Towards Non-Interactive Zero-Knowledge for NP from LWE

Ron D. Rothblum*, Adam Sealfon**, and Katerina Sotiraki***

Abstract. Non-interactive zero-knowledge (NIZK) is a fundamental primitive that is widely used in the construction of cryptographic schemes and protocols. Despite this, general purpose constructions of NIZK proof systems are only known under a rather limited set of assumptions that are either number-theoretic (and can be broken by a quantum computer) or are not sufficiently well understood, such as obfuscation. Thus, a basic question that has drawn much attention is whether it is possible to construct general-purpose NIZK proof systems based on the *learning with errors* (LWE) assumption.

Our main result is a reduction from constructing NIZK proof systems for all of **NP** based on LWE, to constructing a NIZK proof system for a particular computational problem on lattices, namely a decisional variant of the Bounded Distance Decoding (BDD) problem. That is, we show that assuming LWE, *every* language $L \in \mathbf{NP}$ has a NIZK proof system if (and only if) the decisional BDD problem has a NIZK proof system. This (almost) confirms a conjecture of Peikert and Vaikuntanathan (CRYPTO, 2008).

To construct our NIZK proof system, we introduce a new notion that we call *prover-assisted oblivious ciphertext sampling* (POCS), which we believe to be of independent interest. This notion extends the idea of *oblivious ciphertext sampling*, which allows one to sample ciphertexts without knowing the underlying plaintext. Specifically, we augment the oblivious ciphertext sampler with access to an (untrusted) prover to help it accomplish this task. We show that the existence of encryption schemes with a POCS procedure, as well as some additional natural requirements, suffices for obtaining NIZK proofs for **NP**. We further show that such encryption schemes can be instantiated based on LWE, assuming the existence of a NIZK proof system for the decisional BDD problem.

* Technion. Email: rothblum@cs.technion.ac.il. This research was conducted in part while the author was at MIT and Northeastern University. Research supported in part by the Israeli Science Foundation (Grant No. 1262/18). Research also supported in part by NSF Grants CNS-1413920 and CNS-1350619, by the Defense Advanced Research Projects Agency (DARPA) and the U.S. Army Research Office under contracts W911NF-15-C-0226 and W911NF-15-C-0236, the Simons Investigator award agreement dated 6-5-12 and the Cybersecurity and Privacy Institute at Northeastern University.

** MIT. Email: asealfon@mit.edu. Research supported in part by a DOE CSGF fellowship, NSF MACS CNS-1413920, DARPA/NJIT Palisade 491512803, Sloan/NJIT 996698, MIT/IBM W1771646, NSF Center for Science of Information (CSoI) CCF-0939370, and the Simons Investigator award agreement dated 6-5-12.

*** MIT. Email: katesot@mit.edu. Research supported in part by NSF grants CNS-1350619, CNS-1718161, CNS-1414119.

1 Introduction

The *learning with errors* (LWE) problem, introduced by Regev [Reg09], has had a profound impact on cryptography. The goal in LWE is to find a solution to a set of *noisy* linear equations modulo a large integer q, where the noise is typically drawn from a discrete Gaussian distribution. The assumption that LWE cannot be broken in polynomial time can be based on *worst-case* hardness of lattice problems [Reg09,Pei09] and has drawn immense interest in recent years.

Immediately following its introduction, LWE was shown to imply the existence of many important cryptographic primitives such as public-key encryption [Reg09], circular secure encryption [ACPS09], oblivious transfer [PVW08], chosen ciphertext security [PW08,Pei09], etc. Even more remarkably, in recent years LWE has been used to achieve schemes and protocols above and beyond what was previously known from other assumptions. Notable examples include fully homomorphic encryption [BV14], predicate encryption and certain types of functional encryption (see, e.g., [AFV11,GKP+13,GVW15]), and even obfuscation of certain expressive classes of computations [WZ17,GKW17].

Despite this amazing list of applications, one major primitive that has resisted all LWE based attempts is general purpose *Non-Interactive Zero-Knowledge* (NIZK) *proof systems* for **NP**.[1] A NIZK proof system for a language $L \in \mathbf{NP}$, as introduced by Blum *et al.* [BFM88], is a protocol between a probabilistic polynomial-time prover P and verifier V in the *Common Random String* (CRS) model. The prover, given an instance $x \in L$, a witness w, and the random string r, produces a proof string π which it sends to the verifier. Based only on x, the random string r and the proof π, the verifier can decide whether $x \in L$. Furthermore, the protocol is zero-knowledge: the proof π reveals nothing to the verifier beyond the fact that $x \in L$.

Non-interactive zero-knowledge proofs have been used extensively in cryptography, with applications ranging from chosen ciphertext security and non-malleability [NY90,DDN03,Sah99], multi-party computation with a small number of rounds (see, e.g., [MW16]), low-round witness-indistinguishability [DN07] to various types of signatures (e.g. [BMW03,BKM06]) and beyond.

Currently, general purpose NIZK proof systems (i.e., NIZK proof systems for all of **NP**) are only known based on number theoretic assumptions (e.g., the hardness of factoring integers [FLS99] or the decisional linear assumption or symmetric external Diffie-Hellman assumption over bilinear groups [GOS12]) or from indistinguishability obfuscation [SW14,BP15] (see Section 1.2 for further discussion). We remark that the former class of assumptions can be broken by a quantum computer [Sho99] whereas the assumption that indistinguishability obfuscation exists is not yet well understood. Thus, the following basic question remains open:

Can we construct NIZK *proofs for all of* **NP** *based on* LWE*?*

[1] As a matter of fact, resolving this question carries a symbolic cash prize; see https://simons.berkeley.edu/crypto2015/open-problems.

1.1 Our Results

Our main result is a completeness theorem reducing the foregoing question to that of constructing a NIZK proof system for one particular computational problem. Specifically, we will consider a decisional variant of the *bounded distance decoding* (BDD) problem.

Recall that in the BDD problem, the input is a lattice basis and a target vector which is very close to the lattice. The problem is to find the nearby lattice point. This is very similar to the *closest vector problem* CVP except that here the vector is guaranteed to be within the λ_1 radius of the lattice, where λ_1 denotes the length of the shortest non-zero lattice vector (more specifically, the problem is parameterized by $\alpha \geq 1$ and the guarantee is that the point is at distance λ_1/α from the lattice). BDD can also be viewed as a worst-case variant of LWE and is known to be (up to polynomial factors) equivalent to the shortest-vector problem (more precisely, GapSVP) [LM09].

In this work, we consider a decisional variant of BDD, which we denote by dBDD. The $\mathsf{dBDD}_{\alpha,\gamma}$ problem, is a promise problem, parameterized by $\alpha \geq 1$ and $\gamma \geq 1$, where the input is a basis $\mathbf{B}$ of a lattice L and a point $\mathbf{t}$. The goal is to distinguish between pairs $(L, \mathbf{t})$ such that the point $\mathbf{t}$ has distance at most $\frac{\lambda_1(L)}{\alpha}$ from the lattice L from tuples in which $\mathbf{t}$ has distance at least $\gamma \cdot \frac{\lambda_1(L)}{\alpha}$ from L.

Our main result can be stated as follows:

Theorem 1 (Informal; see Theorem 2).
Suppose that LWE *holds and that* $\mathsf{dBDD}_{\alpha,\gamma}$ *has a* NIZK *proof system (where* α *and* γ *depend on the* LWE *parameters). Then, every language in* $\mathbf{NP}$ *has a* NIZK *proof system.*

Since dBDD is a special case of the well studied GapCVP problem, a NIZK for GapCVP would likewise suffice for obtaining NIZKs for all of $\mathbf{NP}$ based on LWE.

Relation to [PV08]. Theorem 1 (almost) confirms a conjecture of Peikert and Vaikuntanathan [PV08]. More specifically, [PV08] conjectured that a NIZK proof-system for a specific computational problem related to lattices would imply a NIZK proof-system for *every* $\mathbf{NP}$ language. The problem that Peikert and Vaikuntanathan consider is GapSVP whereas the problem that we consider is the closely related dBDD problem. While BDD is known to be no harder than GapSVP [LM09] (and the same can be shown for dBDD, see Proposition 1), these results are shown by *Cook* reductions and so a NIZK for one problem does not necessarily yield a NIZK for the other. In particular, we do not know how to extend Theorem 1 to hold with respect to GapSVP.

Tradeoff Between the Modulus and Gap. The tradeoff between α and γ and the LWE parameters is quantified precisely in the technical sections (see Theorem 2). Roughly speaking, we need both α and γ to be small relative to $1/\beta$, where β is the magnitude of the LWE error divided by the LWE modulus q. This tradeoff allows us to obtain NIZK proof systems for $\mathbf{NP}$ from a variety of parameter regimes. In particular, given a NIZK proof system for $\mathsf{dBDD}_{\alpha,\gamma}$ where α and γ are polynomial in the security parameter, we can instantiate Theorem 1 even assuming LWE with a polynomial-size modulus. On the other hand, it suffices to have a NIZK for $\mathsf{dBDD}_{\alpha,\gamma}$ with respect to a super-polynomial or even subexponential α or γ, assuming LWE with a super-polynomial or subexponential modulus.

Furthermore, we emphasize that it suffices for us that $\mathsf{dBDD}_{\alpha,\gamma}$ has a non-interactive *computational* zero-knowledge proof-system under the LWE assumption. However, it is entirely plausible that $\mathsf{dBDD}_{\alpha,\gamma}$ has an (unconditional) non-interactive *statistical* zero-knowledge proof system (NISZK).

1.2 Related Works

Non-Interactive Zero-Knowledge. Non-interactive zero-knowledge proofs were first introduced by Blum, Feldman and Micali [BFM88], who also constructed a NIZK proof system for all of **NP** based on the Quadratic Residuocity assumption. Later work by Feige, Lapidot and Shamir [FLS99] gave a construction under (an idealized version of) trapdoor permutations. Together with additional contributions of Bellare and Yung [BY96] and Goldreich [Gol11], this yields NIZK proofs for **NP** based on factoring (using a variant of Rabin's [Rab79] trapdoor permutation collection).

Groth, Ostrovsky and Sahai [GOS12] construct a more efficient general purpose NIZK proof-system based on hardness assumptions on groups equipped with bilinear maps. Groth and Sahai [GS08] also construct a NIZK proof system for *specific problems* related to such bilinear groups. Groth [Gro10] constructs highly efficient NIZK proofs assuming certain "knowledge of exponent" assumptions (which in particular are not falsifiable, in the sense of [Nao03]). More recently, constructions of NIZK arguments and proofs based on indistinguishability obfuscation were given by Sahai and Waters [SW14] and Bitansky and Paneth [BP15].

Another method for constructing non-interactive zero-knowledge proofs is via the *Fiat-Shamir* heuristic [FS86], for reducing interaction in (public-coin) interactive proofs. Loosely speaking, the Fiat-Shamir heuristic uses a cryptographic hash-function to compute the verifier's messages, and the resulting protocol is known to be secure in the random-oracle model [BR93]. However, replacing the random oracle with a concrete hash function may lead to an insecure protocol [CGH04,GK03], and so it is highly desirable to construct NIZK protocols whose security does not depend on random oracles. In recent works, Kalai *et al.* [KRR17] and Canetti *et al.* [CCRR18] construct hash functions for which the Fiat-Shamir heuristic is sound when applied to interactive *proofs* (i.e., with statistical soundness). However, they use very strong assumptions such as the existence of encryption schemes in which the success probability of a key-dependent message (KDM) key recovery attack succeeds only with *exponentially* small probability.

As mentioned above, Peikert and Vaikuntanathan [PV08] conjecture that a NIZK proof-system for GapSVP would suffice to obtain NIZK for all of **NP** based on LWE. [PV08] also suggest that one approach to proving this conjecture is to translate the prior approach of Blum *et al.* [BDSMP91], which referred to the quadratic residuosity problem, to lattices. Our approach differs from that suggested by [PV08] and is more similar to the [FLS99] paradigm.

Recently, Kim and Wu [KW18] showed a construction of multi-theorem NIZK argument for NP from standard lattice assumptions in the preprocessing model. In the preprocessing multi-theorem model, a trusted setup algorithm produces proving and verification keys, which are reusable for an unbounded number of theorems.

Zero-Knowledge Proofs for Specific Lattice Problems. Highly relevant to our assumption of a NIZK proof system for $\mathsf{dBDD}_{\alpha,\gamma}$ are several works on zero-knowledge of lattice problems. Goldreich and Goldwasser [GG00] show that the *complement* of GapSVP_γ and GapCVP_γ, with parameter $\gamma = \Theta(\sqrt{n}/\log n)$, has an honest-verifier SZK protocol. Combined with results on the structure of SZK (see [Vad99]), this implies that GapSVP_γ and GapCVP_γ themselves are in SZK. Subsequently, Micciancio and Vadhan [MV03] show that GapSVP_γ and GapCVP_γ are in SZK for the same approximation factor even with an efficient prover (given the shortest or closest lattice point, resp., as an auxiliary input). Building on the protocol of [MV03], Goldwasser and Kharchenko [GK05] use the connection between Atjai-Dwork ciphertexts and GapCVP to construct a proof of plaintext knowledge.

Peikert and Vaikuntanathan [PV08] construct *non-interactive* statistical zero-knowledge (NISZK) protocols for a variety of lattice problems and in particular leave the question of whether GapSVP_γ has a NISZK proof system as an open problem. Most recently, Alamati *et al.* [APSD17] construct NISZK and SZK protocols for approximating the smoothing parameter of a lattice.

Lastly, we mention that starting with the work of Stern *et al.* [Ste96], several works [KTX08,Lyu08,LNSW13,LLM+16,dPL17] have constructed zero-knowledge proofs for lattice problems in the context of identification schemes.

1.3 Technical Overview

Let $L \in \mathbf{NP}$ be an arbitrary $\mathbf{NP}$ language. Our goal is to construct a NIZK proof system for L. The starting point for our construction is an (unconditional) NIZK proof system for L in the *hidden-bits model*, a framework introduced by Feige *et al.* [FLS99] and made explicit by Goldreich [Gol01]. In the hidden-bits model, the prover P has access to a string of uniformly random bits $r \in \{0,1\}^N$. Given the input x and a witness w, the prover can decide to reveal some subset $I \subset [N]$ of the bits to the verifier, and in addition sends a proof-string π. The verifier, given only the input x, the *revealed* bits r_I, and the proof π, decides whether $x \in L$. Note that the unrevealed bits remain entirely hidden from the verifier. A hidden-bits proof is *zero-knowledge* if there exists a simulator S that generates a view that is indistinguishable from that of the verifier (including in particular the revealed bits r_I).

Feige *et al.* [FLS99] show that every $\mathbf{NP}$ language has a NIZK proof system in the hidden bits model. Furthermore, they show how to implement the hidden bits model, in a computational sense, using *(doubly enhanced) trapdoor permutations,*[2] thereby obtaining a NIZK proof system for **NP** under the same assumption.

Following Goldreich's presentation, we shall also aim to enforce the hidden-bits model using cryptography. In contrast to [FLS99,Gol01], however, rather than using trapdoor permutations, we shall use an encryption scheme that satisfies some strong yet natural properties. The main technical challenge will be in constructing an LWE-based encryption scheme that satisfies these properties.

We begin by describing the two most intuitive properties that we would like from our public-key encryption scheme (G, E, D).

1. **Oblivious Sampling of Ciphertexts:** Firstly, we require the ability to sample ciphertexts while remaining entirely oblivious to the underlying messages. More precisely, we assume that there exists an algorithm Sample that, given a public key pk, samples a random ciphertext $c \leftarrow \mathsf{Sample}(\mathsf{pk})$ such that the plaintext value $\sigma = D_{\mathsf{sk}}(c)$ is hidden, *even given the random coins used to sample c.*[3] Encryption schemes that have oblivious ciphertext sampling or OCS procedures are known in the literature (see, e.g., [GKM+00,GR13]).
2. NIZK **proof for Plaintext Value:** Secondly, we require a NIZK proof for a specific task, namely proving that a given ciphertext $c = E_{\mathsf{pk}}(\sigma)$ is an encryption of the bit σ (with respect to the public-key pk). Note that this is indeed an **NP** task, since the secret key sk is a witness to the fact that c is an encryption of σ.[4] In particular, we require that the honest prover strategy can be implemented efficiently given access to this witness (i.e., the secret key sk).

With these two ingredients in hand we can describe the high-level strategy for implementing the hidden-bits model. The idea is that the common random string will contain N sequences $\rho_1, \dots, \rho_N$ of random coins for the OCS procedure. Our NIZK prover chooses a public-key/secret-key pair $(\mathsf{pk}, \mathsf{sk})$ and generates the ciphertexts $c_1, \dots, c_N$, where $c_i = \mathsf{Sample}(\mathsf{pk}; \rho_i)$ (i.e., an obliviously sampled ciphertext with respect to the public key pk and randomness ρ_i). The prover further computes the corresponding plaintext bits $\sigma_1, \dots, \sigma_N$, where $\sigma_i = \mathsf{Dec}_{\mathsf{sk}}(c_i)$ (which it can compute efficiently, since it knows the secret key sk). The prover now runs the hidden-bits prover with respect to the random bit sequence $(\sigma_1, \dots, \sigma_N)$ and obtains in return a subset $I \subseteq [N]$ of coordinates and a proof-string π. To reveal the coordinates $(\sigma_i)_{i \in I}$, we use the second ingredient: our NIZK proof for proving the plaintext value of the ciphertexts $(c_i)_{i \in I}$. Intuitively, the OCS guarantee allows the other bits $(\sigma_i)_{i \notin I}$ to remain hidden.

---

[2] *Doubly enhanced* trapdoor permutations were actually introduced in [Gol11] (with the motivation of implementing the hidden-bits model). See further discussion in [GR13,CL17].

[3] In particular, the naive algorithm that chooses at random $b \in \{0, 1\}$ and outputs $E_{\mathsf{pk}}(b)$ is *not* oblivious since its random coins fully reveal b.

[4] For simplicity, we focus for now on schemes with perfect correctness.

Certifying Public Keys. An issue that we run into when trying to implement the blueprint above is that a cheating prover may choose to specify a public key pk that is not honestly generated. Given such a key, it is not clear a priori that the prover cannot control the distribution of the hidden bits, or even equivocate by being able to claim that a single ciphertext c_i is both an encryption of the bit 0 and an encryption of the bit 1. This leads to actual attacks that entirely break the soundness of the NIZK proof system.

A closely related issue actually affects the [FLS99] NIZK construction (based on doubly enhanced trapdoor permutations) and was pointed out by Bellare and Yung [BY96].[5] More specifically, in the [FLS99] protocol the prover needs to specify the index of a permutation (which is analogous to the public key in our setting). However, [BY96] observed that if the prover specifies a function that is *not* a permutation, then it can violate soundness. They resolved this issue by constructing a NIZK proof system for proving that the index indeed specifies a permutation.[6]

We follow the [BY96] approach by requiring conditions (1) and (2) above, as well as a NIZK proof for certifying public keys. Thus, our NIZK prover also supplies a NIZK proof that the public key that it provides is valid.

Instantiating our Approach with LWE So far the approach outlined is basically the [FLS99] implementation of the hidden bits model (where we replace the trapdoor permutations with a suitable encryption scheme). However, when trying to instantiate it using LWE, we encounter significant technical challenges.

For our encryption scheme, we will use Regev's [Reg09] scheme which uses n-dimensional vectors over the integer ring $\mathbb{Z}_q$. The public key in this scheme consists of (1) a matrix $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$, where $m = \Theta(n \cdot \log(q))$, and (2) a vector $\mathbf{b}^T = \mathbf{s}^T \cdot \mathbf{A} + \mathbf{e}^T$, where $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ is the secret key, and $\mathbf{e}$ is drawn from an n-dimensional discrete Gaussian.

To instantiate the approach outlined above we require three procedures: (1) an oblivious ciphertext sampler (OCS), (2) a NIZK proof system for plaintext values, and (3) a NIZK proof system for certifying public keys. We discuss these three requirements in increasing order of complexity.

NIZK *proof for Validating Public Keys.* Recall that a public key in this encryption scheme is of the form $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m$, where $\mathbf{b}^T = \mathbf{s}^T \cdot \mathbf{A} + \mathbf{e}^T$ for error vector $\mathbf{e} \in \mathbb{Z}_q^m$ drawn from a discrete Gaussian and in particular having bounded entries (with all but negligible probability). To validate the public key we shall construct a NIZK proof system that proves that for the input public key $(\mathbf{A}, \mathbf{b})$, there exists a vector $\mathbf{s} \in \mathbb{Z}_q^n$ such that $\mathbf{s}^T \cdot \mathbf{A}$ is very close to $\mathbf{b}^T$.[7]

[5] Further related issues were recently uncovered by Canetti and Lichtenberg [CL17].

[6] Actually, the [BY96] protocol only certifies that the index specifies a function that is *close* to a permutation (i.e., they provide a *non-interactive* zero-knowledge proof of proximity, a notion recently formalized by Berman *et al.* [BRV17]) which suffices in this context.

[7] Actually, it is important for us to also establish that $\mathbf{s}$ is *unique*. We enforce this by having the matrix $\mathbf{A}$ be specified as part of the CRS (rather than by the prover).

Producing such a NIZK proof system is where we need (for the first time) our additional assumption that dBDD has a NIZK proof-system. Indeed, proving that there exists $\mathbf{s} \in \mathbb{Z}_q^n$ such that $\mathbf{s}^T \cdot \mathbf{A}$ is very close to $\mathbf{b}^T$ is a dBDD instance: we must show that the distance of the vector $\mathbf{b}$ from the lattice spanned by the rows of $\mathbf{A}$ is a lot smaller than the length of the shortest non-zero vector of this lattice. We note that since the matrix $\mathbf{A}$ is random (it will part of the CRS), we know that (with very high probability) the length of the shortest non-zero vector is large.

NIZK *proof for Plaintext Value.* The second procedure that we need is a NIZK proof-system that certifies that a given ciphertext encrypts a bit σ. To see how we obtain this, we first need to recall the encryption procedure in Regev's [Reg09] scheme. To encrypt a bit $\sigma \in \{0,1\}$, one selects at random $\mathbf{r} \leftarrow \{0,1\}^m$ and outputs the ciphertext $(\mathbf{c}, \omega)$, where $\mathbf{c} = \mathbf{A} \cdot \mathbf{r}$ and $\omega = \mathbf{b}^T \cdot \mathbf{r} + \sigma \cdot \left\lfloor \frac{q}{2} \right\rfloor$.

Thus, given an alleged public key $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m$ and ciphertext $(\mathbf{c}, \omega) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, we basically want to ensure that there exists a vector $\mathbf{s} \in \mathbb{Z}_q^n$ such that $\mathbf{b}^T \approx \mathbf{s}^T \cdot \mathbf{A}$ and $\omega + \sigma \cdot \left\lfloor \frac{q}{2} \right\rfloor \approx \mathbf{s}^T \cdot \mathbf{c}$, where $\sigma \in \{0,1\}$ is the alleged plaintext value. Put differently, we want to ensure that the vector $\left[\mathbf{b}, \left(\omega + \sigma \cdot \left\lfloor \frac{q}{2} \right\rfloor\right)\right]$ is *close* to the lattice spanned by the rows of $[\mathbf{A}, \mathbf{c}]$. Thus, this problem can also be reduced to an instance of dBDD.

Oblivious Sampling of Ciphertexts. The last ingredient that we need is a procedure for obliviously sampling ciphertexts in Regev's encryption scheme. This is the main technical challenge in our construction.

A first idea for such an OCS procedure is simply to generate a random pair $(\mathbf{c}, \omega)$, where $\mathbf{c} \leftarrow \mathbb{Z}_q^n$ and $\omega \leftarrow \mathbb{Z}_q$. Intuitively, this pair corresponds to a high noise encryption of a random bit. The problem though is precisely the fact that $(\mathbf{c}, \omega)$ is a *high noise* ciphertext. That is, $\mathbf{s}^T \cdot \mathbf{c} - \omega$ will be close to neither 0 nor $\lfloor q/2 \rfloor$. In particular, the above NIZK proof for certifying plaintext values only works for *low noise* ciphertexts.

This issue turns out to be a key one which we do not know how to handle directly. Instead, we shall bypass it by introducing and considering a generalization of OCS in which the (untrusted) prover is allowed to assist the verifier to perform the sampling. We refer to this procedure (or rather protocol) as a *prover-assisted oblivious ciphertext sampler* (POCS). Thus, a POCS is a protocol between a sampler S, which is given the secret key (and will be run by the prover in our NIZK proof), and a checker C which is given the public key (and will be run by the verifier). The common input to the protocol is a *random* string ρ. The sampler basically generates a sampled ciphertext c and sends it to the checker, who runs some consistency checks. If the sampler behaves honestly and ρ is sampled randomly, then the sampled ciphertext c should correspond to an encryption of a random bit σ and the checker's validation process should pass. Furthermore, the protocol should satisfy the following (loosely stated) requirements:

Indeed, it is not too difficult to show that a lattice spanned by a *random* matrix $\mathbf{A}$ does not have short vectors and therefore $\mathbf{b}$ cannot be close to two different lattice points.

- **(Computational) Hiding:** The value $\sigma = \mathsf{Dec}_{\mathsf{sk}}(c)$ is computationally hidden from the checker. That is, it is computationally infeasible to predict the value of σ from c and pk, even given the random coins ρ.
- **(Statistical) Binding:** For any value of ρ there exists a *unique* value σ such that *for every* (possibly cheating) sampler strategy S^*, with high probability either the checker rejects or the generated ciphertext c corresponds to an encryption of σ.

With some care, such a POCS procedure can replace the OCS procedure (which did not use a prover) in our original outline. The key step therefore is constructing a POCS procedure for Regev's encryption scheme, which we describe next.

A POCS *Procedure for Regev's Encryption Scheme.* Fix a public key $(\mathbf{A}, \mathbf{b})$ and let $\mathbf{s}$ be the corresponding secret key. The random input string for our POCS procedure consists of a vector $\boldsymbol{\rho} \in \mathbb{Z}_q^n$ and a value $\tau \in \mathbb{Z}_q$. The pair $(\boldsymbol{\rho}, \tau)$ should be thought of as a (high noise) Regev encryption. Denote by $e = \tau - \mathbf{s}^T \cdot \boldsymbol{\rho}$ the noise in this ciphertext.

As discussed above, since $(\boldsymbol{\rho}, \tau)$ corresponds to a high noise ciphertext, we cannot have the sampler just output it as is. Rather we will have the sampler output a value $\tau' = \mathbf{s}^T \cdot \boldsymbol{\rho} + e' + \sigma' \cdot \left\lfloor \frac{q}{2} \right\rfloor$, where e' is drawn from the same noise distribution as fresh encryptions (i.e., low noise), and the value of the encrypted bit σ' will be specified next. Observe that $(\boldsymbol{\rho}, \tau')$ corresponds to a *fresh* encryption of σ', and so we will need to make sure that σ' is random and that the hiding and binding properties hold.

To do so, we will define σ' as follows: If $|e' - e| \leq q/4$, then set $\sigma' = 0$, and otherwise set $\sigma' = 1$. Observe that in either case it must be that

$$\left| e' + \sigma' \cdot \left\lfloor \frac{q}{2} \right\rfloor - e \right| \leq q/4 \,. \tag{1}$$

We would like our checker to enforce that Eq. (1) holds. Initially this seems problematic since our checker has access to none of e, e', and σ'. However, the checker does have access to τ and τ', and it holds that:

$$|\tau' - \tau| = \left| \mathbf{s}^T \cdot \boldsymbol{\rho} + e' + \sigma' \cdot \left\lfloor \frac{q}{2} \right\rfloor - \mathbf{s}^T \cdot \boldsymbol{\rho} - e \right| = \left| e' + \sigma' \cdot \left\lfloor \frac{q}{2} \right\rfloor - e \right|$$

and so we simply have our checker verify that $|\tau' - \tau| \leq q/4$.

It is not too hard to see that σ' is an unbiased bit in this construction. Moreover, it is unbiased even conditioned on $\boldsymbol{\rho}$ (since its value is entirely undetermined until τ is chosen). Thus, the checker only sees a fresh encryption of a random bit σ' which, by the hardness of LWE, hides the value of σ'.

To see that the scheme is binding, observe that for most choices of $\boldsymbol{\rho}$ and τ the (cheating) sampler cannot equivocate to two values τ' and τ'' which correspond to different plaintext bits, as long as both have small noise. Hence, the sampler *cannot* equivocate to two different valid ciphertexts. This concludes the overview of our construction.

1.4 Organization

In Section 2 we provide definitions and notation used throughout this work (defining in particular NIZK and the hidden bits model, as well as giving sufficient background on lattices). In Section 3 we formalize our abstraction of "prover-assisted oblivious ciphertext sampling" (POCS) and show that encryption schemes admitting such a procedure (as well as some specific NIZK proof systems) imply NIZKs for **NP**. Finally, in Section 4 we show how to instantiate the foregoing framework using LWE.

2 Preliminaries

We follow the notation and definitions as in [Gol01].

For a distribution μ, we use $x \leftarrow \mu$ to denote that x is sampled from the distribution μ, and for a set S we use $x \leftarrow S$ to denote that x is sampled uniformly at random from the set S. We use $X \overset{c}{\approx} Y$, $X \overset{s}{\approx} Y$ and $X \equiv Y$ to denote that the distributions X and Y are computationally indistinguishable, statistically close and identically distributed, respectively (where in the case of computational indistinguishability we actually refer to ensembles of distributions parameterized by a security parameter).

2.1 Public-key Encryption with Public Randomness

For simplicity we restrict our attention to bit-encryption schemes (where messages consist of single bits). We will define a variant of public-key encryption in which all algorithms, including the adversary, have access to some public randomness. We emphasize that this public randomness is an additional input to the key generation algorithm and is revealed also to the adversary. In addition to the public randomness, the key generation algorithm is allowed to toss additional *private* random coins that are not revealed. To avoid cluttering notation, we will assume that the public key includes the public randomness.

Definition 1 (Public-Key Encryption with Public Randomness). *A* public-key encryption scheme with public randomness *is a triple of PPT algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *such that:*

1. *The* key-generation algorithm $\mathsf{Gen}(1^\kappa, \rho_{\mathsf{pk}})$ *on input* public randomness ρ_{pk} *(and while tossing additional private random coins) outputs a pair of keys* $(\mathsf{pk}, \mathsf{sk})$*, where* pk *includes* ρ_{pk}.
2. *The* encryption algorithm $\mathsf{Enc}(\mathsf{pk}, \sigma)$*, where* $\sigma \in \{0,1\}$*, outputs a ciphertext* c*. We denote this output by* $c = \mathsf{Enc}_{\mathsf{pk}}(\sigma)$.
3. *The deterministic* decryption algorithm $\mathsf{Dec}(\mathsf{sk}, c)$ *outputs a message* σ'*. We denote this output by* $\sigma' = \mathsf{Dec}_{\mathsf{sk}}(c)$.

We require that for every $\sigma \in \{0,1\}$*, except with negligible probability over the public randomness* ρ_{pk}*, the keys* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa, \rho_{\mathsf{pk}})$ *and the randomness of the encryption scheme, we have that* $\mathsf{Dec}_{\mathsf{sk}}(\mathsf{Enc}_{\mathsf{pk}}(\sigma)) = \sigma$.

Semantic security [GM84] is defined as follows:

Definition 2 (Semantic Security with Public Randomness). *A public-key encryption scheme with public randomness is* semantically secure *if the distributions* $(\mathsf{pk}, E_{\mathsf{pk}}(0))$ *and* $(\mathsf{pk}, E_{\mathsf{pk}}(1))$ *are computationally indistinguishable, where* $\rho_{\mathsf{pk}} \leftarrow \{0,1\}^{\mathrm{poly}(\kappa)}$ *and* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa, \rho_{\mathsf{pk}})$.

Note that, clearly, any public-key encryption scheme is also a public-key scheme with public randomness, where ρ_{pk} is null. Nevertheless, this notion will be useful in our constructions.

2.2 Non-Interactive Zero-Knowledge Proofs

Non-interactive Zero-knowledge Proofs are a fundamental cryptographic primitive introduced by Blum *et al.* [BFM88].

Definition 3 (NIZK). *A* non-interactive (computational) zero-knowledge proof system (NIZK) *for a language L is a pair of probabilistic polynomial-time algorithms (P, V) such that:*

- **Completeness:** *For every $x \in L$ and witness w for x, we have*
$$\Pr_R \left[V(x, R, P(x, R, w)) = 1\right] > 1 - \mathrm{negl}(|x|)$$
where $R \leftarrow \{0,1\}^{\mathrm{poly}(|x|)}$. If the foregoing condition holds with probability 1, then we say that the NIZK *has* perfect completeness.
- **Soundness:** *For every $x \notin L$ and every (possibly inefficient) cheating prover P^*, we have*
$$\Pr_R \left[V(x, R, P^*(x, R)) = 1\right] < \mathrm{negl}(|x|)$$
where $R \leftarrow \{0,1\}^{\mathrm{poly}(|x|)}$.
- **Zero-Knowledge:** *There exists a probabilistic polynomial-time simulator S such that the ensembles $\{(x, R, P(x, R, w))\}_{x \in L}$ and $\{S(x)\}_{x \in L}$ are computationally indistinguishable, where $R \leftarrow \{0,1\}^{\mathrm{poly}(|x|)}$.*

The random input R received by both P and V is referred to as the common random string *or* CRS.

We extend the definition of NIZK to *promise problems* in the natural way.

We can further define a NIZK proof system with adaptive soundness by allowing the cheating prover to specify the input x as well as the purported witness w.

Definition 4 (Adaptive Soundness for NIZK). *A* NIZK *proof system (P, V) is* adaptively sound *if it satisfies the following property. For any $\kappa \in \mathbb{N}$ and any (possibly inefficient) cheating prover P^* producing output $(x, w) \in \{0,1\}^\kappa$, we have*
$$\Pr_{\substack{R, \\ (x,w) \leftarrow P^*(1^\kappa, R)}} [V(x, R, w) = 1 \textit{ and } x \notin L] < \mathrm{negl}(\kappa)\,.$$

Remark 1 (Achieving Adaptive Soundness). By standard amplification techniques, any ordinary NIZK proof may be transformed into one which is adaptively sound (see, e.g. [Gol01, Chapter 4]).

The Hidden Bits Model The hidden-bits model was introduced by Goldreich [Gol01, Section 4.10.2] as an appealing abstraction of the NIZK proof system of Feige, Lapidot and Shamir [FLS99].

Definition 5 (Hidden Bits Proof-System). *A* hidden-bits proof system *for a language L is a pair of PPT algorithms (P, V) such that the following conditions hold:*

- *(Completeness) For all $x \in L$ and witnesses w for x,*

$$\Pr[V(x, R_I, I, \pi) = 1] > 1 - \mathrm{negl}(|x|)\,,$$

  where R is a uniformly random string of bits (of length $\mathrm{poly}(|x|)$), $(I, \pi) \leftarrow P(x, R, w)$ for I a subset of the indices of R, and R_I is the substring of R corresponding to the indices in I.
- *(Soundness) For all $x \notin L$ and any computationally unbounded cheating prover P^*, we have*

$$\Pr[V(x, R_I, I, \pi) = 1] < \mathrm{negl}(|x|)$$

  where R again is a uniformly random string of bits and $(I, \pi) \leftarrow P^*(x, R)$.
- *(Zero-knowledge) There exists a probabilistic polynomial-time simulator S such that the ensembles $\{(x, R_I, I, \pi)\}_{x \in L}$ and $\{S(x)\}_{x \in L}$ are computationally indistinguishable, where R is a uniformly random string of bits and $(I, \pi) \leftarrow P(x, R)$.*

Feige et al. [FLS99] and Goldreich [Gol01] showed that *every* **NP** language has a hidden-bits proof system *unconditionally* (where the hidden-bits string is of polynomial length and the prover strategy is implemented efficiently given the **NP** witness).

Lemma 1 (See [Gol01, Section 4.10.2]). *For any language $L \in$ **NP**, there exists a zero-knowledge hidden-bits proof system for L. Moreover, the proof-system has perfect completeness.*

2.3 Lattices and Learning With Errors

In this section we give some basic definitions and lemmata about lattices and the Learning With Errors (LWE) assumption.

Standard Notation. We let the elements of the ring $\mathbb{Z}_q$ be identified with the representatives $\left\{-\left\lfloor \frac{q}{2} \right\rfloor, \ldots, \left\lceil \frac{q}{2} \right\rceil - 1\right\}$.

We denote by $[x, y]$ the concatenation of vectors or matrices. For example, if $\mathbf{x} \in \mathbb{Z}_q^n$ and $y \in \mathbb{Z}_q$, then $[\mathbf{x}, y]$ is a vector in $\mathbb{Z}_q^{n+1}$, whose first n components correspond to the n components of $\mathbf{x}$ and whose last component is y. Similarly, if $\mathbf{X} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{y} \in \mathbb{Z}_q^n$, then $[\mathbf{X}, \mathbf{y}]$ is a matrix in $\mathbb{Z}_q^{n \times (m+1)}$, whose last column is $\mathbf{y}$.

For $x \in \mathbb{Z}_q$, we denote by $|x|$ the value in $\left[0, \left\lfloor \frac{q}{2} \right\rfloor\right]$ such that $|x| = x$ if $x < q/2$ and $|x| = q - x$ otherwise. Namely, $|x|$ is the distance from 0 in $\mathbb{Z}_q$. Similarly, for $\mathbf{x} \in \mathbb{Z}_q^n$ we denote by $\|\mathbf{x}\|$ the ℓ_2 norm, namely $\|\mathbf{x}\| = \sqrt{\sum |x_i|^2}$, where x_i are the coordinates of $\mathbf{x}$ and $|\cdot|$ is as defined above.

Lastly, we denote by $\lfloor \cdot \rceil_q : \mathbb{Z}_q \to \{0, 1\}$ the function:

$$\lfloor x \rceil_q = \begin{cases} 0 & \text{if } x \in [-\lfloor q/4 \rfloor, \lceil q/4 \rceil] \\ 1 & \text{otherwise} \end{cases} .$$

Lattices A lattice Λ is an additive subgroup of $\mathbb{Z}^m$. Every lattice is finitely generated as all integer linear combinations of a set of *linearly independent row vectors*[8] $\mathbf{B}$. We call this set a basis for the lattice and its cardinality the rank of the lattice. We denote by $\Lambda(\mathbf{A})$ the lattice that is generated by the *rows* of $\mathbf{A}$ (which might or might not be a basis) and by $\mathbf{B}(\mathbf{A})$ a basis of the lattice $\Lambda(\mathbf{A})$. We denote by $\lambda_1(\Lambda)$ the length of the shortest nonzero lattice vector:

$$\lambda_1(\Lambda) = \min_{\mathbf{x} \in \Lambda \setminus \{\mathbf{0}\}} \|\mathbf{x}\| .$$

We note the following standard lemma about lattice bases.

Lemma 2. *Let $\mathbf{A} \in \mathbb{Z}^{n \times m}$ with $m \geq n$, there is an efficient algorithm to compute $\mathbf{B}(\mathbf{A})$. Namely, given a generating set of a lattice, we can efficiently compute a basis for the same lattice.*

A special family of lattices with numerous applications in cryptographic is the family of q-ary lattices.

Definition 6. *A lattice Λ is called a q-ary lattice if $q\mathbb{Z}^m \subseteq \Lambda$. Equivalently, Λ is q-ary if $\mathbf{x} \in \Lambda$ if and only if $(\mathbf{x} \bmod q) \in \Lambda$.*

We denote a q-ary lattice by Λ_q. More specifically, if $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ then we denote by $\Lambda_q(\mathbf{A})$ the lattice:

$$\Lambda_q(\mathbf{A}) = \{\mathbf{y} \in \mathbb{Z}^m : \exists \mathbf{s} \in \mathbb{Z}_q^n \text{ s.t. } \mathbf{y}^T = \mathbf{s}^T \mathbf{A}\} + q\mathbb{Z}^m.$$

Decisional Bounded Distance Decoding Problem We define some well-studied lattice problems as well as the *decisional Bounded Distance Decoding (*dBDD*)* variant which we use extensively in this work. We also present a reduction from dBDD to the GapSVP problem, showing that dBDD is (up to polynomial loss in the parameters) at most as hard as GapSVP.

Definition 7. *For a given parameter $\gamma > 1$, the promise problem* $\mathsf{GapSVP}_\gamma = (\mathsf{YES}, \mathsf{NO})$ *with input a basis $\mathbf{B} \in \mathbb{Z}^{n \times m}$ and parameter $r > 0$ is defined as:*

[8] In the literature, typically $\mathbf{B}$ is defined as a set of column vectors. However, for our applications it is more convenient to use row vectors.

– $(B, r) \in \mathsf{YES}$ *if* $\lambda_1(\Lambda(\mathbf{B})) < r$, *and*
– $(B, r) \in \mathsf{NO}$ *if* $\lambda_1(\Lambda(\mathbf{B})) > \gamma r$.

Definition 8. *For a given parameter* $\alpha \geq 1$*, the promise search problem* BDD_α *with input a basis* $\mathbf{B} \in \mathbb{Z}^{n\times m}$*, a target vector* $\mathbf{t} \in \mathbb{R}^m$ *such that* $\mathrm{dist}(\Lambda(\mathbf{B}), \mathbf{t}) < \frac{\lambda_1(\mathbf{B})}{\alpha}$ *outputs a lattice vector* $\mathbf{v} \in \Lambda(\mathbf{B})$ *such that* $\|\mathbf{t} - \mathbf{v}\| = \mathrm{dist}(\Lambda(\mathbf{B}), \mathbf{t})$.

We define a decisional version of the BDD_α problem.

Definition 9. *For two given parameters* $\alpha \geq 1$ *and* $\gamma > 1$*, the promise problem* $\mathsf{dBDD}_{\alpha,\gamma} = (\mathsf{YES}, \mathsf{NO})$ *with input a basis* $\mathbf{B} \in \mathbb{Z}^{n\times m}$ *and a target vector* $\mathbf{t} \in \mathbb{R}^m$ *is defined as:*

– $(\mathbf{B}, \mathbf{t}) \in \mathsf{YES}$ *if* $\mathrm{dist}(\mathbf{t}, \Lambda(\mathbf{B})) \leq \frac{\lambda_1(\Lambda(\mathbf{B}))}{\alpha}$; *and*
– $(\mathbf{B}, \mathbf{t}) \in \mathsf{NO}$ *if* $\mathrm{dist}(\mathbf{t}, \Lambda(\mathbf{B})) > \gamma \cdot \frac{\lambda_1(\Lambda(\mathbf{B}))}{\alpha}$.

In order to establish the complexity of the dBDD problem, we show that it is at most as hard as the well studied lattice problem GapSVP.

Proposition 1. *The problem* $\mathsf{dBDD}_{\alpha,\gamma}$ *is Cook-reducible to* $\mathsf{GapSVP}_{\min(\sqrt{\gamma},\alpha/2)}$ *where* γ *and* α *are polynomially-bounded.*

Proof. Let $(\mathbf{B}, \mathbf{t})$ be an input of $\mathsf{dBDD}_{\alpha,\gamma}$. First, using binary search and a $\mathsf{GapSVP}_{\sqrt{\gamma}}$ oracle, we compute an r such that $\frac{\lambda_1(\mathbf{B})}{\sqrt{\gamma}} \leq r \leq \sqrt{\gamma} \cdot \lambda_1(\mathbf{B})$.

From [LM09], BDD_α is reducible to $\mathsf{GapSVP}_{\alpha/2}$ with α polynomially-bounded. Therefore, the $\mathsf{GapSVP}_{\alpha/2}$ oracle returns an alleged closest vector $\mathbf{v}$ to $\mathbf{t}$. If $\mathbf{v} \in \Lambda(\mathbf{B})$ and $\|\mathbf{t} - \mathbf{v}\| \leq \sqrt{\gamma} \cdot \frac{r}{\alpha}$, we output 1. Else, we output 0.

Indeed, if $\mathsf{dBDD}_{\alpha,\gamma}(\mathbf{B}, \mathbf{t}) \in \mathsf{YES}$, then there is a vector $\mathbf{v} \in \Lambda(\mathbf{B})$ such that $\|\mathbf{t} - \mathbf{v}\| \leq \frac{\lambda_1(\mathbf{B})}{\alpha} \leq \sqrt{\gamma} \cdot \frac{r}{\alpha}$ and $\mathsf{GapSVP}_{\alpha/2}$ returns this vector. On the other hand, if $\mathsf{dBDD}_{\alpha,\gamma}(\mathbf{B}, \mathbf{t}) \in \mathsf{NO}$, then for every vector $\mathbf{v} \in \Lambda(\mathbf{B})$ it holds that $\|\mathbf{t} - \mathbf{v}\| > \gamma \cdot \frac{\lambda_1(\mathbf{B})}{\alpha} \geq \sqrt{\gamma} \cdot \frac{r}{\alpha}$, so there is no vector $\mathbf{v}$ for which we output 1.

We remark that even though there is a reduction from dBDD to GapSVP, a NIZK proof system for GapSVP does not automatically imply a NIZK proof system for dBDD since it is a *Cook* reduction (rather than a Karp reduction). In particular, we do not know if a NIZK for GapSVP implies a NIZK for dBDD.

Learning with Errors We proceed to define the main cryptographic assumption we use: Learning With Errors (LWE). First, we define the (one-dimensional) discrete Gaussian distribution:

Definition 10. *For* $q \in \mathbb{N} \setminus \{0\}$ *and parameter* $\beta > 0$*, the* discrete Gaussian probability distribution χ_β *is simply the Gaussian distribution restricted to* $\mathbb{Z}_q$*:*

$$\chi_\beta(x) \propto \begin{cases} \exp(-\pi|x|^2/(\beta q)^2) & \text{if } x \in [-\lfloor q/2 \rfloor, \lceil q/2 \rceil] \cap \mathbb{Z} \\ 0 & \text{otherwise} \end{cases}$$

With the definition of the Discrete Gaussian distribution in hand, we are ready to define LWE:

Definition 11. *The* (Decisional) Learning With Error (LWE) assumption *with parameters n, q, β, denoted by $\mathsf{LWE}_{n,q,\beta}$, states that:*

$$(\mathbf{A}, \mathbf{b}) \stackrel{c}{\approx} (\mathbf{A}, \mathbf{r})$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$ with $m = \mathrm{poly}(n, \log(q))$, $\mathbf{b}^T = \mathbf{s}^T\mathbf{A} + \mathbf{e}^T$, $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow \chi_\beta^m$ and $\mathbf{r} \leftarrow \mathbb{Z}_q^m$.

We utilize the fact that if $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$ with m large enough, then there is a *unique* $\mathbf{s}$ such that $\mathbf{b}^T \approx \mathbf{s}^T\mathbf{A}$. The proof of this fact follows from bounding the shortest vector in the lattice and observing that if $\mathbf{s}_1, \mathbf{s}_2$ are such that $\mathbf{s}_1^T\mathbf{A} \approx \mathbf{s}_2^T\mathbf{A}$, then $(\mathbf{s}_1^T - \mathbf{s}_2^T)\mathbf{A} \approx \mathbf{0}$. The following lemma can be shown by a standard argument with a union bound over all nonzero vectors $\mathbf{s} \in \mathbb{Z}_q^n$.

Lemma 3. *Let $n, q \in \mathbb{N}$, and $m \geq 2n\log(q)$. Then*

$$\Pr_{\mathbf{A}\leftarrow\mathbb{Z}_q^{n\times m}}\left[\lambda_1(\Lambda_q(\mathbf{A})) \leq q/4\right] \leq q^{-n}\,.$$

3 From Prover-Assisted Oblivious Sampling to NIZKs

In this section we introduce the abstraction of a prover-assisted procedure for oblivious ciphertext sampling (POCS) for a public-key encryption scheme (as outlined in the introduction), and show how to combine this notion with NIZK proofs of the validity of public keys and plaintext values to obtain NIZK proofs for any **NP** language.

3.1 Definitions: Valid Public Keys, Ciphertexts and POCS

The first definition we consider is the notion of a *valid* set $\mathcal{PK}$ of public keys. Intuitively, we would like this set to correspond precisely to public keys in the support of the key-generation algorithm. However, due to specifics of our instantiation with LWE, we need to be more lenient and allow public keys that are not quite in the support of the key-generation algorithm but are nevertheless sufficiently well-formed (e.g., keys with a higher level noise).

Loosely speaking, a *valid* public key pk is associated with two sets $C_{\mathsf{pk}}^{(0)}$ and $C_{\mathsf{pk}}^{(1)}$, which correspond to "valid" ciphertexts with respect to that key of messages 0 and 1, respectively. We first require that honestly sampled public keys be valid. We further require that for all valid public keys (i.e., even those not in the support of the key generation algorithm), the associated sets $C_{\mathsf{pk}}^{(0)}$ and $C_{\mathsf{pk}}^{(1)}$ are disjoint (i.e. no ciphertext is a valid encryption both of 0 and of 1).[9]

[9] Note that in the actual definition we only require the latter to hold *with high probability over the choice of the public randomness* for every valid public key. The notion of encryption schemes with public randomness is discussed in Section 2.1.

Definition 12 (Valid Public Keys). *Let* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a public-key encryption scheme with public randomness. For a given security parameter* κ*, let* $\mathcal{VPK} = (\mathcal{VPK}_\kappa)_{\kappa \in \mathbb{N}}$ *be an ensemble of sets, where for each* $\kappa \in \mathbb{N}$*, each* $\mathsf{pk} \in \mathcal{VPK}_\kappa$ *is associated with a pair of sets* $\left(C^{(0)}_{\mathsf{pk}}, C^{(1)}_{\mathsf{pk}}\right)$ *and public randomness* ρ_{pk}*. We say that* $\mathcal{VPK}$ *is* valid *if it satisfies the following properties.*

1. *For all* $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Gen}(1^\kappa, \cdot)$*, we have* $\mathsf{pk} \in \mathcal{VPK}_\kappa$*.*
2. *For every* $b \in \{0, 1\}$ *we have that* $c_b \in C^{(b)}_{\mathsf{pk}}$ *with all but negligible probability over the choice of public randomness* ρ_{pk}*, keys* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa, \rho_{\mathsf{pk}})$*, and ciphertext* $c_b \leftarrow \mathsf{Enc}_{\mathsf{pk}}(b)$*.*
3. *With all but negligible probability over the public randomness* ρ_{pk}*, for all* $\mathsf{pk} \in \mathcal{VPK}_\kappa$ *with public randomness* ρ_{pk}*, it holds that* $C^{(0)}_{\mathsf{pk}} \cap C^{(1)}_{\mathsf{pk}} = \emptyset$*.*

We next formalize the notion of a *prover-assisted oblivious ciphertext sampler* (POCS). This is an extension of oblivious ciphertext samplers (OCS), which (to the best of our knowledge) were introduced by Gertner *et al.* [GKM+00]. An OCS procedure allows one to sample a ciphertext so that the underlying plaintext remains hidden. In this work we introduce a relaxation of this notion in which the sampling is assisted by an *untrusted* prover.

More specifically, a POCS protocol consists of two procedures, a *sampler* and a *checker*, which both have access to a shared random string ρ. The sampler also receives as input the secret-key of the scheme and generates a ciphertext c. The checker receives c, as well as the random string ρ and the public-key (but not the secret-key) and performs a test to ensure that c encodes an unbiased bit depending on the randomness ρ. Jumping ahead, we remark that the role of the sampler is played by the *prover* in our NIZK, whereas the role of the *checker* is played by the verifier.

We require that the POCS procedure satisfy the following loosely stated properties:

1. For honestly sampled ciphertexts c, the checker should accept with overwhelming probability.
2. Given pk, ρ and an honestly sampled ciphertext c, the corresponding plaintext bit $\mathsf{Dec}_{\mathsf{sk}}(c)$ is computationally hidden.
3. For a given random string ρ, except with a small probability there should not exist both an encryption c_0 of 0 and an encryption c_1 of 1 that pass the checker's test. Thus, for any given ciphertext (even a maliciously generated one) that passes the test, the corresponding plaintext bit is almost always fully determined.
4. The sampled plaintext bit should be (close to) unbiased. The latter should hold even with respect to a *malicious* sampler. In our actual instantiation of POCS (via LWE, see Section 4), the plaintext bit will have a small but noticeable (i.e., inverse polynomial) bias. Thus, our definition of POCS leaves the bias as a parameter, which we denote by ϵ.
5. The procedure satisfies the following "zero-knowledge like" simulation property: given only the public-key pk and plaintext bit σ, it should be possible

to generate the distribution (ρ, c) of the sampling procedure, conditioned on $\mathsf{Dec}_{\mathsf{sk}}(c) = \sigma$. This property is captured by the EncryptAndExplain procedure below. In our actual formalization we only require that this property holds in a computational sense (i.e., the simulated distribution should only be computationally indistinguishable from the actual sampling procedure). While a statistical requirement may seem like a more natural choice here, we use a computational notion due to a technical consideration in the LWE instantiation. See Section 4.3 for details.

We proceed to the formal definition of a POCS encryption scheme.

Definition 13 (Prover-assisted Oblivious Ciphertext Sampler (POCS)). *For a parameter $\epsilon = \epsilon(\kappa) \in [0, 1]$, a $(1 - \epsilon(\kappa))$-binding* prover-assisted oblivious ciphertext sampler (POCS)*, with respect to a* valid *set of public keys $\mathcal{VPK} = \{\mathcal{VPK}_\kappa\}_{\kappa \in \mathbb{N}}$ for an encryption scheme* (Gen, Enc, Dec) *with public randomness, is a triple of PPT algorithms* Sample*,* Check*, and* EncryptAndExplain *satisfying the following properties:*

- **Complete:**

$$\Pr_{\substack{\rho_{\mathsf{pk}}, \rho \leftarrow \{0,1\}^{\mathrm{poly}(\kappa)} \\ (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa, \rho_{\mathsf{pk}})}} \Big[\mathsf{Check}\big(\mathsf{pk}, \rho, \mathsf{Sample}(\mathsf{sk}, \rho)\big) = 1 \Big] > 1 - \mathrm{negl}(\kappa).$$

- **Unbiased:** *For any $\kappa \in \mathbb{N}$, $\mathsf{pk} \in \mathcal{VPK}_\kappa$ and any $b \in \{0, 1\}$, we have that:*

$$\Pr_{\rho \leftarrow \{0,1\}^{\mathrm{poly}(\kappa)}} \Big[\exists c \in C_{\mathsf{pk}}^{(b)} \text{ such that } \mathsf{Check}(\mathsf{pk}, \rho, c) = 1 \Big] \geq 1/2 - \mathrm{negl}(\kappa).$$

- **Statistically binding:** *With probability $1 - \mathrm{negl}(\kappa)$ over the public randomness ρ_{pk}, we have for all $\mathsf{pk} \in \mathcal{VPK}_\kappa$ with public randomness ρ_{pk} that*

$$\Pr_{\rho \leftarrow \{0,1\}^{\mathrm{poly}(\kappa)}} \left[\exists c_0 \in C_{\mathsf{pk}}^{(0)}, c_1 \in C_{\mathsf{pk}}^{(1)} \ s.t. \ \begin{array}{l} \mathsf{Check}(\mathsf{pk}, \rho, c_0) = 1 \ \textit{and} \\ \mathsf{Check}(\mathsf{pk}, \rho, c_1) = 1 \end{array} \right] < \epsilon(\kappa).$$

  We emphasize that $\epsilon(\kappa)$ is a parameter and is not necessarily negligible.
- **Simulatable:** *For every $N = \mathrm{poly}(\kappa)$ it holds that:*

$$\Big(\mathsf{pk}, (\rho_i)_{i=1}^N, (c_i)_{i=1}^N, (\sigma_i)_{i=1}^N \Big) \overset{c}{\approx} \Big(\mathsf{pk}, (\rho'_i)_{i=1}^N, (c'_i)_{i=1}^N, (\sigma'_i)_{i=1}^N \Big),$$

  where $\rho_{\mathsf{pk}} \leftarrow \{0,1\}^{\mathrm{poly}(\kappa)}$, $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa, \rho_{\mathsf{pk}})$, and for every $i \in [N]$, it holds that $\rho_i \leftarrow \{0,1\}^{\mathrm{poly}(\kappa)}$, $c_i \leftarrow \mathsf{Sample}(\mathsf{sk}, \rho_i)$, and $\sigma_i = \mathsf{Dec}_{\mathsf{sk}}(c_i)$, $\sigma'_i \leftarrow \{0, 1\}$ and $(\rho'_i, c'_i) \leftarrow \mathsf{EncryptAndExplain}(\mathsf{pk}, \sigma')$.
- **Computationally hiding:** *Let $\rho_{\mathsf{pk}}, \rho \leftarrow \{0,1\}^{\mathrm{poly}(\kappa)}$, $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa, \rho_{\mathsf{pk}})$, and $c \leftarrow \mathsf{Sample}(\mathsf{sk}, \rho)$. Then, for all probabilistic polynomial-time adversaries $\mathcal{A}$,*

$$\Pr\big[\mathcal{A}(\mathsf{pk}, \rho, c) = \mathsf{Dec}_{\mathsf{sk}}(c)\big] \leq \frac{1}{2} + \mathrm{negl}(\kappa).$$

Remark 2 (Relaxing the Hiding Property). We remark that for our construction of NIZK a weaker hiding property suffices, in which the adversary is only given the random string ρ (but not the ciphertext c). Although this definition is strictly weaker, we find it less natural and choose to define the hiding property as specified above.

We next prove two useful propositions showing that the computational hiding property of the POCS implies a hiding property resembling semantic security for the $\mathsf{EncryptAndExplain}$ sampling algorithm. Specifically, we show that the encrypted bit remains hidden given both the ciphertext and the explaining randomness produced by the $\mathsf{EncryptAndExplain}$ algorithm. The intuition is analogous to the usage of the *double enhancement* property of trapdoor permutations in the construction of NIZKs (see, e.g., [GR13]).

Proposition 2. *Suppose* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *has a* $(1-\epsilon)$*-binding* POCS *with respect to an ensemble of valid public keys* $\mathcal{VPK}$*. Then, for all probabilistic polynomial-time adversaries* $\mathcal{A}$*,*

$$\Pr\left[\mathcal{A}(\mathsf{pk}, \rho, c) = \sigma\right] \leq \frac{1}{2} + \mathrm{negl}(\kappa),$$

where $\rho_{\mathsf{pk}}, \rho \leftarrow \{0,1\}^{\mathrm{poly}(\kappa)}$*,* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa, \rho_{\mathsf{pk}})$*,* $\sigma \in \{0,1\}$*, and* $(\rho, c) \leftarrow \mathsf{EncryptAndExplain}(\mathsf{pk}, \sigma)$*.*

Proof. This follows immediately from the simulatable and computationally hiding properties of the POCS.

Proposition 3. *Suppose* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *has a* $(1-\epsilon)$*-binding* POCS *with respect to an ensemble of public keys* $\mathcal{VPK}$*. It holds that*

$$(\mathsf{pk}, \rho_0, c_0) \overset{c}{\approx} (\mathsf{pk}, \rho_1, c_1),$$

where the public randomness $\rho_{\mathsf{pk}} \leftarrow \{0,1\}^{\mathrm{poly}(\kappa)}$*, the keys* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa, \rho_{\mathsf{pk}})$*,* $(\rho_0, c_0) \leftarrow \mathsf{EncryptAndExplain}(\mathsf{pk}, 0)$ *and* $(\rho_1, c_1) \leftarrow \mathsf{EncryptAndExplain}(\mathsf{pk}, 1)$*.*

Proof. This follows from Proposition 2 by a standard argument, similar to the equivalence of semantic security and indistinguishability of encryptions (see, e.g. [Gol04]).

We now define two promise problems for which we will later assume the existence of suitable NIZKs. The first problem that we consider is that of distinguishing public keys which are in the support of the key-generation algorithm (i.e., were honestly generated) from ones which are invalid (i.e., not in the set of valid public keys).

Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme and let us denote by $\mathcal{VPK}$ an ensemble of valid public-keys. We define the promise problem $\mathsf{GoodPK} = (\mathsf{GoodPK}_{\mathtt{Yes}}, \mathsf{GoodPK}_{\mathtt{No}})$ where:

$$\mathsf{GoodPK}_{\mathtt{Yes}} = \left\{\mathsf{pk} : \mathsf{pk} \in \bigcup_\kappa \mathsf{Gen}(1^\kappa)\right\}$$

$$\mathsf{GoodPK}_{\mathtt{No}} = \left\{\mathsf{pk} : \mathsf{pk} \notin \bigcup_\kappa \mathcal{VPK}_\kappa\right\}.$$

We also define a related promise problem GoodCT, which corresponds to triplets containing a public key, ciphertext and a single-bit message. Formally, the problem is defined as $\mathsf{GoodCT} = (\mathsf{GoodCT}_{\mathtt{Yes}}, \mathsf{GoodCT}_{\mathtt{No}})$, where:

$$\mathsf{GoodCT}_{\mathtt{Yes}} = \left\{ (\mathsf{pk}, c, b) : \mathsf{pk} \in \bigcup_{\kappa} \mathsf{Gen}(1^{\kappa}) \text{ and } c \in \mathsf{Enc}_{\mathsf{pk}}(b) \right\}$$
$$\mathsf{GoodCT}_{\mathtt{No}} = \left\{ (\mathsf{pk}, c, b) : \mathsf{pk} \in \bigcup_{\kappa} \mathcal{VPK}_{\kappa} \text{ and } c \notin C_{\mathsf{pk}}^{(b)} \right\}.$$

3.2 From POCS to NIZK

In this section we state and prove our transformation of encryption schemes that support POCS and suitable NIZKs for GoodPK and GoodCT, to general purpose NIZKs for $\mathbf{NP}$. This is captured by the following lemma:

Lemma 4. *Let* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a public-key encryption scheme with public randomness, and* $\mathcal{VPK}$ *be a valid set of public keys (as in Definition 12). Suppose the following conditions hold.*

– $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *has a* $(1-\epsilon)$*-binding* POCS *with respect to* $\mathcal{VPK}$*, for some sufficiently small* $\epsilon = 1/\mathrm{poly}(\kappa)$*.*
– *There is a* NIZK *for* GoodPK*.*
– *There is a* NIZK *for* GoodCT*.*

Then, there exists a NIZK *for every language* $L \in \mathbf{NP}$*.*

Proof. Let $L \in \mathbf{NP}$. By Lemma 1, there exists a hidden-bits zero knowledge proof system $(P_{\mathsf{hb}}, V_{\mathsf{hb}})$ for L (with perfect completeness). We shall use this proof-system to construct a NIZK for L, using the assumptions in the theorem's statement.

We first give a proof system satisfying a weak notion of soundness. Specifically, we shall weaken soundness by assuming that the cheating prover is constrained to choose a public-key pk before reading the CRS. To be more precise, since the public randomness of the pk comes from the CRS, the prover must choose the public key pk before reading any *other* part of the CRS. Also, the verifier is only required to reject inputs $x \notin L$ only with inverse polynomial probability (rather than with all but negligible probability). Using standard amplification techniques, we will subsequently transform this into a full-fledged NIZK (achieving the standard notion of soundness).

We assume without loss of generality that the NIZK proof systems that we have for GoodPK and GoodCT have *adaptive* soundness (see Remark 1). Our base NIZK protocol, achieving only the aforementioned weak soundness notion, is given in Protocol 1.

Protocol 1 *Let* $L \in \mathbf{NP}$*. Let* $(P_{\mathsf{pk}}, V_{\mathsf{pk}})$ *and* $(P_{\mathsf{ct}}, V_{\mathsf{ct}})$ *be adaptively sound* NIZK *proof systems for the promise problems* GoodPK *and* GoodCT*, respectively, and let* $(P_{\mathsf{hb}}, V_{\mathsf{hb}})$ *be a hidden-bits proof system for L that uses* $N = N(n)$ *hidden bits for inputs of length* $n \in \mathbb{N}$*. Consider the following non-interactive proof system.*

– *Input* $x \in \{0,1\}^n$.
– *Common random string* $\rho = (\rho_{\mathsf{pk}}, r_{\mathsf{pk}}, \rho_1, \ldots, \rho_N, r_1, \ldots, r_N)$.
– *Prover's witness* $w \in \{0,1\}^{\mathrm{poly}(n)}$.
– *Prover* P*, given* x*,* w *and* ρ*, performs the following:*
  1. *Let* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n, \rho_{\mathsf{pk}})$.
  2. *Let* $\pi_{\mathsf{pk}} \leftarrow P_{\mathsf{pk}}(\mathsf{pk}, r_{\mathsf{pk}}, \mathsf{sk})$.
  3. *For* $i \in [N]$*, let* $c_i \leftarrow \mathsf{Sample}(\mathsf{sk}, \rho_i)$ *and let* $b_i = \mathsf{Dec}_{\mathsf{sk}}(c_i)$.[10]
  4. *Let* $(I, \pi_{\mathsf{hb}}) \leftarrow P_{\mathsf{hb}}(x, (b_1, \ldots, b_m), w)$.
  5. *For* $i \in I$*, let* $\pi_i \leftarrow P_{\mathsf{ct}}((\mathsf{pk}, c_i, b_i), r_i, \mathsf{sk})$.
  6. *Let* $c_I = (c_i)_{i \in I}, b_I = (b_i)_{i \in I}, \pi_I = (\pi_i)_{i \in I}$.
  7. *Output* $\pi = (\mathsf{pk}, I, \pi_{\mathsf{pk}}, \pi_{\mathsf{hb}}, c_I, b_I, \pi_I)$.
– *Verifier* V *performs the following:*
  1. *Verify* NIZK *proofs by running* $V_{\mathsf{pk}}(\mathsf{pk}, r_{\mathsf{pk}}, \pi_{\mathsf{pk}})$ *and* $V_{\mathsf{ct}}((\mathsf{pk}, c_i, b_i), r_i, \pi_i)$ *for every* $i \in I$*. Reject if any of these tests rejects.*
  2. *Check that* $\mathsf{Check}(\mathsf{pk}, \rho_i, c_i) = 1$ *for every* $i \in I$*. Reject if any of these checks fail.*
  3. *Invoke* $V_{\mathsf{hb}}(x, b_I, I, \pi_{\mathsf{hb}})$*, and accept if and only if it accepts.*

Observe that both the verifier and prover are PPT algorithms. Thus, to show that Protocol 1 is a (weak) NIZK, we need to establish completeness, (weak) soundness and zero-knowledge.

Completeness. From the completeness of the NIZKs $(P_{\mathsf{pk}}, V_{\mathsf{pk}})$ and $(P_{\mathsf{ct}}, V_{\mathsf{ct}})$, we have that the verifiers V_{pk} and V_{ct} (for each $i \in [N]$) accept with all but negligible probability. By the completeness property of the POCS, we have that with all but negligible probability, the verifier's invocation of Check outputs 1 for each $i \in I$.

By the perfect completeness of the hidden-bits proof system, verifier V_{hb} accepts for $x \in L$.[11] Consequently, with probability $1 - \mathrm{negl}(n)$, all of the verifier's tests pass for $x \in L$ and a proof produced by the honest prover.

Zero-Knowledge. We first define the simulator S. Let S_{hb} be the simulator for the hidden bits proof-system $(P_{\mathsf{hb}}, V_{\mathsf{hb}})$, let S_{pk} be the simulator for the NIZK $(P_{\mathsf{pk}}, V_{\mathsf{pk}})$, and let S_{ct} be the simulator for the NIZK $(P_{\mathsf{ct}}, V_{\mathsf{ct}})$. On input $x \in \{0,1\}^n$, simulator S performs the following.

1. Sample public randomness ρ_{pk}, and let $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n, \rho_{\mathsf{pk}})$.
2. Sample $(\pi_{\mathsf{pk}}, r_{\mathsf{pk}}) \leftarrow S_{\mathsf{pk}}(\mathsf{pk})$ (recall that π_{pk} is the simulated proof string and r_{pk} is the simulated CRS).
3. Sample $(I, \pi_{\mathsf{hb}}, b_I) \leftarrow S_{\mathsf{hb}}(x)$, where $b_I = (b_i)_{i \in I}$. Set $b_i = 0$ for every $i \in [N] \setminus I$.

[10] Jumping ahead, we note that for our final NIZK protocol, achieving standard soundness, we will need to repeat steps 3–6 for $\ell = \mathrm{poly}(\kappa)$ times for the same pk to amplify soundness.

[11] Here we are utilizing the fact that the hidden-bits proof-system has *perfect* completeness to save us the effort of arguing that the hidden bits are indeed (sufficiently) unbiased.

4. For $i \in [N]$, sample $(\rho_i, c_i) \leftarrow \mathsf{EncryptAndExplain}(\mathsf{pk}, b_i)$.
5. For $i \in I$, sample $(\pi_i, r_i) \leftarrow S_{\mathsf{ct}}(\mathsf{pk}, c_i, b_i)$.
6. For $i \in [N] \setminus I$, let $r_i \leftarrow \{0,1\}^{\mathrm{poly}(n)}$.
7. Let $c_I = (c_i)_{i \in I}, \pi_I = (\pi_i)_{i \in I}$
8. Output simulated proof $\pi = (\mathsf{pk}, I, \pi_{\mathsf{pk}}, \pi_{\mathsf{hb}}, c_I, b_I, \pi_I)$ and simulated common random string $\rho = (\rho_{\mathsf{pk}}, r_{\mathsf{pk}}, \rho_1, \ldots, \rho_N, r_1, \ldots, r_N)$.

Due to lack of space, we defer the proof of indistinguishability of the real and simulated distributions to the full version [RSS18].

Weak soundness. We first prove a weak notion of soundness with respect to provers that are constrained to choose the public key pk before reading the CRS, other than the public randomness for generating the public-key. Subsequently we will apply an amplification argument to achieve full soundness.

Let us fix $x \notin L$ and a cheating prover P^*, and let us sample a CRS $\rho = (\rho_{\mathsf{pk}}, r_{\mathsf{pk}}, \rho_1, \ldots, \rho_N, r_1, \ldots, r_N)$. Let $\pi = (\mathsf{pk}, I, \pi_{\mathsf{pk}}, \pi_{\mathsf{hb}}, c_I, b_I, \pi_I)$ be the proof produced by P^* on input ρ, where P^* is first given only ρ_{pk} and produces pk, and subsequently is given the full CRS ρ and produces the rest of the proof π. By the adaptive soundness of the NIZKs $(P_{\mathsf{pk}}, V_{\mathsf{pk}})$ and $(P_{\mathsf{ct}}, V_{\mathsf{ct}})$, unless $\mathsf{pk} \in \mathcal{VPK}$ and $c_i \in C_{\mathsf{pk}}^{(b_i)}$ for each $i \in I$, the verifier V will reject with all-but-negligible probability. Additionally, with all-but-negligible probability, the public randomness ρ_{pk} in the CRS is such that the statistical binding property of the POCS holds. In the sequel we condition on these events occurring.

For a given valid public key $\mathsf{pk} \in \mathcal{VPK}$ and $\sigma \in \{0,1\}$, define $U_{\mathsf{pk}}^{(\sigma)}$ to be the set of randomnesses ρ (for the POCS procedure) that correspond to a ciphertext $c \in C_{\mathsf{pk}}^{(\sigma)}$ but no ciphertext in $C_{\mathsf{pk}}^{(1-\sigma)}$. That is,

$$U_{\mathsf{pk}}^{(\sigma)} = \left\{ \rho \in \{0,1\}^{\mathrm{poly}(\kappa)} : \exists c \in C_{\mathsf{pk}}^{(\sigma)} \text{ s.t. } \mathsf{Check}(\mathsf{pk}, \rho, c) = 1 \text{ and } \forall c' \in C_{\mathsf{pk}}^{(1-\sigma)}, \mathsf{Check}(\mathsf{pk}, \rho, c') = 0 \right\}.$$

The set $U_{\mathsf{pk}}^{(\sigma)}$ consists of randomness that can be uniquely interpreted as an encryption of σ and not of $1-\sigma$. Consequently, we have that $U_{\mathsf{pk}}^{(0)} \cap U_{\mathsf{pk}}^{(1)} = \emptyset$. By the unbiased and stastically binding properties of the POCS, we have that

$$\Pr_{\rho}\left[\rho \in U_{\mathsf{pk}}^{(\sigma)}\right] \geq 1/2 - \epsilon - \mathrm{negl}(\kappa),$$

where $\epsilon = \epsilon(\kappa)$ is the binding parameter of the POCS.

Note that $U_{\mathsf{pk}}^{(0)} \cap U_{\mathsf{pk}}^{(1)} = \emptyset$. Arbitrarily fix a set U_{pk} consisting half of elements of $U_{\mathsf{pk}}^{(0)}$ and half of elements of $U_{\mathsf{pk}}^{(1)}$ such that

$$\Pr_{\rho}\left[\rho \in U_{\mathsf{pk}}\right] \geq 1 - 2\epsilon - \mathrm{negl}(\kappa).$$

Recall that we first constrain the prover to choosing pk before reading any part of the CRS other than the public randomness ρ_{pk}. Let U_{pk} be the set defined above. Then, with probability $1 - 2\epsilon N$ the strings $\rho_1, \ldots, \rho_N$ are all in U_{pk}.

Conditioning on this event, we have that the sequence $b_1, \ldots, b_N$ is unbiased and uniquely determined by $\rho_1, \ldots, \rho_N$. Consequently, by the soundness of the hidden bits proof system $(P_{\mathsf{hb}}, V_{\mathsf{hb}})$ we have that with all but negligible probability, in this event V_{hb} will reject since $x \notin L$. Therefore, it follows that the verifier V will reject with probability at least $1 - 2\epsilon N - \mathrm{negl}(n)$, which is at least $1/3 - \mathrm{negl}(n)$ for $\epsilon = 1/N^2$.

Amplification. We now transform Protocol 1 into a protocol with full soundness.

We modify Protocol 1 as follows. After choosing the public key pk, the prover runs steps 3–6 of Protocol 1 $\ell = \mathrm{poly}(n)$ times on different portions of the CRS, generating ℓ independently sampled $(I, \pi_{\mathsf{hb}}, C_I, b_I, \pi_I)$. The verifier checks each of these separately, rejecting if any test fails.

Completeness and zero-knowledge of the new protocol follow immediately from the same argument as before. It remains to prove (full-fledged) soundness. As before, we have that the verifier will reject with probability $1 - \mathrm{negl}(n)$ unless $\mathsf{pk} \in \mathcal{VPK}$ and the public randomness ρ_{pk} in the CRS satisfies the statistical binding property of the POCS, so we can condition on these events. For a fixed pk, we have from the soundness of Protocol 1 that on a single iteration of steps 3–6, the verifier will reject with probability at least $1/3 - \mathrm{negl}(n)$ on $x \notin L$. Since the public key pk has polynomial size, applying a union bound over public keys, we can take $\ell = \mathrm{poly}(n)$ sufficiently large that with probability $1 - \mathrm{negl}(n)$, the verifier will reject for every public key.[12] Consequently soundness holds in the amplified protocol.

4 Instantiating with LWE

In this section we show that, assuming the hardness of LWE and the existence of a NIZK proof system for dBDD, Regev's [Reg09] LWE-based encryption scheme satisfies the conditions of Lemma 4 and therefore yields NIZK proof-systems for *all* of $\mathbf{NP}$:

Theorem 2. *Let κ be the security parameter. Let $n = n(\kappa) \in \mathbb{N}$, $q = q(\kappa) \in \mathbb{N}$, $\beta = \beta(\kappa)$, $\alpha = \alpha(\kappa) \geq 1$ and $\gamma = \gamma(\kappa) > 1$, such that $n = \mathrm{poly}(\kappa)$ and $\beta = o\left(\frac{1}{\log(\kappa)\max(\alpha,\gamma)\sqrt{n\log(q)}}\right)$. Assume that the following conditions hold:*

- *The $\mathsf{LWE}_{n,q,\beta}$ assumption holds; and*
- *There exists a NIZK proof system for $\mathsf{dBDD}_{\alpha,\gamma}$.*

Then, there exists a NIZK proof system for every language $L \in \mathbf{NP}$.

[12] The argument here resembles the standard argument for obtaining adaptively sound NIZKs from NIZKs that only have non-adaptive soundness.

Section Organization. In Section 4.1, we present Regev's [Reg09] encryption scheme. In Section 4.2, we present the NIZK proof systems for certifying public keys and plaintext values for this encryption scheme (based on the NIZK proof system for dBDD in the hypothesis of Theorem 2). In Section 4.3, we show that Regev's encryption has a POCS procedure. Finally, in Section 4.4, we use the tools developed in the prior subsections to prove Theorem 2.

4.1 Regev's Encryption Scheme

A public-key encryption scheme based on the LWE assumption was introduced in [Reg09]. We present the scheme of [Reg09], phrased as an encryption scheme with public randomness in the sense of Definition 1.

Construction 5 *Let κ be the security parameter. Let $n = n(\kappa) \in \mathbb{N}$, $q = q(\kappa) \in \mathbb{N}$, $m = 2n\log(q)$, $\beta = \beta(\kappa) \in [0,1]$ such that $n = \mathrm{poly}(\kappa)$ and $\beta = o(1/\sqrt{m})$. We define the encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with public randomness as follows:*

- **Public Randomness:** *The public randomness is a matrix $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$. We assume without loss of generality that $\lambda_1(\mathbf{A}) > q/4$* [13].
- **Key Generation** $\mathsf{Gen}(1^\kappa, \mathbf{A})$**:** *Sample $\mathbf{s} \leftarrow \mathbb{Z}_q^n \setminus \{\mathbf{0}\}$, and $\mathbf{e} \leftarrow \chi_\beta^m$, where χ_β is a discrete Gaussian with parameter β (see Definition 10). Let $\mathbf{b}^T = \mathbf{s}^T \cdot \mathbf{A} + \mathbf{e}^T$. We assume without loss of generality that $\left\|\mathbf{s}^T \cdot \mathbf{A} - \mathbf{b}^T\right\| = \left\|\mathbf{e}^T\right\| \le \ell\sqrt{m}\beta q$, where $\ell = \omega(\log(\kappa))$*[14]*. Set the public key to be $(\mathbf{A}, \mathbf{b})$ and the secret key to be $\mathbf{s}$.*
- **Encryption** $\mathsf{Enc}_{(\mathbf{A},\mathbf{b})}(\sigma)$**:** *On input a message $\sigma \in \{0,1\}$, sample $\mathbf{r} \leftarrow \{0,1\}^m$ and output $(\mathbf{c}, \omega)$, where $\mathbf{c} = \mathbf{A}\cdot\mathbf{r}$ and $\omega = \mathbf{b}^T \cdot \mathbf{r} + \sigma \cdot \left\lfloor \frac{q}{2} \right\rfloor$. We assume without loss of generality*[15] *that*

$$\left\|\mathbf{s}^T \cdot [\mathbf{A}, \mathbf{c}] - \left[\mathbf{b}, \left(\omega - \sigma \cdot \left\lfloor \frac{q}{2} \right\rfloor\right)\right]^T\right\| \le 2\ell\sqrt{m}\beta q,$$

  where $\ell = \omega(\log(\kappa))$.
- **Decryption** $\mathsf{Dec}_{\mathbf{s}}\big((\mathbf{c}, \omega)\big)$**:** *Output $\sigma = \left\lfloor \mathbf{s}^T \cdot \mathbf{c} - \omega \right\rceil_q$.*

Regev [Reg09] proved that the above scheme is semantically secure (under the LWE assumption).

Proposition 4 (c.f. [Reg09]). *Let $n = n(\kappa) \in \mathbb{N}$, $q = q(\kappa) \in \mathbb{N}$ and $\beta = \beta(\kappa) \in [0,1]$ such that $\beta = o(1/\sqrt{m})$ and $n = \mathrm{poly}(\kappa)$. If the $\mathsf{LWE}_{n,q,\beta}$ assumption holds, then Construction 5 is semantically secure.*

---

[13] From Lemma 3 this happens with overwhelming probability.

[14] Since the complementary event happens with negligible probability in κ, in case it does happen we choose the public-keys to have zero noise.

[15] Again, the complementary event happens with negligible probability, in which case we can output a ciphertext with zero noise.

In order to use the results of Section 3, we need to show that Construction 5 admits a POCS procedure. As our first step, we define a valid set of public keys. Later, we shall show NIZK proofs for the related promise problems GoodPK and GoodCT as well as a POCS procedure for Construction 5.

Fix a security parameter κ. Let $n = \text{poly}(\kappa)$, $q = q(\kappa)$, and $\beta = \beta(\kappa)$ be parameters and set $m = 2n\log(q)$. In the sequel, we omit κ from the notation to avoid cluttering. In addition, we set $\ell = \omega(\log(\kappa))$, $\mathrm{e}_{\max} = \ell\sqrt{m}\beta q$, $1 \le \alpha < \frac{q}{8\mathrm{e}_{\max}}$ and $\gamma > 1$. We assume that the following hold:

- $\beta < \frac{1}{16\ell\gamma\sqrt{m}}$;
- the $\mathsf{LWE}_{n,q,\beta}$ assumption holds; and
- there exists a NIZK proof system for $\mathsf{dBDD}_{\alpha,\gamma/4}$.

Now, we define a set (of alleged public keys) $\mathcal{VPK}$ for $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. Later we will argue that it is in fact a *valid* set of public keys as per Definition 12. Let

$$\mathcal{VPK} = \left\{ (\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m \ : \ \exists\, \mathbf{s} \in \mathbb{Z}_q^n \text{ s.t. } \left\| \mathbf{s}^T \cdot \mathbf{A} - \mathbf{b}^T \right\| \le \gamma \mathrm{e}_{\max} \right\}. \quad (2)$$

We note that the noise level allowed in Eq. (2) is *larger* by a multiplicative γ factor than the noise level that exists in honestly generated public keys.

For each $\mathsf{pk} = (\mathbf{A}, \mathbf{b}) \in \mathcal{VPK}$ and $\sigma \in \{0, 1\}$, define $C_{\mathsf{pk}}^{(\sigma)} \subseteq \mathbb{Z}_q^n \times \mathbb{Z}_q$ as:

$$C_{\mathsf{pk}}^{(\sigma)} = \left\{ (\mathbf{c}, \omega) : \exists\, \mathbf{s}' \in \mathbb{Z}_q^n \text{ s.t. } \left\| {\mathbf{s}'}^T \cdot [\mathbf{A}, \mathbf{c}] - \left[\mathbf{b}, \left(\omega - \sigma \cdot \left\lfloor \frac{q}{2} \right\rceil \right) \right]^T \right\| \le 2\gamma \mathrm{e}_{\max} \right\} \quad (3)$$

The noise level allowed in Eq. (3) is also *larger* by a multiplicative γ factor than the noise level that exists in honestly generated ciphertexts.

Remark 3. As noted in the introduction, we would like for $\mathcal{VPK}$ to contain only the honestly generated public keys and $C_{\mathsf{pk}}^{(\sigma)}$ to contain only the honestly generated encryptions of σ with respect to pk. However, introducing a gap in the definitions allows us to rely on NIZKs for suitable *approximation* problems.

We conclude this section by showing that $\mathcal{VPK}$ is indeed a valid set of public keys.

Proposition 5. *The set $\mathcal{VPK}$ is a valid set of public keys.*

Due to lack of space, we defer the proof to the full version [RSS18].

4.2 NIZKs for Validating Keys and Ciphertexts

Now that we have defined a valid set of public keys $\mathcal{VPK}$, we prove that Construction 5 satisfies the conditions of Lemma 4. To do so we assume the existence of a NIZK proof system for dBDD. Using this NIZK, we obtain NIZK proof systems for the promise problems GoodPK and GoodCT (with respect to $\mathcal{VPK}$).

Lemma 6. *Assume there exists a* NIZK *proof system for* $\mathsf{dBDD}_{\alpha,\gamma/4}$*. Then, there exists a* NIZK *proof system for the promise problem* GoodPK *(with respect to* $\mathcal{VPK}$*).*

Lemma 7. *Assume there exists a* NIZK *proof system for* $\mathsf{dBDD}_{\alpha,\gamma/4}$*. Then, there exists a* NIZK *proof system for the promise problem* GoodCT *(with respect to* $\mathcal{VPK}$*).*

We defer the proofs of Lemmas 6 and 7 to Appendices **??** and **??**.

4.3 A POCS Procedure for Regev's Scheme

The last and most challenging condition that we need is to prove that Construction 5 has a POCS procedure.

Lemma 8. *Construction 5 has a* $(1 - 4\gamma\ell\sqrt{m}\beta)$*-binding* POCS *procedure with respect to* $\mathcal{VPK}$*.*

The rest of Section 4.3 is devoted to the proof of Lemma 8.

Proof (Proof of Lemma 8).
For technical convenience and simplicity, we assume for now that $q \equiv 2 \pmod 4$. The case that $q \not\equiv 2 \pmod 4$ adds some mild complications in order to avoid introducing a small, but noticeable bias (i.e., roughly $1/q$) in the obliviously sampled bits. We describe how to extend our approach to general q in the full version [RSS18].[16]

Let us first describe the algorithms Sample and Check. The Sample algorithm takes as input a secret key $\mathsf{sk} = \mathbf{s}$ and randomness $(\boldsymbol{\rho}, \tau) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, and outputs a ciphertext.

The algorithm Sample transforms a high noise ciphertext $(\boldsymbol{\rho}, \tau)$ into a valid Regev's ciphertext under the secret key $\mathbf{s}$.

$\underline{\mathsf{Sample}\big(\mathsf{s}, (\boldsymbol{\rho}, \tau)\big)\textbf{:}}$

1. Sample $e \leftarrow \chi_{\sqrt{m}\beta}$. Let $\omega_0 = \mathbf{s}^T \cdot \boldsymbol{\rho} + e$ and $\omega_1 = \omega_0 + \left\lfloor \frac{q}{2} \right\rfloor$.
2. If $|\tau - \omega_0| < |\tau - \omega_1|$, set $\sigma = 0$. Otherwise, set $\sigma = 1$.
3. Output $(\boldsymbol{\rho}, \omega_\sigma)$, which is a valid ciphertext for the message σ.

The Check algorithm takes as input a public key $\mathsf{pk} = (\mathbf{A}, \mathbf{b})$, randomness $(\boldsymbol{\rho}, \tau) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, and an alleged ciphertext $(\boldsymbol{\rho}', \omega') \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, and outputs a single bit denoting acceptance or rejection.

$\underline{\mathsf{Check}\big(\mathsf{pk}, (\boldsymbol{\rho}, \tau), (\boldsymbol{\rho}', \omega')\big)\textbf{:}}$

If $\boldsymbol{\rho}' = \boldsymbol{\rho}$ and $|\omega' - \tau| \le \frac{q}{4}$, accept. Otherwise, reject.

[16] Alternatively, we could reduce the bias to be negligible using Von Neumann's trick [VN61] for transforming a biased source to an almost unbiased source.

Finally, we describe the $\mathsf{EncryptAndExplain}$ algorithm, which takes as input a public key $\mathsf{pk} = (\mathbf{A}, \mathbf{b})$ and a message $\sigma \in \{0, 1\}$ and produces randomness and a ciphertext that are close to the distribution induced by Sample.

$\mathsf{EncryptAndExplain}\big((\mathbf{A}, \mathbf{b}), \sigma\big)$*:*

1. Sample $\mathbf{r} \leftarrow \{0,1\}^m$. Compute $\boldsymbol{\rho}' = \mathbf{A} \cdot \mathbf{r}$ and $\omega' = \mathbf{b}^T \cdot \mathbf{r} + \sigma \cdot \lfloor \frac{q}{2} \rfloor$. Note that $(\boldsymbol{\rho}', \omega')$ is a fresh encryption of σ.
2. Sample $\tau' \leftarrow \mathbb{Z}_q$ subject to $|\tau' - \omega'| < \frac{q}{4}$.
3. Output $\big((\boldsymbol{\rho}', \tau'), (\boldsymbol{\rho}', \omega')\big)$.

We now show that these algorithms satisfy each of the conditions of Definition 13.

Complete. Let $(\boldsymbol{\rho}, \tau) \leftarrow \mathbb{Z}_q^n \times \mathbb{Z}_q$ and $(\boldsymbol{\rho}', \omega') \leftarrow \mathsf{Sample}(\mathbf{s}, (\boldsymbol{\rho}, \tau))$. By construction $\boldsymbol{\rho}' = \boldsymbol{\rho}$ and $|\tau - \omega'| \leq \frac{q}{4}$, and so Check always accepts.

Unbiased. We defer the proof that this scheme is unbiased to the full version [RSS18].

Statistically Binding. Let $\mathsf{pk} = (\mathbf{A}, \mathbf{b}) \in \mathcal{VPK}$ with public randomness $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$. By construction $\lambda_1(\mathbf{A}) > q/4$, so there exists a unique $\mathbf{s}$ such that $\left\|\mathbf{s}^T \cdot \mathbf{A} - \mathbf{b}^T\right\| \leq \gamma e_{\max}$. We assume that the above holds for $\mathbf{A}$.

Therefore, it holds that:

$$C_{\mathsf{pk}}^{(\sigma)} = \left\{ (\mathbf{c}, \omega) \in \mathbb{Z}_q^n \times Z_q \ : \ \left\| \mathbf{s}^T \cdot [\mathbf{A}, \mathbf{c}] - \left[\mathbf{b}, \left(\omega - \sigma \cdot \left\lfloor \frac{q}{2} \right\rfloor\right)\right]^T \right\| \leq 2\gamma e_{\max} \right\}.$$

We remark that in this case, $(\mathbf{c}, \omega) \in C_{\mathsf{pk}}^{(0)}$ if and only if $\left(\mathbf{c}, \omega + \lfloor \frac{q}{2} \rfloor\right) \in C_{\mathsf{pk}}^{(1)}$. Furthermore,

$$\begin{aligned}
&\Pr_{\boldsymbol{\rho},\tau}\left[\exists\, (\mathbf{c}_0, \omega_0) \in C_{\mathsf{pk}}^{(0)}, \exists\, (\mathbf{c}_1, \omega_1) \in C_{\mathsf{pk}}^{(1)} \text{ s.t. } \begin{matrix} \mathsf{Check}(\mathsf{pk}, (\boldsymbol{\rho}, \tau), (\mathbf{c}_0, \omega_0)) = 1, \\ \mathsf{Check}(\mathsf{pk}, (\boldsymbol{\rho}, \tau), (\mathbf{c}_1, \omega_1)) = 1 \end{matrix}\right] \\
&= \Pr_{\boldsymbol{\rho},\tau}\left[\exists\, \omega_0, \exists\, \omega_1 \in \mathbb{Z}_q \text{ s.t. } \begin{matrix} \left|\mathbf{s}^T \cdot \boldsymbol{\rho} - \omega_0\right| \leq \gamma e_{\max}, \\ \left|\mathbf{s}^T \cdot \boldsymbol{\rho} - \omega_1 - \lfloor \frac{q}{2} \rfloor\right| \leq \gamma e_{\max}, \\ |\omega_0 - \tau| \leq q/4, \\ |\omega_1 - \tau| \leq q/4 \end{matrix}\right] \\
&\leq \Pr_{\boldsymbol{\rho},\tau}\left[\left(\left|\mathbf{s}^T \cdot \boldsymbol{\rho} - \tau\right| \leq \gamma e_{\max} + \frac{q}{4}\right) \text{ and } \left(\left|\mathbf{s}^T \cdot \boldsymbol{\rho} - \left(\tau + \left\lfloor \frac{q}{2} \right\rfloor\right)\right| \leq \gamma e_{\max} + \frac{q}{4}\right)\right] \\
&\leq \Pr_{r}\left[\left(|r| \leq \gamma e_{\max} + \frac{q}{4}\right) \text{ and } \left(\left|r + \left\lfloor \frac{q}{2} \right\rfloor\right| \leq \gamma e_{\max} + \frac{q}{4}\right)\right] \\
&\leq \Pr_{r}\left[r \in \left[\frac{q}{4} - \gamma e_{\max}, \frac{q}{4} + \gamma e_{\max}\right] \cup \left[-\frac{q}{4} - \gamma e_{\max}, -\frac{q}{4} + \gamma e_{\max}\right]\right] \\
&\leq 4\gamma\ell\sqrt{m}\beta.
\end{aligned}$$

The first equality follows from the definition of $C_{\mathsf{pk}}^{(0)}$ and $C_{\mathsf{pk}}^{(1)}$ and the description of Check. More specifically, the conditions $\left|\mathbf{s}^T \cdot \boldsymbol{\rho} - \omega_0\right| \leq \gamma \mathrm{e}_{\max}$ and $\left|\mathbf{s}^T \cdot \boldsymbol{\rho} - \omega_1 - \left\lfloor \frac{q}{2} \right\rfloor\right| \leq \gamma \mathrm{e}_{\max}$ follow from the fact that $(\mathbf{c}_0, \omega_0) \in C_{\mathsf{pk}}^{(0)}$ and $(\mathbf{c}_1, \omega_1) \in C_{\mathsf{pk}}^{(1)}$, respectively. The conditions $|\omega_0 - \tau| \leq q/4$ and $|\omega_1 - \tau| \leq q/4$ follow from $\mathsf{Check}(\mathsf{pk}, (\boldsymbol{\rho}, \tau), (\mathbf{c}_0, \omega_0)) = 1$ and $\mathsf{Check}(\mathsf{pk}, (\boldsymbol{\rho}, \tau), (\mathbf{c}_1, \omega_1)) = 1$ respectively. The next inequality follows from the triangle inequality. Next, we replace $\mathbf{s}^T \cdot \boldsymbol{\rho} - \tau$ by a uniformly random element r of $\mathbb{Z}_q$. Then, we note that r has to belong to a set of size at most $4\gamma \mathrm{e}_{\max} \leq 4\gamma\ell\sqrt{m}\beta q$, which happens with probability at most $4\gamma\ell\sqrt{m}\beta$. The last inequality then follows.

Simulatable. Let $N = \mathrm{poly}(\kappa)$. Sample $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ and $(\mathsf{pk}, \mathsf{sk}) = \big((\mathbf{A}, \mathbf{b}), \mathbf{s}\big) \leftarrow \mathsf{Gen}(1^\kappa, \mathbf{A})$ and consider the following two experiments:

– For $i \in [N]$, let $(\boldsymbol{\rho}_i, \tau_i) \leftarrow \mathbb{Z}_q^n \times \mathbb{Z}_q$, $(\boldsymbol{\rho}_i, \omega_i) \leftarrow \mathsf{Sample}(\mathbf{s}, (\boldsymbol{\rho}_i, \tau_i))$, $\sigma_i = \mathsf{Dec}_{\mathbf{s}}((\boldsymbol{\rho}_i, \omega_i))$. Output $\big(\mathsf{pk}, (\boldsymbol{\rho}_i, \tau_i, \omega_i, \sigma_i)_{i \in [N]}\big)$.
– For $i \in [N]$, let $\sigma'_i \in_R \{0,1\}$. Set $\big((\boldsymbol{\rho}'_i, \tau'_i), (\boldsymbol{\rho}'_i, \omega'_i)\big) \leftarrow \mathsf{EncryptAndExplain}(\mathsf{pk}, \sigma'_i)$. Output $\big(\mathsf{pk}, (\boldsymbol{\rho}'_i, \tau'_i, \omega'_i, \sigma'_i)_{i \in [N]}\big)$.

In the full version [RSS18] we show that the outputs of these two experiments are computationally indistinguishable.

Computationally Hiding. Given public key $\mathsf{pk} = (\mathbf{A}, \mathbf{b})$ and randomness $(\boldsymbol{\rho}, \tau)$, the procedure Sample simply computes a fresh encryption $(\boldsymbol{\rho}, \omega)$ using the secret-key variant of Regev's scheme. Let $\sigma = \mathsf{Dec}_{\mathbf{s}}((\boldsymbol{\rho}, \omega))$. Then similarly to the above proof

$$\big(\mathsf{pk}, \boldsymbol{\rho}, \tau, \omega, \sigma\big) \equiv \big(\mathsf{pk}, \boldsymbol{\rho}, \tau', \omega', \sigma\big)$$

where $\omega' = \mathbf{s}^T \cdot \boldsymbol{\rho} + \sigma \cdot \left\lfloor \frac{q}{2} \right\rfloor + e$, with $e \leftarrow \chi_{\sqrt{m}\beta}$ and τ' sampled uniformly such that $|\tau' - \omega'| < q/4$.

Then, since τ' is a randomized function of ω', the computational hiding property of the POCS follows immediately from the semantic security of Regev's encryption scheme.

This concludes the proof of Lemma 8 for $q \equiv 2 \pmod 4$. We describe how to extend the proof to general q in the full version [RSS18]. The main difficulty is to sample the boundary points with the correct probability.

4.4 Putting it All Together (Proof of Theorem 2)

We now complete the proof of Theorem 2. We have shown that all of the conditions of Lemma 4 hold, as follows.

1. By Proposition 5, Construction 5 has a valid set of public keys $\mathcal{VPK}$.
2. By Lemma 8, Construction 5 has a POCS with respect to $\mathcal{VPK}$.
3. By Lemma 6, there is a NIZK for GoodPK.
4. By Lemma 7, there is a NIZK for GoodCT.

Theorem 2 then follows immediately by Lemma 4.

References

ACPS09. Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *CRYPTO*, 2009.

AFV11. Shweta Agrawal, David Mandell Freeman, and Vinod Vaikuntanathan. Functional encryption for inner product predicates from learning with errors. In *ASIACRYPT*, 2011.

APSD17. Navid Alamati, Chris Peikert, and Noah Stephens-Davidowitz. New (and old) proof systems for lattice problems. Cryptology ePrint Archive, Report 2017/1226, 2017.

BDSMP91. Manuel Blum, Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Noninteractive zero-knowledge. *SIAM Journal on Computing*, 20(6):1084–1118, 1991.

BFM88. Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *STOC*, 1988.

BKM06. Adam Bender, Jonathan Katz, and Ruggero Morselli. Ring signatures: Stronger definitions, and constructions without random oracles. In *TCC*. Springer, 2006.

BMW03. Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *Eurocrypt*, 2003.

BP15. Nir Bitansky and Omer Paneth. Zaps and non-interactive witness indistinguishability from indistinguishability obfuscation. In *TCC*, 2015.

BR93. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *CCS*, 1993.

BRV17. Itay Berman, Ron D. Rothblum, and Vinod Vaikuntanathan. Zero-knowledge proofs of proximity. *IACR Cryptology ePrint Archive*, 2017:114, 2017.

BV14. Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) \$\mathsf{LWE}\$. *SIAM J. Comput.*, 43(2):831–871, 2014.

BY96. Mihir Bellare and Moti Yung. Certifying permutations: Noninteractive zero-knowledge based on any trapdoor permutation. *J. Cryptology*, 9(3):149–166, 1996.

CCRR18. Ran Canetti, Yilei Chen, Leonid Reyzin, and Ron D. Rothblum. Fiat-shamir and correlation intractability from strong kdm-secure encryption. Cryptology ePrint Archive, Report 2018/131, 2018.

CGH04. Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, 2004.

CL17. Ran Canetti and Amit Lichtenberg. Certifying trapdoor permutations, revisited. *IACR Cryptology ePrint Archive*, 2017:631, 2017.

DDN03. Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM Review*, 45(4):727–784, 2003.

DN07. Cynthia Dwork and Moni Naor. Zaps and their applications. *SIAM J. Comput.*, 36(6):1513–1543, 2007.

dPL17. Rafaël del Pino and Vadim Lyubashevsky. Amortization with fewer equations for proving knowledge of small secrets. In *CRYPTO*, 2017.

FLS99. Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple noninteractive zero knowledge proofs under general assumptions. *SIAM J. Comput.*, 29(1):1–28, 1999.

FS86. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, 1986.

GG00. Oded Goldreich and Shafi Goldwasser. On the limits of nonapproximability of lattice problems. *J. Comput. Syst. Sci.*, 60(3):540–563, 2000.

GK03. Shafi Goldwasser and Yael Tauman Kalai. On the (in)security of the fiat-shamir paradigm. In *FOCS*, 2003.

GK05. Shafi Goldwasser and Dmitriy Kharchenko. Proof of plaintext knowledge for the ajtai-dwork cryptosystem. In *TCC*, 2005.

GKM+00. Yael Gertner, Sampath Kannan, Tal Malkin, Omer Reingold, and Mahesh Viswanathan. The relationship between public key encryption and oblivious transfer. In *FOCS*, 2000.

GKP+13. Shafi Goldwasser, Yael Kalai, Raluca Ada Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. Reusable garbled circuits and succinct functional encryption. In *STOC*, 2013.

GKW17. Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. *IACR Cryptology ePrint Archive*, 2017:274, 2017.

GM84. Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

Gol01. Oded Goldreich. *The Foundations of Cryptography - Volume 1, Basic Techniques*. Cambridge University Press, 2001.

Gol04. Oded Goldreich. *The Foundations of Cryptography - Volume 2, Basic Applications*. Cambridge University Press, 2004.

Gol11. Oded Goldreich. Basing non-interactive zero-knowledge on (enhanced) trapdoor permutations: The state of the art. In *Studies in Complexity and Cryptography*. 2011.

GOS12. Jens Groth, Rafail Ostrovsky, and Amit Sahai. New techniques for non-interactive zero-knowledge. *J. ACM*, 59(3):11:1–11:35, 2012.

GR13. Oded Goldreich and Ron D. Rothblum. Enhancements of trapdoor permutations. *J. Cryptology*, 26(3):484–512, 2013.

Gro10. Jens Groth. Short pairing-based non-interactive zero-knowledge arguments. In *ASIACRYPT*, 2010.

GS08. Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In *EUROCRYPT*, 2008.

GVW15. Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from lwe. In *CRYPTO*. Springer, 2015.

KRR17. Yael Tauman Kalai, Guy N. Rothblum, and Ron D. Rothblum. From obfuscation to the security of fiat-shamir for proofs. In *CRYPTO*, 2017.

KTX08. Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In *ASIACRYPT*, 2008.

KW18. Sam Kim and David J. Wu. Multi-theorem preprocessing nizks from lattices. *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II*, pages 733–765, 2018.

LLM+16. Benoît Libert, San Ling, Fabrice Mouhartem, Khoa Nguyen, and Huaxiong Wang. Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. In *ASIACRYPT*, 2016.

LM09. Vadim Lyubashevsky and Daniele Micciancio. On bounded distance decoding, unique shortest vectors, and the minimum distance problem. In *CRYPTO*, 2009.

LNSW13. San Ling, Khoa Nguyen, Damien Stehlé, and Huaxiong Wang. Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In *PKC*, 2013.

Lyu08. Vadim Lyubashevsky. Lattice-based identification schemes secure under active attacks. In *PKC*, 2008.

MV03. Daniele Micciancio and Salil Vadhan. Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. *CRYPTO*, 2003.

MW16. Pratyay Mukherjee and Daniel Wichs. Two round multiparty computation via multi-key FHE. In *EUROCRYPT*, 2016.

Nao03. Moni Naor. On cryptographic assumptions and challenges. In *CRYPTO*, 2003.

NY90. Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC*, 1990.

Pei09. Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *STOC*, 2009.

PV08. Chris Peikert and Vinod Vaikuntanathan. Noninteractive statistical zero-knowledge proofs for lattice problems. In *CRYPTO*, 2008.

PVW08. Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In *CRYPTO*, 2008.

PW08. Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In *STOC*, 2008.

Rab79. M. O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical report, Cambridge, MA, USA, 1979.

Reg09. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):34:1–34:40, 2009.

RSS18. Ron D. Rothblum, Adam Sealfon, and Katerina Sotiraki. Towards non-interactive zero-knowledge for NP from LWE. *IACR Cryptology ePrint Archive*, 2018:240, 2018.

Sah99. Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *FOCS*, 1999.

Sho99. Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, 41(2):303–332, 1999.

Ste96. Jacques Stern. A new paradigm for public key identification. *IEEE Trans. Information Theory*, 42(6):1757–1768, 1996.

SW14. Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In *STOC*, 2014.

Vad99. Salil P. Vadhan. *A Study of Statistical Zero-Knowledge Proofs*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1999.

VN61. J Von Neumann. Various techniques used in connection with random digits, paper no. 13 in Monte Carlo method. *NBS Applied Mathematics Series*, (12), 1961.

WZ17. Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under LWE. *IACR Cryptology ePrint Archive*, 2017:276, 2017.