and a class of auxiliary inputs $\mathcal{Z} = \{\mathcal{Z}_\lambda\}$ such that

- the distribution ensembles $\{\mathsf{StatGen}(1^\lambda, z_\lambda)\}$ and $\{\mathsf{Gen}(1^\lambda)\}$ are computationally indistinguishable for every sequence of $z_\lambda \in \mathcal{Z}_\lambda$, and
- for every $R \in \mathcal{R}_\lambda$ there exists $z_R \in \mathcal{Z}_\lambda$ such that

$$\Pr_{k \leftarrow \mathsf{StatGen}(1^\lambda, z_R)} [\exists x \text{ s.t. } (x, \mathsf{Hash}(k, x)) \in R] \leq \nu(\lambda) \ .$$

We call z_R the *intractability guarantee* for R.

2.3 (Leveled) Fully Homomorphic Encryption

We recall the notion of leveled FHE from [Gen09].

Definition 2.6. A *leveled fully homomorphic encryption* scheme is a tuple of algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ with the following interfaces (we use only a symmetric-key version, which is sufficient for our purposes):

- $\mathsf{Gen}(1^\lambda, 1^d)$ outputs a secret key sk and an evaluation key ek.
- $\mathsf{Enc}(sk, m \in \{0,1\}^*)$, where m is a message, outputs a ciphertext c.
- $\mathsf{Eval}(C, c)$, where C is a boolean circuit of depth (at most) d, deterministically outputs a ciphertext c'.
- $\mathsf{Dec}(sk, c)$ outputs a message (deterministically).

It should satisfy the following properties:

1. *Completeness*: For any circuit C of depth at most d and message m, $\mathsf{Dec}(sk, \mathsf{Eval}(C, c)) = C(m)$ with probability 1, over the random choice of $sk \leftarrow \mathsf{Gen}(1^\lambda, 1^d)$ and $c \leftarrow \mathsf{Enc}(sk, m)$.

2. *CPA security*: for any sequence of message pairs $\{(m_{0,\lambda}, m_{1,\lambda})\}_\lambda$ where $|m_{0,\lambda}| = |m_{1,\lambda}|$, and any sequence $\{d_\lambda\}$, the distribution ensembles

$$\{\mathsf{Enc}(sk, m_{b,\lambda}) : sk \leftarrow \mathsf{Gen}(1^\lambda, 1^{d_\lambda})\}_\lambda$$

   are computationally indistinguishable for $b = 0, 1$.

3. *Compactness*: the complexity of Dec is a fixed polynomial in λ alone. (This implies that the output of Eval has a fixed polynomial size in λ alone, and does not depend on the evaluated circuit or d.)

2.4 Branching Programs

A width-w boolean permutation branching program BP of length L with input space $\{0,1\}^\ell$ consists of $2L$ permutations $\{\pi_{i,b} \colon [w] \to [w]\}_{i \in [L], b \in \{0,1\}}$ along with an index-to-input map $v : [L] \to [\ell]$. To compute the output of BP on an input $x \in \{0,1\}^\ell$ we first initialize a state variable $st_0 = 1$. Then, for each $i \in [L]$ we set $st_i = \pi_{i, x_{v(i)}}(st_{i-1})$. Finally, if $st_L = 1$ we output 1; otherwise, we output 0. More generally, a branching program can have multi-bit output by just having a separate branching program for each output bit; its length is the maximum length of all the component programs.

Barrington's theorem [Bar86] states that every depth-d boolean circuit can be efficiently converted into a width-5 permutation branching program of length 4^d. In particular, any NC^1 circuit can be converted into a polynomial-length, constant-width permutation branching program.

2.5 Short Integer Solution and Learning With Errors

We recall the Short Integer Solution (SIS) and Learning With Errors (LWE) problems, and their hardness based on worst-case lattice problems.

Definition 2.7. The $\text{SIS}_{n,m,q,\beta}$ problem is: given a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, find a non-zero integral vector $\mathbf{z} \in \mathbb{Z}^m$ such that $\mathbf{A} \cdot \mathbf{z} = 0 \bmod q$ and $\|\mathbf{z}\| \leq \beta$.

We sometimes drop the subscript m when it is an unspecified polynomial in n and $\log q$. When $q \geq \beta \cdot \tilde{O}(\sqrt{n})$, solving $\text{SIS}_{n,q,\beta}$ is at least as hard as approximating certain worst-case lattice problems on n-dimensional lattices to within a $\beta \cdot \tilde{O}(\sqrt{n})$ factor [MR04].

For a positive integer dimension n and modulus q, and an error distribution χ over $\mathbb{Z}$, the LWE distribution and decision problem are defined as follows. For an $\mathbf{s} \in \mathbb{Z}^n$, the LWE distribution $A_{\mathbf{s},\chi}$ is sampled by choosing a uniformly random $\mathbf{a} \leftarrow \mathbb{Z}_q^n$ and an error term $e \leftarrow \chi$, and outputting $(\mathbf{a}, b = \langle \mathbf{s}, \mathbf{a} \rangle + e) \in \mathbb{Z}_q^{n+1}$. If we have m such samples $(\mathbf{a}_i, b_i)$, we can gather them as a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and vector $\mathbf{b}^t = \mathbf{s}^t\mathbf{A} + \mathbf{e}^t \in \mathbb{Z}_q^m$.

Definition 2.8. The $\text{LWE}_{n,m,q,\chi}$ problem is to distinguish, with non-negligible advantage, between m independent samples drawn from $A_{\mathbf{s},\chi}$ for a single $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, and m *uniformly random* and independent samples over $\mathbb{Z}_q^{n+1}$.

(As with SIS, we sometimes drop the subscript m.) A standard instantiation of LWE is to let χ be a *discrete Gaussian* distribution over $\mathbb{Z}$ with parameter $r = 2\sqrt{n}$. A sample drawn from this distribution has magnitude bounded by, say, $r\sqrt{n} = \Theta(n)$ except with probability at most 2^{-n}, and hence this tail of the distribution can be entirely removed. For this parameterization, it is known that LWE is at least as hard as *quantumly* approximating certain "short vector" problems on n-dimensional lattices, in the worst case, to within $\tilde{O}(q\sqrt{n})$ factors [Reg05, PRS17]. Classical reductions are also known for different parameterizations [Pei09, BLP$^+$13]. It is also well-known folklore that for such parameters, $\text{LWE}_{n,m,q,\chi}$ reduces to $\text{SIS}_{n,m,q,\beta}$ for every $\beta \leq q/r$.

2.6 Lattice Gadgets

Here we recall lattice "gadgets" [MP12] over $\mathbb{Z}_q$. For a positive integer modulus q, let $\ell = \lceil \lg q \rceil$. The "gadget" vector over $\mathbb{Z}_q$ is defined as

$$\mathbf{g}^t = (1, 2, 4, \ldots, 2^{\ell-1}) \in \mathbb{Z}_q^\ell.$$

For every $u \in \mathbb{Z}_q$, there is an efficiently computable binary vector $\mathbf{g}^{-1}[u] \in \{0,1\}^\ell$ such that $\langle \mathbf{g}, \mathbf{g}^{-1}[u] \rangle = u \pmod q$. Specifically, $\mathbf{g}^{-1}[u]$ corresponds to the binary representation of the distinguished representative of u in $\{0, 1, \ldots, q-1\}$. We stress that $\mathbf{g}^{-1} \colon \mathbb{Z}_q \to \{0,1\}^\ell$ is a *function*; its name reflects the essential property $\langle \mathbf{g}, \mathbf{g}^{-1}[u] \rangle = u$.

For a dimension n, the gadget matrix is defined as

$$\mathbf{G}_n = \mathbf{I}_n \otimes \mathbf{g}^t \in \mathbb{Z}_q^{n\times m},$$

where $m = n\ell$. We often drop the subscript n when it is clear from context. Similarly to above, we define the function $\mathbf{G}^{-1} = (\mathbf{I} \otimes \mathbf{g}^{-1}) \colon \mathbb{Z}_q^n \to \{0,1\}^m$, which applies $\mathbf{g}^{-1}$ to each coordinate and appends the results. This has the essential property, which is also reflective of the mixed-product property, that for every $\mathbf{u} \in \mathbb{Z}_q^n$,

$$\mathbf{G} \cdot \mathbf{G}^{-1}[\mathbf{u}] = (\mathbf{I} \otimes \mathbf{g}^t) \cdot (\mathbf{I} \otimes \mathbf{g}^{-1})[\mathbf{u}] = \mathbf{u}.$$

2.7 Fully Homomorphic Commitments

Here we recall the relevant homomorphic properties of gadgets, some of which were implicit in [GSW13], and which were developed and exploited further in [BGG$^+$14, BV14, AP14, GVW15]. We particularly focus on their application to fully homomorphic commitments, as laid out in [GVW15], and refer to that work for full details.

Let $\mathbf{A} \in \mathbb{Z}_q^{n \times w}$ be an arbitrary matrix for some dimension w. Let $\mathbf{C}_i = \mathbf{A}\mathbf{R}_i + x_i\mathbf{G}$ for some integral matrix $\mathbf{R}_i \in \mathbb{Z}^{w \times m}$ and scalar $x_i \in \mathbb{Z}_q$ for $i = 1, 2$. We view $\mathbf{C}_i$ as a commitment (relative to $\mathbf{A}$) to x_i under randomness $\mathbf{R}_i$. Observe that these commitments satisfy the following homomorphic properties:

$$\begin{aligned} \mathbf{G} - \mathbf{C}_1 &= \mathbf{A}(-\mathbf{R}_1) + (1 - x_1)\mathbf{G} \\ \mathbf{C}_+ := \mathbf{C}_1 + \mathbf{C}_2 &= \mathbf{A}(\underbrace{\mathbf{R}_1 + \mathbf{R}_2}_{\mathbf{R}_+}) + (x_1 + x_2)\mathbf{G} \\ \mathbf{C}_\times := \mathbf{C}_1 \cdot \mathbf{G}^{-1}[\mathbf{C}_2] &= \mathbf{A}(\mathbf{R}_1 \cdot \mathbf{G}^{-1}[\mathbf{C}_2]) + x_1\mathbf{G} \cdot \mathbf{G}^{-1}[\mathbf{A}\mathbf{R}_2 + x_2\mathbf{G}] \\ &= \mathbf{A}(\underbrace{\mathbf{R}_1 \cdot \mathbf{G}^{-1}[\mathbf{C}_2] + x_1\mathbf{R}_2}_{\mathbf{R}_\times}) + x_1x_2\mathbf{G} \ . \end{aligned}$$

In words, $\mathbf{G} - \mathbf{C}_1, \mathbf{C}_+, \mathbf{C}_\times$ are commitments to $1 - x_1, x_1 + x_2, x_1x_2$ under randomness $-\mathbf{R}_1, \mathbf{R}_+, \mathbf{R}_\times$, respectively. Moreover, if the original committed values x_i and randomness $\mathbf{R}_i$ are "small" in norm, then so are the new values and randomness (though they are somewhat larger), because $\mathbf{G}^{-1}[\mathbf{C}_2]$ is small.

In particular, if the original committed values $x_i \in \{0, 1\}$ are restricted to *bits*, then the above homomorphic operations yield a complete set of logical gates with which we can homomorphically evaluate any boolean circuit. For example, we can implement NAND$(x, y) = 1 - xy$ using the third equation, then the first one. Of course, the size of the randomness in the final committed result depends on the depth and size of the circuit. Similarly, as shown in [BV14, AP14], the asymmetric factors applied to the commitment randomness $\mathbf{R}_1$ versus $\mathbf{R}_2$ in $\mathbf{R}_\times$ can be exploited to implement other models of computation, like branching programs, with tighter control over the magnitude of the derived randomness. In particular, the magnitude can be limited to just polynomial in the length of the branching program.

For our purposes, we need one more simple homomorphic property. Suppose we have a commitment

$$\mathbf{C} = \mathbf{A}\mathbf{R} + \mathbf{x}^t \otimes \mathbf{G} = \mathbf{A}\mathbf{R} + \mathbf{x}^t \otimes \mathbf{I}_n \otimes \mathbf{g}^t$$

to a vector $\mathbf{x} \in \mathbb{Z}_q^L$. (Observe that the ith m-column chunk of $\mathbf{C}$ is $\mathbf{C}_i = \mathbf{A}\mathbf{R}_i + x_i\mathbf{G} \in \mathbb{Z}_q^{n \times m}$, where $\mathbf{R}_i$ is the analogous chunk of $\mathbf{R}$.) Any matrix $\mathbf{M} \in \mathbb{Z}_q^{n \times L}$ can be "vectorized" as an $\mathbf{m} \in \mathbb{Z}_q^{nL}$, so that $(\mathbf{x}^t \otimes \mathbf{I}_n) \cdot \mathbf{m} = \mathbf{M}\mathbf{x}$. Then

$$\begin{aligned} \mathbf{c}_\mathbf{M} := \mathbf{C} \cdot \mathbf{G}_{Ln}^{-1}[\mathbf{m}] &= \mathbf{A}(\underbrace{\mathbf{R} \cdot \mathbf{G}_{Ln}^{-1}[\mathbf{m}]}_{\mathbf{r}_\mathbf{M}}) + (\mathbf{x}^t \otimes \mathbf{I}_n \otimes \mathbf{g}^t) \cdot (\mathbf{I}_L \otimes \mathbf{I}_n \otimes \mathbf{g}^{-t})[\mathbf{m}] \\ &= \mathbf{A}\mathbf{r}_\mathbf{M} + (\mathbf{x}^t \otimes \mathbf{I}_n) \cdot \mathbf{m} \\ &= \mathbf{A}\mathbf{r}_\mathbf{M} + \mathbf{M}\mathbf{x} \in \mathbb{Z}_q^n \ . \end{aligned}$$

We view $\mathbf{c}_\mathbf{M}$ as a "quasi-commitment" to $\mathbf{M}\mathbf{x} \in \mathbb{Z}_q^n$, under randomness $\mathbf{r}_\mathbf{M}$, which is small if $\mathbf{R}$ is small. (We call it a quasi-commitment because it does not appear to support any nonlinear homomorphic operations.)

We summarize all of the above in the following fully homomorphic commitment scheme.

Construction 2.9. The commitment scheme FHC is parameterized by n and q, and is defined as follows. Each input in square brackets is optional, and when provided, the algorithm also produces the additional described output. The algorithm's main output is the same whether or not the optional input is provided.

- Gen chooses a uniformly random $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times w}$, where $w = 2m = 2n\ell$.
- $\mathsf{Com}(\mathbf{A} \in \mathbb{Z}_q^{n\times w}, \mathbf{x} \in \mathbb{Z}_q^S; \mathbf{R} \leftarrow \mathbb{Z}^{w\times Sm})$ outputs a commitment $\mathbf{C} = \mathbf{AR} + \mathbf{x}^t \otimes \mathbf{G} \in \mathbb{Z}_q^{n\times Sm}$. If the randomness $\mathbf{R}$ is not provided explicitly, it is chosen uniformly from $\{0,1\}^{w\times Sm}$ (but note that it is not required to be binary in general).
- $\mathsf{CircuitEval}(C, \mathbf{C} \in \mathbb{Z}_q^{n\times Sm}[, \mathbf{R} \in \mathbb{Z}^{w\times Sm}])$, for a boolean circuit $C\colon \{0,1\}^t \to \{0,1\}^L$, deterministically outputs a commitment matrix $\mathbf{C}_C \in \mathbb{Z}^{n\times Lm}$ [and additionally an integral matrix $\mathbf{R}_C \in \mathbb{Z}^{w\times Lm}$].
- $\mathsf{BranchEval}(B, \mathbf{C} \in \mathbb{Z}_q^{n\times Sm}[, \mathbf{R} \in \mathbb{Z}^{w\times Sm}])$, for a branching program $B\colon \{0,1\}^S \to \{0,1\}^L$, deterministically outputs a commitment matrix $\mathbf{C}_B \in \mathbb{Z}^{n\times Lm}$ [and additionally an integral matrix $\mathbf{R}_B \in \mathbb{Z}^{w\times Lm}$].
- $\mathsf{QuasiEval}(\mathbf{M} \in \mathbb{Z}_q^{n\times L}, \mathbf{C} \in \mathbb{Z}_q^{n\times Lm}[, \mathbf{R} \in \mathbb{Z}^{w\times Lm}])$ deterministically outputs a "quasi-commitment" vector $\mathbf{c}_\mathbf{M} \in \mathbb{Z}_q^n$ [and additionally an integral vector $\mathbf{r}_\mathbf{M} \in \mathbb{Z}^w$].

Proposition 2.10. *The above commitment scheme* FHC *satisfies the following properties:*

1. *By the leftover hash lemma, for any* $\mathbf{x} \in \mathbb{Z}_q^{\mathrm{poly}(m)}$ *the distribution of* $(\mathbf{A}, \mathbf{C})$ *has* $\mathrm{negl}(m)$ *statistical distance from uniformly random, where* $\mathbf{A} \leftarrow \mathsf{Gen}(1^n)$ *and* $\mathbf{C} \leftarrow \mathsf{Com}(\mathbf{A}, \mathbf{x})$.
2. *For any boolean circuit* $C\colon \{0,1\}^S \to \{0,1\}^L$ *of depth* d*, any* $\mathbf{x} \in \{0,1\}^S$*, any* $\mathbf{A} \in \mathbb{Z}_q^{n\times w}$ *and any* $\mathbf{R} \in \mathbb{Z}^{w\times Sm}$*, for commitment* $\mathbf{C} = \mathsf{Com}(\mathbf{A}, \mathbf{x}; \mathbf{R})$ *we have*
$$\mathsf{CircuitEval}(C, \mathbf{C}) = \mathsf{Com}(\mathbf{A}, C(x); \mathbf{R}_C) \ ,$$
where $\mathbf{R}_C \in \mathbb{Z}^{w\times Lm}$ *is the additional output of* $\mathsf{CircuitEval}(C, \mathbf{C}, \mathbf{R})$*, and* $|\mathbf{R}_C| = |\mathbf{R}| \cdot m^{O(d)}$.
3. *For any branching program* $B\colon \{0,1\}^S \to \{0,1\}^L$ *of length* D*, any* $\mathbf{x} \in \mathcal{X}$*, any* $\mathbf{A} \in \mathbb{Z}_q^{n\times w}$ *and any* $\mathbf{R} \in \mathbb{Z}^{w\times Sm}$*, for commitment* $\mathbf{C} = \mathsf{Com}(\mathbf{A}, \mathbf{x}; \mathbf{R})$ *we have*
$$\mathsf{BranchEval}(B, \mathbf{C}) = \mathsf{Com}(\mathbf{A}, B(x); \mathbf{R}_B) \ ,$$
where $\mathbf{R}_B \in \mathbb{Z}^{w\times Lm}$ *is the additional output of* $\mathsf{BranchEval}(B, \mathbf{C}, \mathbf{R})$*, and* $|\mathbf{R}_B| = |\mathbf{R}| \cdot m^{O(1)}D$.
4. *For any matrix* $\mathbf{M} \in \mathbb{Z}_q^{n\times L}$*, any* $\mathbf{x} \in \{0,1\}^L$*, any* $\mathbf{A} \in \mathbb{Z}_q^{n\times w}$ *and any* $\mathbf{R} \in \mathbb{Z}^{w\times Lm}$*, for commitment* $\mathbf{C} = \mathbf{AR} + \mathbf{x}^t \otimes \mathbf{G}$ *we have*
$$\mathsf{QuasiEval}(\mathbf{M}, \mathbf{C}) = \mathbf{A}\mathbf{r}_\mathbf{M} + \mathbf{M}\mathbf{x} \ ,$$
where $\mathbf{r}_\mathbf{M} \in \mathbb{Z}^w$ *is the additional output of* $\mathsf{QuasiEval}(\mathbf{M}, \mathbf{C}, \mathbf{R})$*, and* $|\mathbf{r}_\mathbf{M}| \le |\mathbf{R}| \cdot Lm$.

3 Correlation-Intractable Hashing from SIS/LWE

In this section we construct correlation-intractable hash families for (searchable relations defined by) arbitrary functions of bounded complexity, based on SIS. Particular cases of interest are functions computable by *log-depth* (i.e., NC^1) circuits, and polynomial-length *branching programs*, either of which are sufficient to invoke our bootstrapping transform in Section 4.

3.1 Construction for Circuits

Let FHC be the fully homomorphic commitment scheme from Section 2.7. Recall that FHC is parameterized by an SIS dimension n and a modulus q, which we instantiate below as functions of the security parameter λ based on the targeted class of functions. Our hash families work for functions of arbitrary input length, and output length *exactly* $m = n\ell = n\lceil \log q \rceil$. Correlation intractability immediately extends to functions of output length greater than m, simply by appending zeros to the length-m hash output.

We start with a construction that is correlation intractable for boolean *circuits*.

Construction 3.1 (CIH for circuits). The hash family $\mathsf{CIH} = (\mathsf{Gen}, \mathsf{Hash})$ with fake-key generation algorithm $\mathsf{StatGen}$ is parameterized by an arbitrary circuit size $S = S(\lambda) = \mathrm{poly}(\lambda)$ and depth $d = d(\lambda) \leq S(\lambda)$. Let $U(C, x) = C(x)$ denote a depth-universal circuit for size-S circuits.

- $\mathsf{Gen}(1^\lambda)$: generate $\mathbf{A} \leftarrow \mathsf{FHC.Gen}$ and $\mathbf{C} \leftarrow \mathsf{Com}(\mathbf{A}, 0^{S(\lambda)})$, choose a uniformly random $\mathbf{a} \leftarrow \mathbb{Z}_q^n$, and output the hash key $k = (\mathbf{a}, \mathbf{C})$.
- $\mathsf{StatGen}(1^\lambda, C)$: given a circuit C of size S, choose a uniformly random $\bar{\mathbf{A}} \leftarrow \mathbb{Z}_q^{(n-1)\times m}$ and $\bar{\mathbf{a}} \leftarrow \mathbb{Z}_q^{n-1}$. Choose $\mathbf{s} \leftarrow \mathbb{Z}_q^{n-1}$, $\mathbf{e} \leftarrow \chi^m$ and $e \leftarrow \chi$, where χ is an LWE error distribution. Let
$$\mathbf{A} := \begin{bmatrix} \bar{\mathbf{A}} \\ \mathbf{s}^t\bar{\mathbf{A}} + \mathbf{e}^t \end{bmatrix} \in \mathbb{Z}_q^{n\times m}, \quad \mathbf{a} := \begin{bmatrix} \bar{\mathbf{a}} \\ \mathbf{s}^t \cdot \bar{\mathbf{a}} + e - \lfloor q/2 \rfloor \end{bmatrix} \in \mathbb{Z}_q^n \ .$$
Compute $\mathbf{C} \leftarrow \mathsf{Com}(\mathbf{A}, C)$ and output the hash key $k = (\mathbf{a}, \mathbf{C})$.
- $\mathsf{Hash}(k = (\mathbf{a}, \mathbf{C}), x)$: let circuit $U_x(\cdot) = U(\cdot, x)$, and output
$$\mathbf{G}_n^{-1}[\mathbf{a} + \mathsf{QuasiEval}(\mathbf{G}_n, \mathsf{CircuitEval}(U_x, \mathbf{C}))] \in \{0,1\}^m.$$

Remark 3.2. By Item 1 of Proposition 2.10, the hash key $k = (\mathbf{a}, \mathbf{C})$ produced by Gen is statistically close to uniformly random, so CIH is public coin.

3.2 Correlation Intractability

We now prove that Construction 3.1 is *computationally* correlation intractable under an appropriate SIS assumption (Theorem 3.3), and *statistically* correlation intractable under an appropriate LWE assumption (Theorem 3.4).

Theorem 3.3. *Assuming the hardness of SIS$_{n,m+1,q,\beta}$ for a sufficiently large $\beta = m^{O(d)}$, Construction 3.1 is correlation intractable for the class of functions with output length m that can be implemented by size-S, depth-d boolean circuits.*

Proof. Let $\mathcal{A} = \{\mathcal{A}_\lambda\}$ be any non-uniform polynomial-size adversary, and fix any sequence of functions $\{f_\lambda\}$, where f_λ has output length $m = m(\lambda)$ and can be implemented by a circuit of size $S = S(\lambda)$ and depth $d = d(\lambda)$. To show that Construction 3.1 is correlation intractable with respect to f, we first define a hybrid experiment and show that it is statistically indistinguishable from the real experiment. Then we show that in this hybrid, it is hard for an adversary to break correlation intractability against $\{f_\lambda\}$.

In the hybrid experiment we merely modify how $\mathbf{C}$ is generated, letting it be $\mathbf{C} \leftarrow \mathsf{Com}(\mathbf{A}, f)$ for $f = f_\lambda$. By Item 1 of Proposition 2.10, this experiment is within statistical distance $\mathrm{negl}(m) = \mathrm{negl}(\lambda)$

from the real one, so $\mathcal{A}$'s success probability can differ by at most this much between the real and hybrid experiments.

We now show that under the hardness hypothesis, $\nu(\lambda) := \Pr_k[x = \mathcal{A}_\lambda(k) : \mathsf{Hash}(k,x) = f(x)]$ is a negligible function that depends only on $\mathcal{A}$ (not $\{f_\lambda\}$). To do this we use $\mathcal{A}$ to construct a non-uniform polynomial-size attacker $\mathcal{S} = \{\mathcal{S}_\lambda\}$ against SIS that also has success probability $\nu(\lambda)$, as follows.

The attacker $\mathcal{S}_\lambda$, given an SIS instance $\mathbf{A}' = [\mathbf{a} \mid \mathbf{A}] \in \mathbb{Z}_q^{n\times(m+1)}$, generates $\mathbf{C} \leftarrow \mathsf{Com}(\mathbf{A}, f)$ and retains the commitment randomness $\mathbf{R} \in \{0,1\}^{w\times Sm}$. It defines a hash key $k = (\mathbf{a}, \mathbf{C})$ and lets $x = \mathcal{A}_\lambda(k)$. If $\mathsf{Hash}(k,x) = f(x)$, then $\mathcal{S}$ lets $(\mathbf{C}_x, \mathbf{R}_x) = \mathsf{CircuitEval}(U_x, \mathbf{C}, \mathbf{R})$ and then lets $\mathbf{r}_x$ be the additional output of $\mathsf{QuasiEval}(\mathbf{G}_n, \mathbf{C}_x, \mathbf{R}_x)$. It outputs $\mathbf{z}_x = (1, \mathbf{r}_x) \in \mathbb{Z}^{m+1}$ as the nonzero SIS solution.

We now analyze $\mathcal{S}$. First observe that the distribution of the hash key k it provides to $\mathcal{A}_\lambda$ is exactly as in the hybrid experiment, by the uniform distribution of the SIS instance $\mathbf{A}' = [\mathbf{a} \mid \mathbf{A}]$. We claim that $\mathbf{z}_x = (1, \mathbf{r}_x)$ is a valid SIS solution whenever $\mathsf{Hash}(k,x) = f(x)$. To see this, observe that this condition implies that

$$\begin{aligned}
\mathbf{G}_n \cdot f(x) &= \mathbf{G}_n \cdot \mathsf{Hash}(k,x) \\
&= \mathbf{a} + \mathsf{QuasiEval}(\mathbf{G}_n, \mathsf{CircuitEval}(U_x, \mathbf{C})) \\
&= \mathbf{a} + (\mathbf{A}\mathbf{r}_x + \mathbf{G}_n \cdot f(x)) \\
&= \mathbf{A}'\mathbf{z}_x + \mathbf{G}_n \cdot f(x)
\end{aligned}$$

and that $\|\mathbf{z}_x\| = m^{O(d)} \leq \beta$, both by Item 2 and Item 4 of Proposition 2.10. Therefore, $\mathbf{A}'\mathbf{z}_x = 0$ and $\mathbf{z}_x$ satisfies the norm bound, as desired. □

Theorem 3.4. *Assuming the hardness of LWE$_{n-1,m+1,q,\chi}$ for a* $\mathrm{poly}(n)$*-bounded χ and a sufficiently large $q = m^{O(d)}$, Construction 3.1 is somewhere statistically correlation intractable for the class of functions with output length m that can be implemented by size-S, depth-d boolean circuits; each circuit serves as the intractability guarantee for itself.*

Proof. First, it follows immediately from the LWE assumption that the outputs of $\mathsf{Gen}(1^\lambda)$ and $\mathsf{Gen}(1^\lambda, C_\lambda)$ are computationally indistinguishable for any sequence of circuits C_λ of size S.

Now fix any sequence of functions $\{f_\lambda\}$, where f_λ has output length $m = m(\lambda)$ and can be implemented by a circuit of size $S = S(\lambda)$ and depth $d = d(\lambda)$. We will show that

$$\Pr_{k\leftarrow \mathsf{StatGen}(1^\lambda, f_\lambda)}[\exists x \text{ s.t. } \mathsf{Hash}(k,x) = f(x)] = 0 .$$

Using the notation from $\mathsf{StatGen}$, let $\mathbf{A}' = [\mathbf{a} \mid \mathbf{A}] \in \mathbb{Z}_q^{n\times(m+1)}$ and let $\bar{\mathbf{A}}' = [\bar{\mathbf{a}} \mid \bar{\mathbf{A}}] \in \mathbb{Z}_q^{(n-1)\times(m+1)}$ be its top $(n-1)$ rows. Similarly, let $\mathbf{e}' = [\mathbf{e} \mid e] \in \mathbb{Z}^{m+1}$. For any hash input x, define $\mathbf{r}_x$ and $\mathbf{z}_x = (1, \mathbf{r}_x) \in \mathbb{Z}^{m+1}$ exactly as in the proof of Theorem 3.3 above. Now, notice that if $\mathsf{Hash}(k,x) = f(x)$ then as above we have

$$\mathbf{G}_n \cdot f(x) = \mathbf{A}'\mathbf{z}_x + \mathbf{G}_n \cdot f(x).$$

This implies that

$$\begin{bmatrix} \bar{\mathbf{A}}' \cdot \mathbf{z}_x \\ \mathbf{s}^t \cdot \bar{\mathbf{A}}' \cdot \mathbf{z}_x + \mathbf{e}'^t \cdot \mathbf{z}_x \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \lfloor q/2 \rfloor \end{bmatrix}$$

and hence $\langle \mathbf{e}', \mathbf{z}_x \rangle = \lfloor q/2 \rfloor$. But this is impossible because $|\langle \mathbf{e}', \mathbf{z}_x \rangle| \leq \|\mathbf{e}'\| \cdot \|\mathbf{z}_x\| = n^{O(1)} \cdot m^{O(d)} = m^{O(d)}$, which is smaller than $q/2$ for a sufficiently large choice of $q = m^{O(d)}$. □

3.3 Construction for Branching Programs

We now describe a correlation-intractable hash family for *branching programs* of arbitrary length $D(\lambda) = \mathrm{poly}(\lambda)$. By Barrington's Theorem [Bar86] this is sufficient for evaluating *log-depth* (i.e., NC^1) circuits, and in particular the decryption functions of known FHE schemes. (It is also possible to express the decryption functions more efficiently, directly using branching programs [AP14].)

The construction is almost identical to Construction 3.1, except that it uses a universal branching program (in place of the universal circuit U) and $\mathsf{BranchEval}$ (in place of $\mathsf{CircuitEval}$). The proof of security is also essentially identical to those above, but due to Item 3 of Proposition 2.10, the derived randomness for the ultimate quasi-commitment grows only polynomially, as $m^{O(1)} \cdot D$. This yields the following two security theorems.

Theorem 3.5. *Assuming the hardness of $\mathit{SIS}_{n,m+1,q,\beta}$ for a sufficiently large $\beta = m^{O(1)} \cdot D$, the above-described construction is correlation intractable for the class of functions with output length m that can be implemented by length-D branching programs.*

Theorem 3.6. *Assuming the hardness of $\mathit{LWE}_{n-1,m,q,\chi}$ for a $\mathrm{poly}(n)$-bounded χ and a sufficiently large $q = m^{O(1)} \cdot D$, the above-described construction is somewhere statistically correlation intractable for the class of functions with output length m that can be implemented by length-D branching programs; each branching program serves as the intractability guarantee of itself.*

3.4 Parameter Instantiations

Here we show how the parameters n, q (with $\ell := \lceil \log q \rceil$ and $m := n\ell$) can be chosen, with a focus on the SIS problem and the branching program instantiation; a very similar process can be followed for LWE and/or circuits. For a branching program of length $D = \lambda^d$ and desired output size of (at most) $L = \lambda^c$ for some constants $c, d > 0$, let $\beta = m^{c_1} \cdot D$ for the (small) constant $c_1 > 0$ be the norm bound given by Theorem 3.5. To invoke worst-case hardness theorems, we can take some $q = \beta \cdot \tilde{O}(\sqrt{n})$ and $n = \lfloor L/\ell \rfloor$, so that the true output size $m = n\ell \le L$.

With these choices, we have $q = \mathrm{poly}(\lambda)$, $n = L/\Theta(\log \lambda) = \lambda^{c-o(1)}$, and $D = n^{d/c+o(1)}$. This corresponds to a worst-case approximation factor

$$\gamma(n) = \beta \cdot \tilde{O}(\sqrt{n}) = n^{c_1+d/c+1/2+o(1)} = \mathrm{poly}(n)$$

for the underlying n-dimensional lattice problem.

Two noteworthy extremes are as follows. We can obtain a very short hash output length of λ^c for arbitrarily small $c > 0$, where security is supported by (large) $\mathrm{poly}(n)$-approximate lattice problems in $n = \lambda^{c-o(1)}$ dimensions, which are plausibly subexponentially hard. On the other extreme, in our NIZK application using the bootstrapping transform, the value of d is fixed by the FHE scheme and we may choose $L = \lambda^c$ freely. So, by taking a large enough constant c, security is supported by (small) $n^{c_1+1/2+\epsilon}$ approximation factors for any desired constant $\epsilon > 0$.

4 Bootstrapping Correlation Intractability

In this section we present our bootstrapping theorem for correlation-intractable hash functions.

Construction 4.1. Let $\mathsf{FHE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be a (symmetric-key) homomorphic encryption scheme supporting a circuit class $\mathcal{C} = \{\mathcal{C}_\lambda\}$, and let $U_\lambda(C, x) = C(x)$ denote a universal circuit for circuits $C \in \mathcal{C}_\lambda$. Let $\mathsf{CIH} = (\mathsf{Gen}, \mathsf{Hash})$ be a hash function family with fake-key generation algorithm $\mathsf{StatGen}$. Define a new hash family $\mathsf{CIH}' = (\mathsf{Gen}', \mathsf{Hash}')$ with fake-key generation algorithm $\mathsf{StatGen}'$ as follows:

- $\mathsf{Gen}'(1^\lambda)$: generate $k \leftarrow \mathsf{CIH.Gen}(1^\lambda)$ and $(sk, ek) \leftarrow \mathsf{FHE.Gen}(1^\lambda)$. Generate $c \leftarrow \mathsf{Enc}(pk, D)$ for some arbitrary "dummy" circuit $D \in \mathcal{C}_\lambda$, and output hash key $k' = (k, ek, c)$.
- $\mathsf{StatGen}'(1^\lambda, C \in \mathcal{C}_\lambda)$: generate $(sk, ek) \leftarrow \mathsf{FHE.Gen}(1^\lambda)$ and $k \leftarrow \mathsf{StatGen}(1^\lambda, \mathsf{FHE.Dec}(sk, \cdot))$. Generate $c \leftarrow \mathsf{Enc}(pk, C)$ and output hash key $k' = (k, ek, c)$.
- $\mathsf{Hash}'(k' = (k, ek, c), x)$: let circuit $U_x(\cdot) = U_\lambda(\cdot, x)$, and output $\mathsf{Hash}(k, \mathsf{Eval}(ek, U_x, c))$.

Remark 4.2. Observe that if the original CIH family has (pseudo)random hash keys, and FHE has jointly pseudorandom evaluation keys and ciphertexts, then CIH' has pseudorandom hash keys as well.

Let $\mathcal{R} = \{\mathcal{R}_\lambda = \{R_\lambda\}\}$ be a class of relations. For each $R_\lambda \in \mathcal{R}_\lambda$, each secret key sk that may be output by $\mathsf{FHE.Gen}(1^\lambda)$, and each circuit $C \in \mathcal{C}_\lambda$, define the associated relations

$$\begin{aligned} R_{\lambda,sk} &= \{(c, y) : (\mathsf{FHE.Dec}(sk, c), y) \in R_\lambda\} \\ R_{\lambda,C} &= \{(x, y) : (C(x), y) \in R_\lambda\} \ . \end{aligned}$$

Essentially, these relations first apply some computation (either decryption with a certain fixed secret key, or some circuit C) to the input, then check whether the provided witness is valid (under the original relation) for the result. They naturally yield the associated relation classes $\mathcal{R}^{\mathsf{Dec}} := \{\mathcal{R}_\lambda^{\mathsf{Dec}} = \{R_{\lambda,sk} : R_\lambda \in \mathcal{R}_\lambda\}\}$ and $\mathcal{R}^{\mathcal{C}} := \{\mathcal{R}_\lambda^{\mathcal{C}} = \{R_{\lambda,C} : R_\lambda \in \mathcal{R}_\lambda, C \in \mathcal{C}_\lambda\}\}$.

Theorem 4.3. *If* FHE *is CPA-secure (for the sequence of message spaces* $\{\mathcal{C}_\lambda\}$*) and* CIH *is correlation intractable for the relation class* $\mathcal{R}^{\mathsf{Dec}}$*, then* CIH' *is correlation intractable for the relation class* $\mathcal{R}^{\mathcal{C}}$*.*

Proof. Let $\mathcal{A}' = \{\mathcal{A}'_\lambda\}$ be a non-uniform polynomial-size adversary against the correlation intractability of CIH' for $\mathcal{R}^{\mathcal{C}}$, and fix any sequence of relations $\{R_{\lambda,C_\lambda}\}$ for some choice of $C_\lambda \in \mathcal{C}_\lambda$ for each λ.

We first define a hybrid experiment and show that it is computationally indistinguishable from the real experiment. In the hybrid experiment we modify only how the c component of the hash key is generated, letting $c \leftarrow \mathsf{Enc}(pk, C_\lambda)$. By the CPA-security of FHE, the success probability of $\mathcal{A}'$ can differ by only a negligible amount between the real and the hybrid experiments. (The reduction showing this is straightforward, because sk is not used in the experiment.)

Our goal is prove that in the hybrid experiment,

$$\nu(\lambda) := \Pr_{k'}[x \leftarrow \mathcal{A}'(k') : (x, \mathsf{Hash}'(k', x)) \in R_{\lambda,C_\lambda}]$$

is a negligible function that depends only on $\mathcal{A}'$ (and not R_{λ,C_λ}). First, observe that by construction of CIH',

$$\Pr\left[\begin{array}{c} k \leftarrow \mathsf{CIH.Gen}(1^\lambda) \\ (sk, ek) \leftarrow \mathsf{FHE.Gen}(1^\lambda) \\ c \leftarrow \mathsf{Enc}(ek, C_\lambda) \\ x = \mathcal{A}'_\lambda(k' = (k, ek, c)) \\ c_x = \mathsf{Eval}(ek, U_x, c) \end{array} \middle| \ (C_\lambda(x), \mathsf{Hash}(k, c_x)) \in R_{\lambda,C_\lambda} \right] = \nu(\lambda) \ .$$

By an averaging argument, there exists (sk_λ, ek_λ) in the support of $\mathsf{FHE.Gen}(1^\lambda)$ such that

$$\Pr\left[\begin{array}{c} k \leftarrow \mathsf{CIH.Gen}(1^\lambda) \\ c \leftarrow \mathsf{Enc}(ek_\lambda, C_\lambda) \\ x = \mathcal{A}'_\lambda(k' = (k, ek_\lambda, c)) \\ c_x = \mathsf{Eval}(ek_\lambda, U_x, c) \end{array} \middle| (C_\lambda(x), \mathsf{Hash}(k, c_x)) \in R_{\lambda, C_\lambda} \right] \geq \nu(\lambda) \ . \tag{4.1}$$

We use $\mathcal{A}'$ to construct a non-uniform polynomial-size attacker $\mathcal{A} = \{\mathcal{A}_\lambda\}$ against the correlation intractability of CIH for $\mathcal{R}^{\mathsf{Dec}}$, and specifically the sequence of relations $\{R_{\lambda, sk_\lambda}\}$. Given a CIH key k, $\mathcal{A}_\lambda$ generates $c \leftarrow \mathsf{Enc}(ek_\lambda, C_\lambda)$, lets $x = \mathcal{A}'_\lambda(k' = (k, ek_\lambda, c))$, and outputs $c_x = \mathsf{Eval}(ek_\lambda, U_x, c)$.

We now prove that $\mathcal{A}_\lambda$ succeeds with probability at least $\nu(\lambda)$, hence $\nu(\lambda)$ is a negligible function (that does not depend on the choice of relations). First, notice that the distribution of k' that $\mathcal{A}_\lambda$ passes to $\mathcal{A}'_\lambda$ is exactly as in Equation (4.1). Next, observe that by the correctness of FHE, we have $\mathsf{Dec}(sk_\lambda, c_x) = C_\lambda(x)$. Therefore, whenever $(C_\lambda(x), \mathsf{Hash}(k, c_x)) \in R_{\lambda, C_\lambda}$ we have $(x, \mathsf{Hash}(k, c_x)) \in R_\lambda$ and hence $(c_x, \mathsf{Hash}(k, c_x)) \in R_{\lambda, sk_\lambda}$, as needed. □

Theorem 4.4. *If* FHE *is CPA-secure (for the sequence of message spaces* $\{\mathcal{C}_\lambda\}$*) and* CIH *is somewhere statistical correlation intractable for the relation class* $\mathcal{R}^{\mathsf{Dec}}$*, where for each* $R_{\lambda, sk}$ *the intractability guarantee is the description of the circuit* $\mathsf{FHE.Dec}(sk, \cdot)$*, then* CIH' *is somewhere statistical correlation intractable for the relation class* $\mathcal{R}^{\mathcal{C}}$*, and for each* $R_{\lambda, C}$ *the intractability guarantee is the circuit* C*.*

Proof. First we have to argue that the outputs of $\mathsf{Gen}'(1^\lambda)$ and $\mathsf{StatGen}'(1^\lambda, C_\lambda)$ are computationally indistinguishable for any $C_\lambda \in \mathcal{C}_\lambda$. This follows immediately from the CPA-security of FHE and the fact that CIH is somewhere statistically correlation intractable with fake-key generation $\mathsf{StatGen}$.

Now fix any sequence of relations $\{R_{\lambda, C_\lambda}\}$ for some choice of $C_\lambda \in \mathcal{C}_\lambda$ for each λ. We need to show that

$$\nu(\lambda) := \Pr_{k' \leftarrow \mathsf{StatGen}'(1^\lambda, C_\lambda)} [\exists x \text{ s.t. } (x, \mathsf{Hash}'(k', x)) \in R_{\lambda, C_\lambda}]$$

is a negligible function (that does not depend on R_{λ, C_λ}). First, observe that by construction of CIH',

$$\Pr\left[\begin{array}{c} (sk, ek) \leftarrow \mathsf{FHE.Gen}(1^\lambda) \\ k \leftarrow \mathsf{StatGen}(1^\lambda, \mathsf{FHE.Dec}(sk, \cdot)) \\ c \leftarrow \mathsf{Enc}(ek, C_\lambda) \end{array} \middle| \begin{array}{c} \exists x \text{ s.t.} \\ (C_\lambda(x), \mathsf{Hash}(k, c_x)) \in R_{\lambda, C_\lambda} \text{ where} \\ c_x = \mathsf{Eval}(ek, U_x, c) \end{array} \right] = \nu(\lambda) \ .$$

By an averaging argument, there exists (sk_λ, ek_λ) in the support of $\mathsf{FHE.Gen}(1^\lambda)$ such that

$$\Pr\left[\begin{array}{c} k \leftarrow \mathsf{StatGen}(1^\lambda, \mathsf{FHE.Dec}(sk_\lambda, \cdot)) \\ c \leftarrow \mathsf{Enc}(ek_\lambda, C_\lambda) \end{array} \middle| \begin{array}{c} \exists x \text{ s.t.} \\ (C_\lambda(x), \mathsf{Hash}(k, c_x)) \in R_{\lambda, C_\lambda} \text{ where} \\ c_x = \mathsf{Eval}(ek_\lambda, U_x, c) \end{array} \right] \geq \nu(\lambda) \ . \tag{4.2}$$

Next, observe that by the correctness of FHE, we have $\mathsf{Dec}(sk_\lambda, c_x) = C_\lambda(x)$. Therefore, whenever $(C_\lambda(x), \mathsf{Hash}(k, c_x)) \in R_{\lambda, C_\lambda}$ we have $(x, \mathsf{Hash}(k, c_x)) \in R_\lambda$ and hence $(c_x, \mathsf{Hash}(k, c_x)) \in R_{\lambda, sk_\lambda}$. So, Equation (4.2) implies that

$$\Pr_{k \leftarrow \mathsf{StatGen}(1^\lambda, \mathsf{Dec}(sk_\lambda, \cdot))} [\exists c_x \text{ s.t. } (c_x, \mathsf{Hash}(k, c_x)) \in R_{\lambda, C_\lambda}] \geq \nu(\lambda).$$

The theorem follows by the somewhere statistical correlation intractability of CIH. □

5 Putting it All Together

In this section we assemble the components from the previous sections and prior works to obtain correlation-intractable hash families for all bounded circuits, and our main result of noninteractive zero knowledge for all of NP. (Throughout this section, for simplicity we assume the standard LWE error distribution χ, i.e., a discrete Gaussian of parameter $r = 2\sqrt{n}$ for LWE dimension n.)

5.1 Correlation-Intractable Hashing for all Circuits

In this subsection let $L = L(\lambda)$, $S = S(\lambda)$, and $d = d(\lambda)$ be arbitrary $\mathrm{poly}(\lambda)$-bounded functions, and define the relation class $\mathcal{R}_{L,S,d} = \{\mathcal{R}_{\lambda,L,S,d}\}$, where $\mathcal{R}_{\lambda,L,S,d} = \{R_f = \{(x, f(x))\}\}$ is the set of all efficiently searchable relations whose search functions f can be computed by a circuit with output length $L(\lambda)$, size $S(\lambda)$, and depth $d(\lambda)$.

Let FHE be a leveled fully homomorphic encryption scheme instantiated to support circuits of depth at most $d = d(\lambda)$, with decryption circuit having size $S_{\mathsf{Dec}}(\lambda)$ and logarithmic depth $d_{\mathsf{Dec}}(\lambda) = O(\log \lambda)$. Let CIH denote Construction 3.1 for circuit size $S = L \cdot S_{\mathsf{Dec}}(\lambda)$ (allowing for the decryption of L ciphertexts) and depth $d = d_{\mathsf{Dec}}(\lambda)$, and with FHC parameters n, q satisfying $L \geq n\lceil \lg q \rceil$.

Theorem 5.1. *Assuming the hardness of SIS$_{n,q,\beta}$ for a suitable $\beta = \mathrm{poly}(S)$ (respectively LWE$_{n-1,q,\chi}$ for a $\mathrm{poly}(n)$-bounded χ and a suitable $q = \mathrm{poly}(S)$) and the CPA-security of* FHE*, Construction 4.1 instantiated with* FHE *and* CIH *is correlation intractable with respect to $\mathcal{R}_{L,S,d}$ (respectively, somewhere statistically correlation intractable with respect to $\mathcal{R}_{L,S,d}$, where for each $R_f \in \mathcal{R}_{L,S,d}$ the intractability guarantee is f.)*

Proof. Let $\mathcal{I} = \{I_\lambda = \{(x, x) : x \in \{0,1\}^{L(\lambda)}\}\}$ be the class of equality relations. Because $\mathsf{FHE.Dec}$ has circuit depth $d_{\mathsf{Dec}} = O(\log \lambda)$, by Theorem 3.5 CIH is correlation intractable (respectively, somewhere statistically correlation intractable) for the relation class $\mathcal{I}^{\mathsf{Dec}}$ (as defined in Section 4). The theorem follows by noticing that $\mathcal{R}_{L,S,d} = \mathcal{I}^{\mathcal{C}}$ where $\mathcal{C}$ is the class of circuits used to define $\mathcal{R}_{L,S,d}$, and applying Theorem 4.3. □

Using any known leveled FHE scheme based on LWE with polynomial factors that has jointly pseudorandom evaluation keys and ciphertexts (e.g., [BV14]), we get the following corollary.

Corollary 5.2. *Assuming the hardness of LWE with suitable polynomial factors, there exists a somewhere statistically correlation-intractable hash family (with pseudorandom hash keys) for $\mathcal{R}_{L,S,d}$, where for each $R_f \in \mathcal{R}_{L,S,d}$ the intractability guarantee is f.*

5.2 Noninteractive Zero Knowledge for NP

We are now ready to instantiate the noninteractive zero-knowledge protocol from [CCH$^+$19] with our correlation-intractable hash functions. We first recall the following theorem; see Definition 2.2 for a reminder of the NIZK modifiers.

Theorem 5.3 ([CCH$^+$19]). *Assuming the existence of*

- *a lossy public-key encryption scheme with uniformly random lossy public keys (respectively, an ordinary CPA-secure public-key encryption scheme), and*

- *a hash family with (pseudo)random keys which is CI for all circuits of output length $L(\lambda) \geq \lambda^c$ for some constant $c > 0$ and size bounded by some sufficiently large $S(\lambda) = \mathrm{poly}(\lambda)$ (respectively, a hash family that is somewhere statistically correlation intractable for all such circuits, where the intractability guarantee for each circuit is itself),*

there exists an adaptively sound, statistically zero-knowledge noninteractive argument system with common random string for any NP language (respectively, a statistically sound, adaptively zero-knowledge noninteractive proof system with common reference string).

A lossy encryption scheme satisfying the requirements of Theorem 5.3 can be constructed based on LWE with polynomial factors (see, e.g., [Reg05, PVW08]). So, by Corollary 5.2 we get our main result:

Theorem 5.4. *Assuming the hardness of LWE with suitable polynomial factors, for any NP language there exists*

- *an adaptively sound, statistically zero-knowledge noninteractive argument system having a common random string, and*
- *a statistically sound, adaptively zero-knowledge noninteractive proof system having a common reference string.*

Remark 5.5. We remark that intractability bootstrapping and leveled FHE are not actually necessary for the NIZK construction, because we just need a hash family that is correlation intractable for the class of "bad challenge" functions of the underlying graph-Hamiltonicity protocol of [FLS90]. As pointed out to us by Alex Lombardi, a trick from [CCH+19] allows the bad-challenge functions to be implemented in NC^1 (i.e., logarithmic depth), so we can obtain the required correlation intractability merely from SIS with small polynomial factors. (However, we still need LWE for the lossy encryption ingredient.)

In short, the bad-challenge function decrypts the prover's ciphertexts to recover a graph, then checks whether the graph is a cycle. Decryption of LWE-based lossy encryption in NC^1 is standard. To implement the cycle check, we additionally require the prover to (de)commit to a permutation between its committed graph and a canonical cycle graph. The bad-challenge function (and verifier) performs the appropriate checks, which can be done in logarithmic depth by brute force. (Without the explicit permutation, the best known parallel complexity for cycle checking is NC^2, which is not good enough for the present purpose.)

References

[Ajt96] M. Ajtai. Generating hard instances of lattice problems. *Quaderni di Matematica*, 13:1–32, 2004. Preliminary version in STOC 1996.

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

[Bar86] D. A. M. Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. *J. Comput. Syst. Sci.*, 38(1):150–164, 1989. Preliminary version in STOC 1986.

[BCG+14] E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza. Zerocash: Decentralized anonymous payments from Bitcoin. In *2014 IEEE Symposium on Security and Privacy, SP 2014, Berkeley, CA, USA, May 18-21, 2014*, pages 459–474. 2014.

[BDMP88] M. Blum, A. De Santis, S. Micali, and G. Persiano. Noninteractive zero-knowledge. *SIAM J. Comput.*, 20(6):1084–1118, 1991. Preliminary version in STOC 1988.

[BFM88] M. Blum, P. Feldman, and S. Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *STOC*, pages 103–112. 1988.

[BG89] M. Bellare and S. Goldwasser. New paradigms for digital signatures and message authentication based on non-interative zero knowledge proofs. In *CRYPTO*, pages 194–211. 1989.

[BGG$^+$14] D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, V. Nikolaenko, G. Segev, V. Vaikuntanathan, and D. Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In *EUROCRYPT*, pages 533–556. 2014.

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

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

[CCH$^+$19] R. Canetti, Y. Chen, J. Holmgren, A. Lombardi, G. N. Rothblum, R. D. Rothblum, and D. Wichs. Fiat-Shamir: From practice to theory. In *STOC*. 2019. To appear.

[CCRR18] R. Canetti, Y. Chen, L. Reyzin, and R. D. Rothblum. Fiat-Shamir and correlation intractability from strong KDM-secure encryption. In *EUROCRYPT*, pages 91–122. 2018.

[CDLP13] K. Chung, D. Dadush, F. Liu, and C. Peikert. On the lattice smoothing parameter problem. In *IEEE Conference on Computational Complexity*, pages 230–241. 2013.

[CGH98] R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, 2004. Preliminary version in STOC 1998.

[DN00] C. Dwork and M. Naor. Zaps and their applications. *SIAM J. Comput.*, 36(6):1513–1543, 2007.

[FLS90] U. Feige, D. Lapidot, and A. Shamir. Multiple noninteractive zero knowledge proofs under general assumptions. *SIAM J. Comput.*, 29(1):1–28, 1999. Preliminary version in FOCS 1990.

[FS86] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, pages 186–194. 1986.

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

[GGH$^+$13] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *FOCS*, pages 40–49. 2013.

[GMR85] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989. Preliminary version in STOC 1985.

[GO94] O. Goldreich and Y. Oren. Definitions and properties of zero-knowledge proof systems. *J. Cryptology*, 7(1):1–32, 1994.

[GOS06] J. Groth, R. Ostrovsky, and A. Sahai. Perfect non-interactive zero knowledge for NP. In *EUROCRYPT*, pages 339–358. 2006.

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

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

[HL18] J. Holmgren and A. Lombardi. Cryptographic hashing from strong one-way functions (or: One-way product functions and their applications). In *FOCS*, pages 850–858. 2018.

[KRR17] Y. T. Kalai, G. N. Rothblum, and R. D. Rothblum. From obfuscation to the security of Fiat-Shamir for proofs. In *CRYPTO*, pages 224–251. 2017.

[KW18] S. Kim and D. J. Wu. Multi-theorem preprocessing NIZKs from lattices. In *CRYPTO*, pages 733–765. 2018.

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

[MR04] D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measures. *SIAM J. Comput.*, 37(1):267–302, 2007. Preliminary version in FOCS 2004.

[MV03] D. Micciancio and S. P. Vadhan. Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In *CRYPTO*, pages 282–298. 2003.

[NY90] M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC*, pages 427–437. 1990.

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

[PRS17] C. Peikert, O. Regev, and N. Stephens-Davidowitz. Pseudorandomness of Ring-LWE for any ring and modulus. In *STOC*, pages 461–473. 2017.

[PV08] C. Peikert and V. Vaikuntanathan. Noninteractive statistical zero-knowledge proofs for lattice problems. In *CRYPTO*, pages 536–553. 2008.

[PVW08] C. Peikert, V. Vaikuntanathan, and B. Waters. A framework for efficient and composable oblivious transfer. In *CRYPTO*, pages 554–571. 2008.

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

[SW14] A. Sahai and B. Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In *STOC*, pages 475–484. 2014.