NIZK from LPN and Trapdoor Hash via Correlation Intractability for Approximable Relations

Zvika Brakerski* Venkata Koppula* Tamer Mour*

Abstract

We present new non-interactive zero-knowledge argument systems (NIZK), based on standard assumptions that were previously not known to imply it. In particular, we rely on the hardness of both the learning parity with noise (LPN) assumption, and the existence of *trapdoor hash functions* (TDH, defined by Döttling et al., Crypto 2019). Such TDH can be based on a number of standard assumptions, including DDH, QR, DCR, and LWE.

We revisit the correlation intractability (CI) framework for converting Σ-protocols into NIZK, and present a different strategy for instantiating it by putting together two new components. First, while prior works considered the search-complexity of the relations for which CI is sought, we consider their *probabilistic representation*. Namely, a distribution over lower-complexity functions that bitwise-computes the target function with all but small (constant) probability. The second component is a new perspective for quantifying the class of relations for which CI is achieved. We show that it is instructive to consider *CI for approximable relations* (CI-Apx) which is quantified by a class of relations, but requires CI to hold against *any approximation* of any relation in this class.

We show that CI-Apx for just constant-degree polynomials suffices for NIZK, if the underlying Σ-protocol is implemented using a suitable commitment scheme. We show that such a commitment scheme can be constructed based on LPN. We then show how to construct CI-Apx for constant-degree polynomials from any suitable TDH (with an enhanced correctness property that is satisfied by all existing TDH constructions).

1 Introduction

Zero-Knowledge (ZK) [GMR85] is one of the most celebrated and widely used notions in modern cryptography. A ZK proof is a protocol in which a prover conveys the validity of a statement to a verifier in a way that reveals no additional information. In a non-interactive ZK proof system (NIZK), we wish to construct a singe-message ZK proof system. Common setup is necessary for NIZK, and by default (and always in this work) NIZK is considered in the common random/reference string (CRS) model. In the CRS model, a trusted string is sampled from a prescribed distribution (preferably uniform) and made available to both the prover and the verifier. Ideally, we would have liked to construct a NIZK proof system for all NP languages (or equivalently for some NP-complete language).[1] NIZK for NP turns out to be extremely useful for many applications such as CCA

*Weizmann Institute of Science, {zvika.brakerski, venkata.koppula, tamer.mour}@weizmann.ac.il. Supported by the Binational Science Foundation (Grant No. 2016726), and by the European Union Horizon 2020 Research and Innovation Program via ERC Project REACT (Grant 756482) and via Project PROMETHEUS (Grant 780701).

[1] In this work we only consider ZK/NIZK proof systems where the honest prover is computationally efficient given a witness to the NP language.

security [NY90, DDN91], signatures [BMW03, BKM06], and numerous other applications, including recent applications in the regime of cryptocurrencies [BCG+14]. From this point and on, we use the term NIZK to refer to "NIZK for NP" unless otherwise stated.

While ZK proofs for all NP languages are known under the minimal assumption that one-way functions exist, this is far from being the case for NIZK. We focus our attention on constructions in the standard model and under standard cryptographic assumptions. For many years, NIZK under standard assumptions were only known based on Factoring [BFM88] (or doubly enhanced trapdoor functions, which are only known to exist based on Factoring [FLS99]) or assumptions on groups with bilinear maps [GOS12].

More recently, constructions based on indistinguishability obfuscation were presented as well [SW14]. Most recently, a new line of works, starting with [KRR17, CCRR18, HL18], focused on obtaining NIZK based on the notion of *correlation intractability* (CI) [CGH04]. In the CI framework, it was shown that in order to construct NIZK, it suffices to construct a family of hash functions $\mathcal{H}$ with the following property. For every efficient f, given a hash function $H \leftarrow \mathcal{H}$ from the family, it is computationally hard to find x s.t. $f(x) = H(x)$. If such correlation intractable hash (CIH) is constructed, then it can be used to securely instantiate the Fiat-Shamir paradigm [FS87] and derive NIZK from so-called Σ-protocols. This line of works culminated in two remarkable achievements. Canetti et al. [CCH+19] constructed NIZK based on the existence of circular secure fully homomorphic encryption. Peikert and Shiehian [PS19] constructed NIZK based on the hardness of the learning with errors (LWE) problem.[2]

These recent results opened a new avenue in the study of NIZK and raised hope that construction under additional assumptions can be presented. However, it appears that there is an inherent barrier to expanding known techniques beyond LWE-related assumptions. The current approaches for constructing CI hash from standard assumptions use the notion of *somewhere statistical CI*, in which, for any f, it is possible to sample from a distribution $\mathcal{H}_f$ which is indistinguishable from the real $\mathcal{H}$, and for which the CI game is *statistically* hard to win. Roughly speaking, this is achieved, in known constructions [CCH+19, PS19] by making $\mathcal{H}_f$ perform some homomorphic evaluation of f on the input x. Thus, it appears that homomorphic evaluation of complex functions f is essential to apply these tools.

The starting point of our work is the observation that, under the learning parity with noise (LPN) assumption, we can reduce the complexity of functions for which achieving CIH implies NIZK down to functions with *probabilistic constant-degree representation*. That is, ones that can be approximated by a distribution on constant-degree polynomials.

We substantiate the usefulness of this approach by identifying a general connection between correlation intractability for a function class $\mathcal{F}$, which has probabilistic representation by a class $\mathcal{C}$ (potentially of lower complexity), and CI for relations that are *approximable* by $\mathcal{C}$.

Correlation Intractability for relations approximable by $\mathcal{C}$ (denoted "CI-Apx for $\mathcal{C}$") is a stronger notion than the one studied in prior works, namely CI for relations searchable by $\mathcal{C}$. In CI-Apx, we require that for all $C \in \mathcal{C}$ it is hard not only to find x such that $C(x) = \mathcal{H}(x)$ but, rather, that it is hard to find an x such that $\mathcal{H}(x)$ and $C(x)$ are *close* in Hamming distance.[3] When the probabilistic representation $\mathcal{C}$ of our target class $\mathcal{F}$ is sufficiently simple, e.g. constant-degree

[2] To be more accurate, [PS19] showed how to construct a CI hash function for size s circuits, for any parameter s. This is slightly weaker than a single $\mathcal{H}$ for all functions, but it suffices in order to instantiate the framework.

[3] Note that even non-searchable relations can potentially be approximable by a class of functions. Thus via the notion CI-Apx we can extend our capabilities for constructing CIH even beyond searchable relations. This is not of direct use in this work, but may be useful for future works.

polynomials, then the reduction from CI for $\mathcal{F}$ to CI-Apx for $\mathcal{C}$ opens the possibility for new constructions of CIH from standard assumptions. Specifically from assumptions that are not known to apply fully-homomorphic encryption or similarly strong primitives.

In particular, we show that CI-Apx for a function class $\mathcal{C}$ can be constructed based on a *rate-1 trapdoor hash* scheme for $\mathcal{C}$. Trapdoor hash (TDH) is a fairly new cryptographic primitive which was recently introduced by Döttling et al. [DGI+19]. They also constructed rate-1 TDH for constant-degree polynomials from a number of standard assumptions, including DDH, QR, and DCR (which are not known to imply fully-homomorphic encryption) and also LWE. Consequently, we obtain CI-Apx for constant-degree polynomials from such assumptions and, therefore, CI for any class of functions with probabilistic constant-degree representation. We note that we require a slightly stronger correctness property from TDH, compared to the definition provided in [DGI+19], but it is satisfied by all known constructions.

Consequences. We get non-interactive (computational) zero knowledge argument systems for NP, in the common random string model, based on the existence of any rate-1 trapdoor hash for constant degree and further assuming LPN. We stress that we can generically abstract the LPN requirement as a requirement for an extractable commitment scheme with very low-complexity approximate-extraction. By instantiating our construction using the rate-1 TDH from [DGI+19], we get, in particular, the first NIZK from LPN and DDH.

Open Questions. The main open question we leave unanswered is whether it is possible to minimize the required assumptions for constructing NIZK using CI-Apx. One may approach this problem either by constructing CI-Apx for constant degree functions based on the LPN assumption, or by further extending the CI-Apx framework to allow a more general utilization for NIZKs, possibly depending on assumptions already implying CI-Apx.

Another open question is whether we can obtain stronger notions of NIZKs, in particular NIZK proofs or NISZK, from a similar set of standard assumptions. To achieve statistical ZK using our approach simply requires the underlying commitment (with low-degree extraction) to be lossy. Getting statistically sound proof systems via CI-Apx, however, seems to be inherently more difficult, as it requires the resulting CI to be "somewhere statistical" for the approximated class of functions.

Lastly, the new constructions of ZAPs [LVW19, BFJ+19, JJ19] rely on the CI framework but, unfortunately, we do not know how to extend them since the notion of commitment that is required for the ZAPs is not known to be constructible from LPN (or other assumptions with very low complexity extraction). At a high level, these works requires the public parameters of the commitment scheme to be statistically close to uniform (and this seems hard to achieve with our LPN noise regime).

1.1 Overview of Our Techniques and Results

Our construction of NIZK instantiates the general Correlation Intractability (CI) framework. The approach followed in prior work for constructing CI hash, for relations searchable by a function class $\mathcal{F}$, considers the straight-forward representation of $\mathcal{F}$ as a class of circuits. In this work, we take a different angle, and tackle the CI problem for relations searchable by $\mathcal{F}$ through its probabilistic representation by a much simpler class $\mathcal{C}$. Such an approach allows us to obtain CI hash for classes of relations that are sufficiently rich to imply NIZK, while avoiding the use of FHE or similar heavy machinery.

1.1.1 NIZK from Correlation Intractability

Our starting point for constructing NIZK is similar to the approach in previous works of applying Fiat-Shamir on ZK protocols, in a provably-sound manner, using CI hash. We start with a public-coin trapdoor Σ-protocol that follows the natural "commit-then-open" paradigm, where the prover first sends a set of commitments, then, upon receiving the verifier's challenge bit $e \in \{0,1\}$, he replies by opening some of the commitments. Lastly, the verifier checks that the openings are valid, and then performs an additional check over the opened values. An example of such a protocol is the ZK protocol for Hamiltonicity from [Blu87, FLS99].

An important property of commit-then-open trapdoor-Σ protocols is the *unique bad challenge property*: for any instance x not in the language, if (a, e, z) is an accepting transcript, then e is uniquely determined by the first message a. This connection is characterized by a function denoted by $\mathsf{BadChallenge} : a \mapsto e$. In the CI paradigm, we apply Fiat-Shamir over sufficiently many repetitions of such a protocol, using a CI hash for the relation searchable by $\mathsf{BadChallenge}$, which is defined as follows. A vector of first messages $\mathbf{a}$ is in a relation with a vector of verifier's challenges $\mathbf{e}$ if on each coordinate, the corresponding $\mathbf{e}$ entry is the unique bad challenge of that coordinate in $\mathbf{a}$. If a cheating prover P^* succeeds in breaking the soundness of the protocol, then he must have found a $\mathsf{BadChallenge}$ correlation, i.e. vectors $(\mathbf{a}, \mathbf{e})$ in the relation, implying an attack against the CI of the underlying hash family.

Prior work considered protocols where the bad challenge is efficiently computable and, consequently, focused on constructing CI for all efficiently searchable relations. These contain, in particular, the relations efficiently searchable by $\mathsf{BadChallenge}$. We deviate from this approach. We observe that $\mathsf{BadChallenge}$ can be approximated by a distribution over constant-degree polynomials when instantiating this template with an appropriate commitment scheme. This reduces our CI task to achieving CIH for functions with constant-degree *probabilistic representation*. Such CIH is implied by a special notion of correlation intractability against constant-degree functions – CI for *approximable relations*, or CI-Apx for short. Details follow.

1.1.2 Probabilistic Representation, Approximable Relations and CI

Assume that a class of functions $\mathcal{F} : \{0,1\}^n \to \{0,1\}^m$ has a *probabilistic representation* by some simpler class of functions $\mathcal{C}$. Namely, for any $f \in \mathcal{F}$, there exists a distribution $\mathfrak{C}_f$ over $\mathcal{C}$ such that $\Pr[\Delta(C(x), f(x)) \leq \epsilon m] > 1 - \mathsf{negl}(\lambda)$ for any x and a random $C \xleftarrow{\$} \mathfrak{C}_f$.

Let $\mathcal{H} : \{0,1\}^n \to \{0,1\}^m$ be a hash family. An adversary $\mathcal{A}$ that is able to find a correlation $\mathcal{H}(x) = f(x)$ for some f is able to find, with overwhelming probability over a random $C \leftarrow \mathfrak{C}_f$, an "approximate correlation" $\Delta(\mathcal{H}(x), C(x)) \leq \epsilon m$ for some small ϵ. It follows therefore that by considering probabilistic representation, we can identify a connection between correlation intractability against f and correlation intractability against any relation that is *approximable* (or approximately searchable) by some function $C \in \mathfrak{C}_f$. We denote this class of relations

$$R_C^\epsilon = \{(x, y) \in \{0,1\}^n \times \{0,1\}^m \mid \Delta(y, C(x)) \leq \epsilon m\} \ .$$

More formally, an adversary that breaks the CI of $\mathcal{H}$ for a relation searchable by f must be able to break the CI of the same hash $\mathcal{H}$ for the relation $\mathcal{R}_C^\epsilon$ defined by some $C \in \mathfrak{C}_f$. Hence, CI-Apx for $\mathcal{C}$ (i.e. CI for all relations $\mathcal{R}_C^\epsilon$) implies CI for $\mathcal{F}$.

Theorem 1.1 (CI through Probabilistic Representation, Informal)**.** *Let $\mathcal{F}$ be a class of functions with probabilistic representation by $\mathcal{C}$. Then, any CI-Apx hash family for $\mathcal{C}$ is a CI hash for $\mathcal{F}$.*

1.1.3 Probabilistic Constant-Degree Representation of the Bad Challenge Function

Recall that in a commit-then-open trapdoor Σ-protocol, the verification is either performed over a subset of commitment openings corresponding to $e = 0$ or a subset of openings corresponding to $e = 1$. From the unique bad challenge property, it is impossible that the verification on both subsets succeed if $x \notin L$. Thus, the $\mathsf{BadChallenge}$ function can be computed in two steps: an extraction step, to extract the messages underlying the commitments of one of the aforementioned subsets, say the one corresponding to $e = 1$, followed by an efficient verification (for $e = 1$) over the extracted values. If the verification accepts, then the bad challenge must be $e = 1$ and, otherwise, the bad challenge is either $e = 0$ or does not exist (in which case a is not in the relation and the output may be arbitrary). Hence, we can split the task of probabilistically representing $\mathsf{BadChallenge}$ to two sub-tasks: extraction and post-extraction verification.

Post-Extraction Verification as a 3-CNF. The post-extraction verification is an arbitrary polynomial computation and, generally, may not have probabilistic constant-degree representation as is. The first step towards a constant-degree approximation of $\mathsf{BadChallenge}$ is observing that, by relying on the Cook-Levin approach for expressing the verification procedure as a 3-CNF satisfiability problem, we may reduce the complexity of the verification to 3-CNF as follows. Let Φ_e denote the 3-CNF formula that captures the verification corresponding to challenge e; that is, Φ_e has a satisfying witness w_e if and only if the verifier accepts the prover's second message for challenge bit e. The prover can compute w_e efficiently (using the Cook-Levin approach, this witness simply consists of all intermediate steps of the verification). Therefore, we let the prover also include commitments to w_0, w_1 in his first message. When the verifier sends challenge e, the prover also provides openings for w_e, and the verifier checks decommitments then evaluates Φ_e. By transforming the protocol as described, the bad challenge computation now consists, as before, of extraction, then an evaluation of the 3-CNF formula Φ_1, rather than an arbitrary poly-time verification.

We can then use standard well-known randomization techniques to probabilistically approximate any 3-CNF formula by constant-degree polynomials (see Lemma 3.13).

Extraction via a Randomized Linear Function. For the extraction step, we show that by adapting the LPN-based PKE scheme of Damgård and Park [DP12] (which is closely related to the PKE scheme by Alekhnovich [Ale03]) we can construct an extractable commitment scheme whose extraction algorithm can be probabilistically represented by a linear function. The secret extraction key is a matrix $\mathbf{S}$, and the public key consists of a matrix $\mathbf{A}$ together with $\mathbf{B} = \mathbf{A} \cdot \mathbf{S} + \mathbf{E}$. Here, $\mathbf{E}$ is chosen from a noise distribution with suitably low noise rate. To compute a commitment for bit x, the Commit algorithm chooses a low Hamming weight vector $\mathbf{r}$, and outputs $\mathbf{u} = \mathbf{rA}$ and $\mathbf{c} = \mathbf{rB} + x^\ell$. The opening for the commitment is the randomness $\mathbf{r}$, and the verification algorithm simply checks that $\mathbf{r}$ has low Hamming weight, and that the Commit algorithm, using $\mathbf{r}$, outputs the correct commitment. Finally, note that using $\mathbf{S}$, one can extract the message underlying a commitment $(\mathbf{u}, \mathbf{c})$: simply compute $\mathbf{uS} + \mathbf{c} = x^\ell + \mathbf{rE}$. By carefully setting the LPN-parameters, we ensure that if $(\mathbf{u}, \mathbf{c})$ is a valid commitment (i.e. can be opened with some x and $\mathbf{r}$), then $\mathbf{rE}$ has sufficiently low Hamming weight. Therefore, by sampling a random column $\mathbf{s}$ in $\mathbf{S}$, we get that $\mathbf{us} + \mathbf{c} = x$ with sufficiently high probability.

The Case of Invalid Commitments. We have shown that, using a distribution over linear functions, we can approximate extraction of valid commitments. A cheating prover, however, may

chose to send invalid commitments. We claim that, in such a case, we may allow the probabilistic representation to behave arbitrarily.

Fix some $x \notin L$ and a first message a. If there exist no bad challenge for a or if the (unique) bad challenge is $e = 1$, then all commitments in a corresponding to inputs of Φ_1 must be valid (since the prover is able to open them in a way that is accepted by the verifier). Thus, we potentially have a problem only in the case where $e = 0$ is the bad challenge, i.e. the commitments of input bits to Φ_0 are valid and $\Phi_0(w_0) = 1$ on their respective openings w_0. Our concern is that since our bad challenge function only looks at the Φ_1 locations, which may be arbitrary invalid commitments, we have no guarantee on the extraction, and therefore our bad challenge function will output $e = 1$ even though the unique bad challenge is $e = 0$. We show that this is not possible.

Let w_1' be the arbitrary value computed by the approximate extraction algorithm on the possibly invalid commitments in the locations of the Φ_1 inputs. We will see that it still must be the case that $\Phi_1(w_1') = 0$ and therefore the bad challenge function outputs $e = 0$ as required. The reason is that otherwise we can put together *valid commitments of both* w_0 and w_1', so as to create a first message a' which refutes the soundness of the original Σ-protocol, since it can be successfully opened both for $e = 0$ and for $e = 1$.

1.1.4 Constructing CI for Approximable Relations

The main idea behind recent constructions of CI for relations searchable by some function class $\mathcal{C}$ [CCH+19, PS19] is to construct a *somewhere statistical* CI hash family $\mathcal{H}$. That is, one where there exists, for any $C \in \mathcal{C}$, a distribution of hash functions $\mathcal{H}_C$ that are indistinguishable from the real $\mathcal{H}$, and are statistically CI for that specific C. Namely, for any C, there exists no x such that $\mathcal{H}_C(x) = C(x)$ or, equivalently, the image of the "correlation function" $x \mapsto \mathcal{H}_C(x) + C(x) \mod 2$ does not contain 0.

Our Approach for CI-Apx: Sparse Correlations. Our first observation is that if we are able to construct a hash family $\mathcal{H}$ where, for every $C \in \mathcal{C}$, the function $x \mapsto \mathcal{H}_C(x) + C(x)$ actually has exponentially-sparse image (as a fraction of the entire space), then we obtain (somewhere statistical) CI-Apx for $\mathcal{C}$.

To see this, consider the hash function $\hat{\mathcal{H}}(x) = \mathcal{H}(x) + r \mod 2$, where r is a uniformly random string sampled together with the hash key. The task of breaking CI of $\hat{\mathcal{H}}(x)$ for some $C \in \mathcal{C}$ reduces to the task of finding x s.t. $\mathcal{H}_C(x) + C(x) = r \mod 2$. Clearly, with overwhelming probability, such x does not exist when the image of $\mathcal{H}_C(x) + C(x)$ is sufficiently small. We can push our statistical argument even further to claim CI-Apx for $\mathcal{C}$: an adversary that breaks the CI-Apx of $\hat{\mathcal{H}}$ for $\mathcal{C}$ finds x s.t. $\mathcal{H}_C(x)$ is in a small Hamming-ball around $C(x)$, i.e $\mathcal{H}_C(x) + C(x) + z = r \mod 2$, where z is a vector with relative Hamming weight at most ϵ. If $x \mapsto \mathcal{H}_C(x) + C(x)$ has exponentially-sparse image, then (for properly set parameters) so does $(x, z) \mapsto \mathcal{H}_C(x) + C(x) + z$, and therefore it is very unlikely that r is in the image.

Our goal is thus reduced to constructing a hash family $\mathcal{H}$, with indistinguishable distributions $\mathcal{H}_C$ as described above, such that, for every $C \in \mathcal{C}$, the function $x \mapsto \mathcal{H}_C(x) + C(x)$ has exponentially-sparse image.

Construction from Trapdoor Hash. Our construction of CI-Apx is based on trapdoor hash (TDH) [DGI+19]. At a high level, trapdoor hash allows us to "encrypt" any function $C : x \mapsto y$ to an encoding $\mathsf{E} : x \mapsto \mathsf{e}$ such that C is computationally hidden given a description of E and yet, for

any input x, $y = C(x)$ is *almost* information-theoretically determined by $\mathsf{e} = \mathsf{E}(x)$. More accurately, the range of the correlation $\mathsf{e} + y \pmod 2$ is *sparse*. The idea is then to use such an encoding as the hash function $\mathcal{H}_C$ described above.

More specifically, in a rate-1 TDH for a function class $\mathcal{C}$, we can generate, for any $C \in \mathcal{C}$, an encoding key ek_C that comes with a trapdoor td_C. Using the encoding key ek_C, one can compute a value $\mathsf{e} \leftarrow \mathsf{E}(\mathsf{ek}_C, x)$ which is essentially a rate-1 encoding of $C(x)$ (i.e. $|\mathsf{e}| = |C(x)|$). There exists also a decoding algorithm D which determines the value $C(x)$ as $C(x) = \mathsf{e} + \mathsf{D}(\mathsf{td}_C, \mathsf{h}, \mathsf{e})$, i.e. given e and "little additional information" about x in the form of a hash value $\mathsf{h} = \mathsf{H}(x)$ whose length is independent of the length of x. The security property we are interested in is *function privacy*: for any $C, C' \in \mathcal{C}$, the encoding keys ek_C and $\mathsf{ek}_{C'}$ are indistinguishable.

We use rate-1 TDH to construct, for every $C \in \mathcal{C}$, a hash family $\mathcal{H}_C$ such that: (i) the "correlation function" $x \mapsto \mathcal{H}_C(x) + C(x)$ has exponentially-sparse image for all $C \in \mathcal{C}$, and (ii) $\mathcal{H}_C$ and $\mathcal{H}_{C'}$ are indistinguishable, for all $C \neq C'$. This suffices to construct CI hash for any class of functions $\mathcal{F}$ with probabilistic representation in $\mathcal{C}$, as outlined above.

In the heart of our construction is the following simple observation: from the correctness of the TDH, it holds that $\mathsf{E}(\mathsf{ek}_C, x) + \mathsf{D}(\mathsf{td}_C, \mathsf{H}(x), \mathsf{e}) = C(x)$. Put differently, if we define $\mathcal{H}_C(x) = \mathsf{E}(\mathsf{ek}_C, x)$, then it holds that $\mathcal{H}_C(x) + C(x) = \mathsf{D}(\mathsf{td}_C, \mathsf{H}(x), \mathsf{e})$. This value depends on x only through its hash $\mathsf{H}(x)$. If the hash function H is sufficiently compressing, i.e. the length of the hash is much smaller than $|C(x)|$, then we obtain an exponentially-sparse image for $\mathcal{H}_C(x) + C(x)$ and, essentially, requirement (i) from above. Property (ii) follows from the function privacy of the underlying TDH. Overall, we get the following result.

Theorem 1.2 (CI-Apx from TDH, Informal)**.** *Assume there exists a rate-1 TDH for $\mathcal{C}$. Then, there exists a CI hash for relations approximable by $\mathcal{C}$ (CI-Apx for $\mathcal{C}$).*

We note that the notion of TDH that we require deviates slightly from the one defined in [DGI+19]. On one hand, they require properties that we do not, such as input privacy, and they require that the decoding algorithm is efficiently computable, whereas for our purposes inefficient decoding would have sufficed. On the other hand, we require that the underlying TDH satisfies an enhanced notion of correctness, which is satisfied by all known constructions of TDH.

We obtain CI-Apx for constant degree from standard assumptions by instantiating Theorem 1.2 based on the work of Döttling et al. [DGI+19]. They construct rate-1 TDH scheme for constant-degree polynomials from various standard assumptions, including QR, DCR and LWE. For the DDH assumption they construct TDH for a stricter class of "index functions". We show in Appendix A that their construction can be slightly adjusted, based on existing ideas, to capture also constant-degree functions and, hence, get an instantiation also from DDH.

1.2 Paper Organization

In Section 2, we provide some essential preliminaries. In Section 3, we present the framework which allows using our CI constructions to obtain NIZK, starting with the generic paradigm laid out by prior work. In Section 4, we show how to exploit a simple probabilistic representation of a function class for obtaining CI hash. In Section 5, we show our construction of CI-Apx from TDH and, lastly, in Section 6, we present an LPN-based commitment scheme that is useful for instantiating our construction of NIZK. Appendix A contains the formal proof for DDH-based TDH for linear functions that does not explicitly appear in [DGI+19].

2 Preliminaries

Notation. For an integer $n \in \mathbb{N}$, $[n]$ denotes the set $\{1, \ldots, n\}$. We use λ for the security parameter and $\mathsf{negl}(\lambda)$ and $\mathsf{poly}(\lambda)$ for a negligible function and, resp., a polynomial in λ. We use $\overset{c}{\equiv}$ and $\overset{s}{\equiv}$ to denote computational and, resp., statistical indistinguishability between two distribution ensembles. For a distribution (or a randomized algorithm) D we use $x \overset{\$}{\leftarrow} D$ to say that x is sampled according to D and use $x \in D$ to say that x is in the support of D. For a set S we overload the notation to use $x \overset{\$}{\leftarrow} S$ to indicate that x is chosen uniformly at random from S.

2.1 Learning Parity with Noise

We hereby define the standard *Decisional Learning Parity with Noise (DLPN)* assumption, which we use in this paper.

Definition 2.1 (Decisional LPN Assumption)**.** *Let $\tau : \mathbb{N} \to \mathbb{R}$ be such that $0 < \tau(\lambda) < 0.5$ for all λ, and let $n := n(\lambda)$ and $m := m(\lambda)$ be polynomials such that $m(\lambda) > n(\lambda)$ for all λ. The (n, m, τ)-*Decisional LPN *((n, m, τ)-DLPN) assumption states that for any PPT adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl} : \mathbb{N} \to \mathbb{R}$, such that*

$$|\Pr[\mathcal{A}(\mathbf{A}, \mathbf{As} + \mathbf{e}) = 1] - \Pr[\mathcal{A}(\mathbf{A}, \mathbf{b}) = 1]| < \mathsf{negl}(\lambda)$$

where $\mathbf{A} \overset{\$}{\leftarrow} \mathbb{Z}_2^{m \times n}$, $\mathbf{s} \overset{\$}{\leftarrow} \mathbb{Z}_2^n$, $\mathbf{e} \overset{\$}{\leftarrow} \mathsf{Ber}_\tau^m$ and $\mathbf{b} \overset{\$}{\leftarrow} \mathbb{Z}_2^m$.

It is well-known that DLPN remains secure even given polynomially many samples of independent secrets and error vectors.

Proposition 2.2. *Let τ, n and m be as in Definition 2.1 above, and let $k := k(\lambda)$ be an aribitrary polynomial in the security parameter. Then, under the (n, m, τ)-DLPN assumption, for any PPT adversary $\mathcal{A}$, there exists a negligible function negl such that*

$$|\Pr[\mathcal{A}(\mathbf{A}, \mathbf{AS} + \mathbf{E}) = 1] - \Pr[\mathcal{A}(\mathbf{A}, \mathbf{B}) = 1]| < \mathsf{negl}(\lambda)$$

where $\mathbf{A} \overset{\$}{\leftarrow} \mathbb{Z}_2^{m \times n}$, $\mathbf{S} \overset{\$}{\leftarrow} \mathbb{Z}_2^{n \times k}$, $\mathbf{E} \overset{\$}{\leftarrow} \mathsf{Ber}_\tau^{m \times k}$ and $\mathbf{B} \overset{\$}{\leftarrow} \mathbb{Z}_2^{m \times k}$.

2.2 Trapdoor Hash

We hereby recall the definition of *trapdoor hash functions* (TDH) from Döttling et al. [DGI+19], with few minor modifications. First, we are fine with weakly correct trapdoor hash schemes (as defined in [DGI+19]), where we allow the error in correctness to be two-sided. This modification further allows us to simplify the syntax of decoding for rate-1 schemes. Second, to construct correlation intractable hash, we do not require the trapdoor hash scheme to be input-private (i.e. that the hash of an input x hides x) and, consequently, we assume w.l.o.g. that the hash and encoding functions, H and E, are deterministic (in the original definition, H and E share the same randomness - this was necessary for achieving both input privacy and correctness).

Definition 2.3 (Rate-1 Trapdoor Hash)**.** *A* rate-1 trapdoor hash scheme (TDH) *for a function class $\mathcal{C} = \{\mathcal{C}_n : \{0,1\}^n \to \{0,1\}\}$ is a tuple of five PPT algorithms $\mathsf{TDH} = (\mathsf{S}, \mathsf{G}, \mathsf{H}, \mathsf{E}, \mathsf{D})$ with the following properties.*

- ***Syntax:***
  - $\mathsf{hk} \leftarrow \mathsf{S}(1^\lambda, 1^n)$. *The sampling algorithm takes as input a security parameter* λ *and an input length* n*, and outputs a hash key* hk.
  - $(\mathsf{ek}, \mathsf{td}) \leftarrow \mathsf{G}(\mathsf{hk}, C)$. *The generating algorithm takes as input a hash key* hk *a function* $C \in \mathcal{C}_n$*, and outputs a pair of an encoding key* ek *and a trapdoor* td.
  - $\mathsf{h} \leftarrow \mathsf{H}(\mathsf{hk}, \mathsf{x})$. *The hashing algorithm takes as input a hash key* hk *and a string* $\mathsf{x} \in \{0,1\}^n$*, and deterministically outputs a hash value* $\mathsf{h} \in \{0,1\}^\eta$.
  - $\mathsf{e} \leftarrow \mathsf{E}(\mathsf{ek}, \mathsf{x}; \rho)$. *The encoding algorithm takes as input an encoding key* ek *and a string* $\mathsf{x} \in \{0,1\}^n$*, and deterministically outputs an encoding* $\mathsf{e} \in \{0,1\}$.
  - $\mathsf{e}' \leftarrow \mathsf{D}(\mathsf{td}, \mathsf{h})$. *The decoding algorithm takes as input a trapdoor* td*, a hash value* $\mathsf{h} \in \{0,1\}^\eta$*, and outputs a 0-encoding* $\mathsf{e}' \in \{0,1\}$.
- ***Correctness:*** TDH *is (weakly)* $(1-\tau)$-correct *(or has two-sided* τ *error probability), for* $\tau := \tau(\lambda) < 1$*, if there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that the following holds for any* $\lambda, n \in \mathbb{N}$*, any* $\mathsf{x} \in \{0,1\}^n$ *and any function* $C \in \mathcal{C}_n$.

$$\Pr[\mathsf{e} + \mathsf{e}' = C(x) \mod 2] \geq 1 - \tau - \mathsf{negl}(\lambda)$$

  where $\mathsf{hk} \leftarrow \mathsf{S}(1^\lambda, 1^n)$*,* $(\mathsf{ek}, \mathsf{td}) \leftarrow \mathsf{G}(\mathsf{hk}, C)$*,* $\mathsf{h} \leftarrow \mathsf{H}(\mathsf{hk}, \mathsf{x})$*,* $\mathsf{e} \leftarrow \mathsf{E}(\mathsf{ek}, \mathsf{x})$*, and* $\mathsf{e}' \leftarrow \mathsf{D}(\mathsf{td}, \mathsf{h})$*. When* $\tau = 0$ *we say that the scheme is* fully correct.
- ***Function Privacy:*** TDH *is* function-private *if for any polynomial-length* $\{1^{n_\lambda}\}_{\lambda \in \mathbb{N}}$ *and any* $\{f_n\}_{n \in \mathbb{N}}$ *and* $\{f'_n\}_{n \in \mathbb{N}}$ *such that* $f_n, f'_n \in \mathcal{F}_n$ *for all* $n \in \mathbb{N}$*, it holds that*

$$\{(\mathsf{hk}_\lambda, \mathsf{ek}_\lambda)\}_{\lambda \in \mathbb{N}} \stackrel{c}{\equiv} \{(\mathsf{hk}_\lambda, \mathsf{ek}'_\lambda)\}_{\lambda \in \mathbb{N}}$$

  where $\mathsf{hk}_\lambda \stackrel{\$}{\leftarrow} \mathsf{S}(1^\lambda, 1^{n_\lambda})$*,* $(\mathsf{ek}_\lambda, \mathsf{td}_\lambda) \stackrel{\$}{\leftarrow} \mathsf{G}(\mathsf{hk}_\lambda, f_{n_\lambda})$ *and* $(\mathsf{ek}'_\lambda, \mathsf{td}'_\lambda) \stackrel{\$}{\leftarrow} \mathsf{G}(\mathsf{hk}_\lambda, f'_{n_\lambda})$.
- ***Compactness:*** *we require that the image length of the hash function,* η*, is independent of* n*, and is bounded by some polynomial in the security parameter* λ.

As pointed in [DGI+19] (Remark 4.2), we may consider a natural extension of trapdoor hash for a general class of functions $\mathcal{C} = \{\mathcal{C}_n : \{0,1\}^n \to \{0,1\}^m\}$ (where $m := m(\lambda) > 1$ is a fixed polynomial). Further, if any $C \in \mathcal{C}_n$ can be represented as m parallel computations in some class $\mathcal{C}'_n : \{0,1\}^n \to \{0,1\}$, then a trapdoor hash scheme for $\mathcal{C}' = \{\mathcal{C}'_n\}$ directly implies a trapdoor hash scheme for $\mathcal{C}$ with hash length independent in m.

2.3 Extractable Commitments

We hereby provide the definition of an extractable commitment scheme.[4]

Definition 2.4 (Extractable Commitment)**.** *An* extractable (bit) commitment scheme *is a tuple of four PPT algorithms* $\mathsf{Com} = (\mathsf{Gen}, \mathsf{Commit}, \mathsf{Verify}, \mathsf{Extract})$ *with the following properties.*

- ***Syntax:***

[4]The notion of extractable commitment is equivalent to standard public-key encryption. We use the commitment terminology since it is more natural for our setting.

- $(\mathsf{pk}, \mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda)$*: The key generation algorithm takes as input the security parameter* 1^λ *and outputs a pair of a public key* pk *and trapdoor* td*.*
- $\mathsf{com} \leftarrow \mathsf{Commit}(\mathsf{pk}, \mathsf{x}; r)$*: The committing algorithm takes as input a public key* pk*, a bit* $\mathsf{x} \in \{0,1\}$ *and randomness* r*, and outputs a commitment* com*.*
- $\{0,1\} \leftarrow \mathsf{Verify}(\mathsf{pk}, \mathsf{com}, \mathsf{x}; r)$*: The verification algorithm takes as input a public key* pk*, a commitment* com*, a bit* $\mathsf{x} \in \{0,1\}$ *and randomness* $r \in \{0,1\}^*$*, then either accepts or rejects.*
- $\mathsf{x}' \leftarrow \mathsf{Extract}(\mathsf{td}, \mathsf{com})$*: The extraction algorithm takes as input a trapdoor* td *and a commitment* com *and outputs a bit* $\mathsf{x}' \in \{0,1\}$ *or* $\perp$*.*

- ***Correctness****:* Com *is* correct *if there exists a negligible function* negl*, such that for any* $\mathsf{x} \in \{0,1\}$*,*

$$\Pr[\mathsf{Verify}(\mathsf{pk}, \mathsf{Commit}(\mathsf{pk}, \mathsf{x}; r), \mathsf{x}; r)] > 1 - \mathsf{negl}(\lambda)$$

  where $(\mathsf{pk}, \cdot) \xleftarrow{\$} \mathsf{Gen}(1^\lambda)$ *and* $r \xleftarrow{\$} \{0,1\}^*$*.*

- ***Hiding****:* Com *is* (computationally) hiding *if it holds that*

$$\{\mathsf{Commit}(\mathsf{pk}, 0; r)\}_\lambda \stackrel{c}{\equiv} \{\mathsf{Commit}(\mathsf{pk}, 1; r)\}_\lambda$$

  where $(\mathsf{pk}, \cdot) \xleftarrow{\$} \mathsf{Gen}(1^\lambda)$ *and* $r \xleftarrow{\$} \{0,1\}^*$ *for all* $\lambda \in \mathbb{N}$*.*

- ***Binding****:* Com *is* (statistically) binding *if there exists a negligible function* negl *such that*

$$\Pr[\exists \mathsf{com}, r_0, r_1 \quad s.t. \quad \mathsf{Verify}(\mathsf{pk}, \mathsf{com}, 0, r_0) = \mathsf{Verify}(\mathsf{pk}, \mathsf{com}, 1, r_1) = 1] < \mathsf{negl}(\lambda)$$

  where $(\mathsf{pk}, \cdot) \xleftarrow{\$} \mathsf{Gen}(1^\lambda)$*.*

- ***Extraction****:* Com *has* correct extraction *if there exists a negligible function* negl*, such that for any* $\mathsf{x} \in \{0,1\}$ *and* $r \in \{0,1\}^*$*, if* $\mathsf{Verify}(\mathsf{pk}, \mathsf{Commit}(\mathsf{pk}, \mathsf{x}; r), \mathsf{x}; r)$

$$\Pr[\mathsf{Verify}(\mathsf{pk}, \mathsf{com}, \mathsf{x}; r) = 1 \wedge \mathsf{Extract}(\mathsf{td}, \mathsf{com}) \neq \mathsf{x}] < \mathsf{negl}(\lambda)$$

  where $(\mathsf{pk}, \mathsf{td}) \xleftarrow{\$} \mathsf{Gen}(1^\lambda)$ *and* $\mathsf{com} = \mathsf{Commit}(\mathsf{pk}, \mathsf{x}; r)$*.*

Remark 2.5. *Throughout the paper, we will implicitly assume that if* $\mathsf{Commit}(\mathsf{pk}, \mathsf{x}; r) \neq \mathsf{com}$ *then* $\mathsf{Verify}(\mathsf{pk}, \mathsf{x}, r) \neq 1$*. This is achieved by any commitment scheme with a natural verification function (that possibly performs additional verification). Notice that in such a case correct extraction implies statistical binding.*

2.4 Non-Interactive Zero-Knowledge Arguments

We formally define non-interactive zero knowledge arguments as follows.

Definition 2.6 (Non-Interactive Zero Knowledge)**.** *Let* $n := n(\lambda)$ *be a polynomial in the security parameter. A* non-interactive zero knowledge (NIZK) argument Π *for an NP language* L*, with a corresponding instance-witness relation* R*, consists of three PPT algorithms* $\Pi = (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ *with the following properties.*

- ***Syntax:***
  - $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$*: the setup algorithm takes a security parameter* 1^λ *and ouputs a common reference string* crs*.*
  - $\pi \leftarrow \mathsf{P}(\mathsf{crs}, \mathsf{x}, \mathsf{w})$*: the prover takes ass input the common reference string* crs*, a statement* $\mathsf{x} \in \{0,1\}^n$ *and a witness* w *such that* $(\mathsf{x}, \mathsf{w}) \in R$*, and outputs a proof* π*.*
  - $\{0,1\} \leftarrow \mathsf{V}(\mathsf{crs}, \mathsf{x}, \pi)$*: the verifier takes as input the common reference string* crs*, a statement* $\mathsf{x} \in \{0,1\}^n$ *and a proof* π*, and either accepts (outputs 1) or rejects (outputs 0).*
- ***Completeness:*** Π *is* complete *if for every* $\lambda \in \mathbb{N}$ *and* $(\mathsf{x}, \mathsf{w}) \in R$*, it holds that*

$$\Pr[\mathsf{V}(\mathsf{crs}, \mathsf{x}, \mathsf{P}(\mathsf{crs}, \mathsf{x}, \mathsf{w}))] = 1$$

  where $\mathsf{crs} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$*.*
- ***Soundness:*** Π *is* sound *if for every PPT cheating prover* P^**, there exists a negligible function* negl*, such that for every* $\{\mathsf{x}_\lambda \notin L\}_\lambda$ *where* $\mathsf{x}_\lambda \in \{0,1\}^n$ *for all* λ*, it holds that*

$$\Pr[\mathsf{V}(\mathsf{crs}, \mathsf{x}_\lambda, \mathsf{P}^*(\mathsf{crs})) = 1] < \mathsf{negl}(\lambda)$$

  where $\mathsf{crs} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$*.*
- ***Zero Knowledge:*** Π *is* zero knowledge *if there exists a PPT simulator* Sim *such that for every* $\{(\mathsf{x}_\lambda, \mathsf{w}_\lambda) \in R\}_\lambda$*, where* $\mathsf{x}_\lambda \in \{0,1\}^n$ *for all* $\lambda \in \mathbb{N}$*, it holds that*

$$\{(\mathsf{crs}, \mathsf{P}(\mathsf{crs}, \mathsf{x}_\lambda, \mathsf{w}_\lambda))\}_\lambda \stackrel{c}{\equiv} \{\mathsf{Sim}(1^\lambda, \mathsf{x}_\lambda)\}_\lambda$$

  where $\mathsf{crs} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$*.*

We further consider few optional stronger properties that a NIZK system can satisfy:

- ***Adaptive Soundness:*** Π *is* adaptively sound *if for every PPT cheating prover* P^**, there exists a negligible function* negl*, such that*

$$\Pr[\mathsf{x} \notin L \ \wedge \ \mathsf{V}(\mathsf{crs}, \mathsf{x}, \pi) = 1] < \mathsf{negl}(\lambda)$$

  where $\mathsf{crs} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$ *and* $(\mathsf{x}, \pi) \leftarrow \mathsf{P}^*(\mathsf{crs})$*.*
- ***Adaptive Zero Knowledge:*** Π *is* adaptively zero knowledge *if for every PPT dishonest verifier* V^**, there exists a PPT simluator* Sim*, such that*

$$\{(\mathsf{crs}, \mathsf{P}(\mathsf{crs}, \mathsf{x}, \mathsf{w}), \mathsf{aux})\}_\lambda \stackrel{c}{\equiv} \{\mathsf{Sim}(1^\lambda)\}_\lambda$$

  where $\mathsf{crs} \xleftarrow{\$} \mathsf{Sample}(1^\lambda)$ *and* $(\mathsf{x}, \mathsf{w}, \mathsf{aux}) \leftarrow \mathsf{V}^*(\mathsf{crs})$*.*

2.5 Correlation Intractability

Correlation intractable hash [CGH04] constitutes one of the main building blocks in our work. We hereby provide a formal definition.

Definition 2.7 (Correlation Intractable Hash)**.** *Let $\mathcal{R} = \{\mathcal{R}_\lambda\}$ be a relation class. A hash family $\mathcal{H} = (\mathsf{Sample}, \mathsf{Hash})$ is said to be* correlation intractable *for $\mathcal{R}$ if for every non-uniform polynomial-time adversary $\mathcal{A} = \{\mathcal{A}_\lambda\}$, there exists a negligible function $\mathsf{negl}(\lambda)$, such that for every $R \in \mathcal{R}_\lambda$, it holds that*

$$\Pr[(x, \mathsf{Hash}(\mathsf{k}, x)) \in R] \leq \mathsf{negl}(\lambda)$$

where $\mathsf{k} \xleftarrow{\$} \mathsf{Sample}(1^\lambda)$ and $x = \mathcal{A}_\lambda(\mathsf{k})$.

We further define an essential property for utilizing CI hash for obtaining NIZK protocols.

Definition 2.8 (Programmable Hash Family)**.** *A hash family $\mathcal{H} = (\mathsf{Sample}, \mathsf{Hash})$, with input and output length $n := n(\lambda)$ and, resp., $m := m(\lambda)$, is said to be* programmable *if the following two conditions hold:*

- ***1-Universality**: For every $\lambda \in \mathbb{N}$, $x \in \{0,1\}^n$ and $y \in \{0,1\}^m$,*

  $$\Pr[\mathsf{Hash}_\mathsf{k}(x) = y] = 2^{-m}$$

  where $\mathsf{k} \xleftarrow{\$} \mathsf{Sample}(1^\lambda)$.

- ***Programmability**: There exists a PPT algorithm $\widetilde{\mathsf{Sample}}(1^\lambda, x, y)$ that samples from the conditional distribution $\mathsf{Sample}(1^\lambda) \mid \mathsf{Hash}_\mathsf{k}(x) = y$.*

3 Non-Interactive Zero Knowledge from Correlation Intractability

In this section, we provide the formal framework for constructing NIZK for NP from the following building blocks:

(i) An extractable commitment scheme where the extraction function can be probabilistically presented by constant-degree polynomials.

(ii) A correlation intractable hash function for relations probabilistically searchable by constant-degree polynomials.

Our framework is essentially a special case of a more general paradigm that was extensively investigated in prior works [KRR17, CCRR18, CCH+19] for constructing NIZKs from general correlation intractability. Our contribution in this part of the paper is relaxing the requirement for correlation intractability, assuming a commitment scheme with the above property exists.

3.1 A Generic Framework

We first recall the generic framework from Canetti et al. [CCH+19] for achieving non-interactive zero knowledge systems from correlation intractable hash.

In its most general form, the paradigm applies the Fiat-Shamir transform [FS87] over Σ-protocols, which are special honest-verifier ZK protocols (possibly in the CRS model), using correlation intractable hash, in a provably-sound manner.

Roughly speaking, in Σ-protocols, for every prover's first message a there exists (if any) a unique verifier's challenge e that may allow a cheating prover to cheat. Thus, if we instantiate Fiat-Shamir using a hash family $\mathcal{H}$ that is CI for the relation between such pairs (a, e), then the soundness of the transform can be reduced to the correlation intractability of $\mathcal{H}$: any prover who finds a first message a where $\mathcal{H}(a)$ is the "bad challenge" e, essentially breaks $\mathcal{H}$.

Therefore, the type of relations we target in the above outline is formally specified as follows.

Definition 3.1 (Unique-Output Relations)**.** *We say that a class of relations $\mathcal{R} \subset \{0,1\}^n \times \{0,1\}^m$ is* unique-output *if for every $x \in \{0,1\}^n$ there exists at most one value $y \in \{0,1\}^m$ such that $(x, y) \in \mathcal{R}$. We sometimes use function notation to describe such an $\mathcal{R}$ where every $R \in \mathcal{R}$ is denoted by a function $R : \{0,1\}^n \to \{0,1\}^m \cup \perp$ with $R(x) = y$ for $(x, y) \in R$ and $R(x) = \perp$ if there exists no such y.*

As observed in [CCH+19], we can reduce the class of relations we target in the CI to relations that are efficiently searchable, i.e. unique-output relations where the unique output is efficiently computable. It is not the case, however, that any Σ-protocol defines such a corresponding relation. This leads us to define *trapdoor Σ-protocols* [CCH+19], which are Σ-protocol where the relation between a prover's first message and its unique "bad challenge" is efficiently computable given a trapdoor. We formalize below.

Definition 3.2 (Searchable Relations)**.** *Let $\mathcal{R} : \{0,1\}^n \to \{0,1\}^m \cup \perp$ be a unique-output class of relations. We say that $\mathcal{R}$ is searchable by a function class $\mathcal{F} : \{0,1\}^n \to \{0,1\}^m \cup \perp$ if for every $R \in \mathcal{R}$, there exists $f_R \in \mathcal{F}$ such that*

$$\forall x \quad s.t. \quad R(x) \neq \perp, \quad (x, f_R(x)) \in R$$

We say that $\mathcal{R}$ is efficiently searchable *if $\mathcal{F}$ is efficiently computable.*

Definition 3.3 (Trapdoor Σ-Protocol [CCH+19])**.** *Let $\Pi = (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ be a public-coin three-message honest-verifier zero knowledge proof system for a language L in the common reference string model. Define the relation class $\mathcal{R}_\Sigma(\Pi) = \{R_{\mathsf{crs},x} \mid \mathsf{crs} \in \mathsf{Setup}(1^\lambda), x \notin L\}$ where*

$$R_{\mathsf{crs},x} = \{(\mathbf{a}, \mathbf{e}) \mid \exists \mathbf{z} \quad s.t. \quad \mathsf{V}(\mathsf{crs}, x, \mathbf{a}, \mathbf{e}, \mathbf{z}) = 1\}$$

We say that Π for L is a trapdoor Σ-protocol *if $R_{\mathsf{crs},\mathsf{x}}$ is a unique-output relation (see Definition 3.1) and there exist two PPT algorithms,* $\mathsf{tdSetup}$ *and* $\mathsf{BadChallenge}$*, with the following properties:*

- ***Syntax:***
  - $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{tdSetup}(1^\lambda)$*: The trapdoor setup algorithms takes as input a security parameter 1^λ and outputs a common reference string* crs *and a trapdoor* td*.*
  - $e \leftarrow \mathsf{BadChallenge}(\mathsf{crs}, \mathsf{td}, \mathsf{x}, a)$*: The bad challenge algorithm takes as input a common reference string* crs *and its trapdoor* td*, an instance* x*, and a first message a, and outputs a second message e or $\perp$.*

- ***CRS Indistinguishability:*** *We require that a common reference string* $\mathsf{crs} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$ *is computationally indistinguishable from a random reference string* crs' *sampled with a trapdoor by* $(\mathsf{crs}', \mathsf{td}) \xleftarrow{\$} \mathsf{tdSetup}(1^\lambda)$.

- ***Correctness:*** *We require that for all* $\lambda \in \mathbb{N}$ *and any instance* $\mathsf{x} \notin L$*, first message* a*, and* $(\mathsf{crs}, \mathsf{td})$*, such that* $R_{\mathsf{crs},\mathsf{x}}(a) \neq \perp$*, it holds*

$$\mathsf{BadChallenge}(\mathsf{crs}, \mathsf{td}, \mathsf{x}, a) = R_{\mathsf{crs},x}(a)$$

  Equivalently, we require that $\mathcal{R}_\Sigma(\Pi)$ *is searchable by*

$$\mathcal{F}_\Sigma(\Pi) = \{f_{\mathsf{crs},\mathsf{td},x}(a) = \mathsf{BadChallenge}(\mathsf{crs}, \mathsf{td}, x, \cdot) \mid (\mathsf{crs}, \mathsf{td}) \in \mathsf{Setup}(1^\lambda), x \notin L\}$$

We recall the following theorem from [CCH+19].

Theorem 3.4 ([CCH+19])**.** *Assume:*

(i) Π *is a trapdoor* Σ*-protocol for* L*.*

(ii) $\mathcal{H}$ *is a programmable correlation intractable hash family for relation searchable by* $\mathcal{F}_\Sigma(\Pi)$*.*

Then, the Fiat-Shamir [FS87] transform over Π *using* $\mathcal{H}$*,* $\mathsf{FS}(\Pi, \mathcal{H})$*, is an NIZK argument system for* L *with adaptive soundness and adaptive zero-knowledge.*

Canetti et al. [CCH+19] show that any correlation intractable hash family for a reasonable class of relations can be easily transformed to a programmable hash family while preserving correlation intractability. We stress, however, that our Construction of correlation intractable hash in Section 2.5 directly satisfies programmability.

3.2 Special Case: Commit-then-Open Protocols

Equipped with the generic framework laid by prior work, we may now present a special case that comprises the starting point of our work.

3.2.1 Commit-then-Open Protocols.

We consider protocols of a special form called *commit-then-open* Σ*-protocols*. This notion captures a natural approach for constructing ZK protocols. In particular, a variant of the ZK protocol for Graph Hamiltonicity from [Blu87, FLS99] is a commit-then-open Σ-protocol.

Roughly speaking, commit-then-open Σ-protocols are protocols that use a commitment scheme (possibly in the CRS model), where the prover's first message is a commitment on some proof string π, and his second message is always a decommitment on a subset of π, which depends on the verifier's challenge. Upon receiving the decommitments, the verifier checks that they are valid, then runs some verification procedure on the opened values. We hereby provide a formal definition.

Definition 3.5 (Commit-then-Open Σ-Protocols)**.** *A* commit-then-open Σ-protocol *is a* Σ*-protocol* $\Pi^{\mathsf{Com}} = (\mathsf{Setup}^{\mathsf{Com}}, \mathsf{P}^{\mathsf{Com}}, \mathsf{V}^{\mathsf{Com}})$*, with black-box access to a commitment scheme* Com *(possibly in the CRS model), such that there exist four PPT algorithms:*

- $\mathsf{crs}' \leftarrow \mathsf{Setup}'(1^\lambda, \mathsf{pk})$*: Takes as input a security parameter* 1^λ *and a commitment key* pk*, and outputs a common reference string* crs'*.*

- $(\pi, \mathsf{state}) \leftarrow \mathsf{P}_1(\mathsf{crs}, \mathsf{x}, \mathsf{w})$*: Takes as input a common reference string* crs*, an instance* x *and its witness* w *and outputs a proof* $\pi \in \{0,1\}^\ell$ *(for some polynomial* $\ell := \ell(\lambda)$*) and a local state* state*.*

- $I \leftarrow \mathsf{P}_2(\mathsf{crs}, \mathsf{x}, \mathsf{w}, e, \mathsf{state})$*: Takes as input* crs*,* x*,* w *and* state *as above, and a verifier's challenge* $e \in \{0,1\}^*$*, and outputs a subset* $I \subseteq [\ell]$*.*

- $\{0,1\} \leftarrow \mathsf{V}'(\mathsf{crs}, \mathsf{x}, e, (I, \pi_I))$*: Takes as input* crs*,* x*,* $e \in \{0,1\}^*$*,* $I \subseteq [\ell]$ *as above, and a substring of the proof* $\pi_I \in \{0,1\}^{|I|}$*.*

using which Π^{Com} *is defined as follows:*

- $\mathsf{Setup}^{\mathsf{Com}}(1^\lambda)$*: Sample a commitment key* $\mathsf{pk} \leftarrow \mathsf{Com.Gen}(1^\lambda)$ *and possibly additional output* $\mathsf{crs}' \leftarrow \mathsf{Setup}'(1^\lambda, \mathsf{pk})$*, and output*
$$\mathsf{crs} = (\mathsf{crs}', \mathsf{pk})$$

- $\mathsf{P}^{\mathsf{Com}}(\mathsf{crs}, \mathsf{x}, \mathsf{w})$*: The prover computes* $(\pi, \mathsf{state}) \leftarrow \mathsf{P}_1(\mathsf{crs}, \mathsf{x}, \mathsf{w})$*, keeps the local state* state*, and sends a commitment on the proof* π *to the verifier,*
$$a = \mathsf{Com.Commit}(\mathsf{pk}, \pi)$$

- $\mathsf{P}^{\mathsf{Com}}(\mathsf{crs}, \mathsf{x}, \mathsf{w}, e)$*: The prover's second message consists of a decommitment on the proof bits corresponding to locations* $I \leftarrow \mathsf{P}_2(\mathsf{crs}, \mathsf{x}, \mathsf{w}, e, \mathsf{state})$*,*
$$z = (I, \mathsf{Com.Decommit}(a_I))$$

- $\mathsf{V}^{\mathsf{Com}}(\mathsf{crs}, \mathsf{x}, a, e, z)$*: The verifier verifies that* z *contains a valid decommitment to* π_I *and outputs*
$$\mathsf{V}'(\mathsf{crs}, \mathsf{x}, e, (I, \pi_I))$$

We sometimes override notation and denote $\Pi^{\mathsf{Com}} = (\mathsf{Setup}', \mathsf{P}_1, \mathsf{P}_2, \mathsf{V}')$*.*

Proposition 3.6 ([Blu87, FLS99])**.** *There exists a commit-then-open* Σ*-protocol with soundness* $1/2$ *for an NP-complete language* L*.*

It turns out that commit-then-open Σ-protocols allow us to relax the CI requirement for a sound Fiat-Shamir to CI for relations that are *probabilistically searchable* by constant-degree polynomials. We elaborate in the following.

3.3 Probabilistically Searchable Relations

We consider a standard notion of approximation, which we refer to as *probabilistic representation*. Roughly speaking, a function f is probabilistically represented by a function class $\mathcal{C}$ if there exists a randomized $C \in \mathcal{C}$ that computes f with high probability, on any input.

Definition 3.7 (Probabilistic Representation)**.** *Let $n, m \in \mathbb{N}$ and $0 < \epsilon < 1$. Let $f : \{0,1\}^n \to \{0,1\}^m \cup \perp$ be a function and denote $f(x) = (f_1(x), \ldots, f_m(x))$ where $f_i : \{0,1\}^n \to \{0,1\} \cup \perp$ for all $i \in [m]$. A* (bit-by-bit) *ϵ-probabilistic representation of f by a class of functions $\mathcal{C} : \{0,1\}^n \to \{0,1\}$ consists of m distributions $\mathfrak{C}_1, \ldots, \mathfrak{C}_m \subseteq \mathcal{C}$ such that*

$$\forall i \in [m],\ \forall x \quad s.t. \quad f(x) \neq \perp, \qquad \Pr_{C_i \xleftarrow{\$} \mathfrak{C}_i} [f_i(x) = C_i(x)] > 1 - \epsilon$$

The following simple lemma connects between probabilistic representation and approximation. Its proof follows immediately from Chernoff's tail bound.

Lemma 3.8 (From Probabilistic Representation to Approximation)**.** *Let $n \in \mathbb{N}$, $\epsilon := \epsilon(\lambda) > 0$, and $m := m(\lambda)$ be a sufficiently large polynomial. For any $\lambda \in \mathbb{N}$, let $f : \{0,1\}^n \to \{0,1\}^m \cup \perp$, and let $\mathfrak{C} = (\mathfrak{C}_1, \ldots, \mathfrak{C}_m)$ be an ϵ-probabilistic representation of f by $\mathcal{C} : \{0,1\}^n \to \{0,1\}$. Then, there exists a negligible function* negl*, such that*

$$\forall x \quad s.t. \quad f(x) \neq \perp, \qquad \Pr_{C \xleftarrow{\$} \mathfrak{C}} [\Delta(f(x), C(x)) > 2\epsilon m] < \mathsf{negl}(\lambda)$$

If a class of functions $\mathcal{R}$ is searchable by functions with probabilistic representation by $\mathcal{C}$, we say that $\mathcal{R}$ is *probabilistically searchable* by $\mathcal{C}$.

Definition 3.9 (Probabilistically-Searchable Relations)**.** *Let $\mathcal{R} : \{0,1\}^n \to \{0,1\}^m \cup \perp$ be a unique-output class of relations. We say that $\mathcal{R}$ is* ϵ-probabilistically searchable by *$\mathcal{C} : \{0,1\}^n \to \{0,1\}$ if it is searchable by $\mathcal{F}$ and, for every $R \in \mathcal{R}$, letting $f_R \in \mathcal{F}$ be the corresponding search function (see Definition 3.2), $f_R \in \mathcal{F}$ has an ϵ-probabilistic representation by $\mathcal{C}$.*

Notice that CI for relations searchable by $\mathcal{F}$ is a weaker notion than relation probabilistically searchable by $\mathcal{F}$. Our hope is to be able to probabilistically represent $\mathcal{F}$ by a much simpler class of functions $\mathcal{C}$ such that the CI task is actually simplified.

3.4 CI for Probabilistic Constant-Degree is Sufficient for NIZK

Lastly, we show that through commit-then-open protocols, we can reduce our task to achieving CI for relations probabilistically searchable by constant-degree polynomials. More specifically, we show that any commit-then-open Σ-protocol Π^{Com} can be transformed to a slightly different commit-then-open Σ-protocol $\widetilde{\Pi}^{\mathsf{Com}}$ such that:

- Assuming Com is extractable, $\widetilde{\Pi}^{\mathsf{Com}}$ is a trapdoor Σ-protocol.
- Assuming, further, that the extraction function $f_{\mathsf{td}}(a) = \mathsf{Com.Extract}(\mathsf{td}, a)$ has probabilistic constant-degree representation, then so does the trapdoor function $\mathsf{BadChallenge}$, corresponding to $\widetilde{\Pi}^{\mathsf{Com}}$ and, therefore $\mathcal{R}_\Sigma(\widetilde{\Pi}^{\mathsf{Com}})$ is probabilistically searchable by constant-degree polynomials.

We formalize below.

Theorem 3.10. *Let Π^{Com} be a commit-then-open Σ-protocol for L with soundness $1/2$ where the output of P_1 is of length $\ell := \ell(\lambda)$. Let Com be a statistically-binding extractable commitment scheme where, for any td, the function $f_{\mathsf{td}}(x) = \mathsf{Com.Extract}(\mathsf{td}, x)$ has an ϵ-probabilistic representation*

by c-degree polynomials, for a constant $c \in \mathbb{N}$ and $0 < \epsilon(\lambda) < 1/\ell$. Then, for any polynomial $m := m(\lambda)$, there exists a trapdoor Σ-protocol $\widetilde{\Pi}^{\mathsf{Com}}$ for L with soundness 2^{-m} such that $\mathcal{R}_{\Sigma}(\widetilde{\Pi}^{\mathsf{Com}})$ (see Definition 3.3) is ϵ'-probabilistically searchable by $6cc'$-degree polynomials, where $c' \in \mathbb{N}$ is an arbitrary constant and $\epsilon' = \ell \cdot \epsilon + 2^{-c'}$.

Combining Proposition 3.6, Theorem 3.10, and Theorem 3.4, we obtain the following.

Corollary 3.11 (Sufficient Conditions for NIZK for NP)**.** *The following conditions are sufficient to obtain a NIZK argument system for NP (with adaptive soundness and adaptive zero-knowledge):*

(i) A statistically-binding extractable commitment scheme where, for any td, the function $f_{\mathsf{td}}(x) = \mathsf{Extract}(\mathsf{td}, x)$ has an ϵ-probabilistic representation by c-degree polynomials, for a constant $c \in \mathbb{N}$ and $0 < \epsilon(\lambda) < 1/\ell(\lambda)$ for an arbitrarily large polynomial ℓ.

(ii) A programmable correlation intractable hash family for relations ϵ-probabilistically searchable by c'-degree polynomials, for some constant $\epsilon > 0$ and arbitrarily large constant $c' \in \mathbb{N}$.

We now proceed and prove Theorem 3.10.

3.4.1 Proof of Theorem 3.10.

We start by presenting the transformation from Π^{Com} to $\widetilde{\Pi}^{\mathsf{Com}}$. In fact, for simplicity, we first show how to construct a protocol $\widetilde{\Pi}_1^{\mathsf{Com}}$ which has soundness $\frac{1}{2}$. The final protocol $\widetilde{\Pi}^{\mathsf{Com}}$ with amplified soundness simply consists of m parallel repetitions of $\widetilde{\Pi}_1^{\mathsf{Com}}$. We later show that all required properties are preserved under parallel repetition and, therefore, we now focus on $\widetilde{\Pi}_1^{\mathsf{Com}}$.

Using the Cook-Levin approach, we represent any (poly-size) circuit C as a (poly-size) 3-CNF formula Φ_C such that for any input x, $C(x) = 1$ if and only if there exists an assignment w for which $\Phi_C(x, w) = 1$. We call such an assignment w a *Cook-Levin witness for* $C(x)$.

Construction 3.1. *Let $\Pi^{\mathsf{Com}} = (\mathsf{Setup}', \mathsf{P}_1, \mathsf{P}_2, \mathsf{V}')$ be a commit-then-open Σ-protocol with soundness $1/2$, i.e. where the verifier's challenge e consists of a single public coin. We construct a commit-then-open Σ-protocol $\widetilde{\Pi}_1^{\mathsf{Com}} = (\mathsf{Setup}', \widetilde{\mathsf{P}}_1, \widetilde{\mathsf{P}}_2, \widetilde{\mathsf{V}}')$ as follows.*[5]

- $\widetilde{\mathsf{P}}_1(\mathsf{crs}, \mathsf{x}, \mathsf{w})$*: The prover generates a proof $\pi \leftarrow \mathsf{P}_1(\mathsf{crs}, \mathsf{x}, \mathsf{w})$ and computes $I_0 \leftarrow \mathsf{P}_2(\mathsf{crs}, \mathsf{x}, \mathsf{w}, 0)$ and $I_1 \leftarrow \mathsf{P}_2(\mathsf{crs}, \mathsf{x}, \mathsf{w}, 1)$. Without loss of generality, we assume that subsets $I_0, I_1 \subseteq [\ell]$ are represented, in the natural way, as matrices over $\mathbb{Z}_2$ such that $I_e \cdot \pi = \pi_{I_e}$ (for $e \in \{0, 1\}$). It then generates, for every $e \in \{0, 1\}$, a Cook-Levin witness w_e for the computation $C_{\mathsf{crs},\mathsf{x},e}(I_e, \pi_{I_e}) = 1$ where*

$$C_{\mathsf{crs},\mathsf{x},e}(I_e, \pi_{I_e}) := \mathsf{V}'(\mathsf{crs}, \mathsf{x}, e, I_e, \pi_{I_e})$$

The prover then outputs

$$\tilde{\pi} = (\pi, I_0, I_1, w_0, w_1)$$

- $\widetilde{\mathsf{P}}_2(\mathsf{crs}, \mathsf{x}, \mathsf{w}, e)$*: Outputs the subset $\tilde{I}_e$, which corresponds to the locations of π_{I_e}, I_e, and w_e in z.*

[5] Recall that the algorithms for the actual setup, prover and verifier, are obtained by combining the algorithms in the construction with the commitment scheme Com, as described in Definition 3.5.

- $\widetilde{\mathsf{V}}'(\mathsf{crs}, \mathsf{x}, e, (\tilde{I}, \tilde{\pi}_{\tilde{I}}))$: *The verifier parses* $\tilde{\pi}_{\tilde{I}} = (I_e, \pi_{I_e}, w_e)$ *then verifies that*

$$\Phi_{\mathsf{crs},\mathsf{x},e}(I_e, \pi_{I_e}, w_e) = 1$$

where $\Phi_{\mathsf{crs},\mathsf{x},e}$ *is the Cook-Levin 3-CNF formula for* $C_{\mathsf{crs},\mathsf{x},e}$ *verification.*

We begin by showing that, if the underlying commitment scheme is extractable, then $\tilde{\Pi}_1^{\mathsf{Com}}$ is a trapdoor Σ-protocol.

Lemma 3.12. *Let* $\mathsf{Com} = (\mathsf{Gen}, \mathsf{Commit}, \mathsf{Verify}, \mathsf{Extract})$ *be a statistically binding extractable commitment scheme, and let* $\Pi^{\mathsf{Com}} = (\mathsf{Setup}', \mathsf{P}_1, \mathsf{P}_2, \mathsf{V}')$ *be commit-then-open* Σ*-protocol with soundness* $1/2$*. Then,* $\widetilde{\Pi}_1^{\mathsf{Com}}$ *from Construction 3.1 is a trapdoor* Σ*-protocol with:*

- $\mathsf{tdSetup}(1^\lambda)$: *Sample* $(\mathsf{pk}, \mathsf{td}) \leftarrow \mathsf{Com.Gen}(1^\lambda)$ *and* $\mathsf{crs}' \leftarrow \mathsf{Setup}'(1^\lambda, \mathsf{pk})$*, then output*

$$((\mathsf{crs}', \mathsf{pk}), \mathsf{td})$$

- $\mathsf{BadChallenge}(\mathsf{crs}, \mathsf{td}, \mathsf{x}, a)$: *Compute* $\tilde{\pi}' \leftarrow \mathsf{Extract}(\mathsf{td}, a)$*, and parse* $\tilde{\pi}' = (\pi', I_0, I_1, w_0, w_1) \in \{0,1,\perp\}^*$*. For every* $e \in \{0,1\}$*, if* $I_e \in \{0,1\}^*$*, set* $\tilde{\pi}'_e = (I_e, \pi'_{I_e}, w_e)$ *and otherwise set* $\tilde{\pi}'_e = \perp$*.*
  1. *If* $\tilde{\pi}'_0 \in \{0,1\}^*$ *and* $\Phi_{\mathsf{crs},\mathsf{x},0}(\tilde{\pi}'_0) = 1$*, output 0.*
  2. *If* $\tilde{\pi}'_1 \in \{0,1\}^*$ *and* $\Phi_{\mathsf{crs},\mathsf{x},1}(\tilde{\pi}'_1) = 1$*, output 1.*
  3. *Otherwise, output* $\perp$*.*

Proof. It is evident that, based on the statistical binding of Com, the transformation preserves the soundness of the protocol and that, based on the computational hiding of Com, it also preserves honest-verifier zero knowledge (the honest-verifier uses the simulator of Π^{Com} in a straight-forward manner and generates random commitments where necessary).

It is also clear that $\mathsf{tdSetup}(1^\lambda)$ outputs a common reference string identical to $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$. We therefore focus on proving correctness of $\mathsf{BadChallenge}$.

Let $\mathsf{x} \notin L$, and crs, td and a be such that $R_{\mathsf{crs},\mathsf{x}}(a) = e \neq \perp$ (where $R_{\mathsf{crs},\mathsf{x}} \in \mathcal{R}_\Sigma(\widetilde{\Pi}_1^{\mathsf{Com}})$ as defined in Definition 3.3). From definition of $R_{\mathsf{crs},\mathsf{x}}$, there exists $(\tilde{I}, \tilde{\pi}_{\tilde{I}})$ such that $\tilde{\mathsf{V}}(\mathsf{crs}, \mathsf{x}, a, e, (\tilde{I}, \tilde{\pi}_{\tilde{I}})) = 1$. From the statistical binding and correct extraction of Com, it necessarily holds that $\tilde{\pi}_{\tilde{I}} = \mathsf{Extract}(a_{\tilde{I}}) = \tilde{\pi}'_e$. Further, we have $\mathsf{V}'(\mathsf{crs}, \mathsf{x}, e, (\tilde{I}, \tilde{\pi}_{\tilde{I}})) = 1$ and, therefore, $\Phi_{\mathsf{crs},\mathsf{x},e}(\tilde{\pi}_{\tilde{I}}) = 1$ implying

$$\Phi_{\mathsf{crs},\mathsf{x},e}(\tilde{\pi}'_e) = 1 \tag{1}$$

On the other hand, since $R_{\mathsf{crs},\mathsf{x}}$ is a unique-output relation (due to Lemma 3.12 and Definition 3.3), then there exists no $(\tilde{I}, \tilde{\pi}_{\tilde{I}})$ such that $\tilde{\mathsf{V}}(\mathsf{crs}, \mathsf{x}, a, 1-e, (\tilde{I}, \tilde{\pi}_{\tilde{I}})) = 1$ and, in particular, this holds for $\tilde{\pi}'_{1-e}$. Therefore, if $\tilde{\pi}'_{1-e}$ is a valid opening of $a_{\tilde{I}}$ (with $\tilde{I}$ being the set of locations supposedly corresponding to $(I_{1-e}, \pi'_{I_{1-e}}, w_{1-e})$ in a), i.e. $\tilde{\pi}'_{1-e} = \mathsf{Extract}(a_{\tilde{I}})$, then

$$\Phi_{\mathsf{crs},\mathsf{x},1-e}(\tilde{\pi}'_{1-e}) = 0 \tag{2}$$

By combining (1) and (2), we obtain that $\mathsf{BadChallenge}(\mathsf{crs}, \mathsf{td}, \mathsf{x}, a) = e = R_{\mathsf{crs},\mathsf{x}}(a)$ and we finish. □

Having shown that the protocol is a trapdoor Σ-protocol, our goal now is to show that the trapdoor function $\mathsf{BadChallenge}$, which is specified in Lemma 3.12, has probabilistic representation as constant degree polynomials. Observe that, roughly speaking, $\mathsf{BadChallenge}$ is a composition of the extraction function, which we assume has a probabilistic constant-degree representation, and an evaluation of two CNF formulas. Since the protocol is a Σ-protocol, we show that, in fact, the randomized polynomials need to (probabilistically) evaluate only one of these formulas on the extracted value.

Thus, as a first step towards constructing efficient probabilistic constant-degree representation for $\mathsf{BadChallenge}$, we show how to evaluate CNF formulas using randomized polynomials.

Lemma 3.13 (k-CNF via Probabilistic Polynomials)**.** *Let $\ell, k, c \in \mathbb{N}$. For any k-CNF formula $\Phi : \{0,1\}^\ell \to \{0,1\}$, there exists a 2^{-c}-probabilistic representation by $c(k+1)$-degree polynomials $\mathfrak{P}_\Phi$.*

Proof. Fix a k-CNF formula Φ. For simplicity, we describe the distribution over polynomials $\mathfrak{P}_\Phi$ as a randomized algorithm.

$\mathfrak{P}_\Phi(x)$:

1. Let $(C_1, \ldots, C_N)$ be the clauses of Φ, where $N = O(\ell^k)$ is the maximal number of clauses in a k-CNF formula over n variables.
2. Compute $y = (\neg C_1(x) \| \ldots \| \neg C_N(x)) \in \{0,1\}^N$, which is the negation of the evaluation of the clauses of Φ on x (that is, $y[j] = \neg C_j(x)$).
3. Sample a random matrix $R \xleftarrow{\$} \mathbb{Z}_2^{c \times N}$, and compute $z = Ry \in \{0,1\}^c$.
4. Output $w = \neg(z[1] \wedge \cdots \wedge z[c])$.

Notice that the transformation $x \mapsto y$ is k-local and, therefore, can be evaluated using a k-degree polynomial. The transformation $y \mapsto z$ is just a randomized linear function, and $z \mapsto w$ is c-degree. Overall, we can describe the computation of $\mathfrak{P}_\Phi$ as a distribution over polynomials of degree $c \cdot (k+1)$.

It remains to show that $\mathfrak{P}_\Phi$ indeed computes Φ with high probability. If $\Phi(x) = 1$, then $y = 0^N$ and, consequently, $z = 0^c$ and $w = 1$ for any choice of R. If $\Phi(x) = 0$, then $y \neq 0^N$ and, since R is random, z is a uniformly random vector. Thus, we can bound $\Pr[w = 1] = \Pr[z = 0^c] = 2^{-c}$. □

We now use Lemma 3.13, and the assumption that $\mathsf{Extract}$ has probabilistic constant-degree representation, to obtain such a representation for $\mathsf{BadChallenge}$.

Lemma 3.14. *Let $c, c' \in \mathbb{N}$ be arbitrary constants, and let $0 < \epsilon(\lambda) < 1/\ell(\lambda)$. Let Com be an extractable commitment scheme where, for any td, the extraction function $\mathsf{Extract}(\mathsf{td}, \cdot)$ has an ϵ-probabilistic representation by c-degree polynomials. Consider the protocol $\tilde{\Pi}^{\mathsf{Com}}$ from Construction 3.1. Then, the function*

$$f_{\mathsf{crs},\mathsf{td},\mathsf{x}}(a) = \mathsf{BadChallenge}(\mathsf{crs}, \mathsf{td}, \mathsf{x}, a),$$

as defined in Lemma 3.12, has ϵ'-probabilistic representation by $6cc'$-degree polynomials, with $\epsilon' = \ell \cdot \epsilon + 2^{-c'}$.

Proof. Let $\mathfrak{P}_{\mathsf{td}}$ be the efficient ϵ-probabilistic representation of $\mathsf{Extract}(\mathsf{td}, \cdot)$ by c-degree polynomials. We now show a probabilistic representation of $f_{\mathsf{crs},\mathsf{td},\mathsf{x}}$ by c'-degree polynomials, denoted by $\mathfrak{P}_{\mathsf{crs},\mathsf{td},\mathsf{x}}$. For simplicity, we describe $\mathfrak{P}_{\mathsf{crs},\mathsf{td},\mathsf{x}}$ as a randomized algorithm.

$\mathfrak{P}_{\mathsf{crs},\mathsf{td},\mathsf{x}}(a)$:

1. Sample $P_{\mathsf{td}} \xleftarrow{\$} \mathfrak{P}_{\mathsf{td}}^{\ell}$, and compute $\tilde{z} = P_{\mathsf{td}}(a)$.
2. Parse $\tilde{z} = (z, I_0, I_1, w_0, w_1)$ and compute $\tilde{z}_1 = (I_1, z_{I_1}, w_1)$.
3. Denote by $\mathfrak{P}_{\Phi}$ the $2^{-c'}$-probabilistic representation of $\Phi_{\mathsf{crs},\mathsf{x},1}$ by $3c'$-degree polynomials (due to Lemma 3.13). Sample $P_{\Phi} \xleftarrow{\$} \mathfrak{P}_{\Phi}$, then output $b = P_{\Phi}(\tilde{z}_e)$.

We know that P_{td} and P_{Φ} are random polynomials of degrees c and $3c'$, respectively. It is also clear that, from the representation of I_1 as a matrix, then the transformation $(I_1, z) \mapsto z_{I_1}$ and, therefore, step 2 of $\mathfrak{P}_{\mathsf{crs},\mathsf{td},\mathsf{x}}$, can be described using a fixed 2-degree polynomial. We conclude that $\mathfrak{P}_{\mathsf{crs},\mathsf{td},\mathsf{x}}$ can be described as a distribution over $6cc'$-degree polynomials.

It remains to show that $\mathfrak{P}$ probabilistically computes $f_{\mathsf{crs},\mathsf{td},\mathsf{x}}$. From the correctness of $\mathfrak{P}_{\mathsf{td}}$ and following Definition 3.7, if $\tilde{\pi}'_1 = \mathsf{Extract}_{\mathsf{td}}(a_{I_1}) \in \{0,1\}^*$, then

$$\forall i \in I_1, \Pr[\tilde{\pi}'_i \neq \tilde{z}_i] \leq \epsilon$$

Applying union bound on the above, we get that $\Pr[\tilde{\pi}' \neq \tilde{z}] \leq |I_1| \cdot \epsilon \leq \ell \cdot \epsilon$.

Now, conditioning on $\tilde{\pi}' = \tilde{z}$, and from the correctness of $\mathfrak{P}_{\Phi}$, we get that $\Pr[b \neq \Phi_{\mathsf{crs},\mathsf{x},1}(\tilde{\pi}'_1)] \leq 2^{-c'}$ and, therefore, overall, we get that

$$\begin{aligned} &\Pr_{P \xleftarrow{\$} \mathfrak{P}_{\mathsf{crs},\mathsf{td},\mathsf{x}}} [P(a) \neq \Phi_{\mathsf{crs},\mathsf{x},1}(\tilde{\pi}'_1)] \\ &\leq \Pr[\tilde{\pi}' \neq \tilde{z}] + \Pr[P(a) \neq \Phi_{\mathsf{crs},\mathsf{x},1}(\tilde{\pi}'_1) \mid \tilde{\pi}' = \tilde{z}] \\ &\leq \ell \cdot \epsilon + 2^{-c'} \end{aligned} \tag{3}$$

Now, if $f_{\mathsf{crs},\mathsf{td},\mathsf{x}}(a) = 1$, then it must be the case that $\tilde{\pi}'_1 \in \{0,1\}^*$ and $\Phi_{\mathsf{crs},\mathsf{x},1}(\tilde{\pi}'_1) = 1$, and therefore, from (3), $P(a) = 1$ with the required probability. Otherwise, if $f_{\mathsf{crs},\mathsf{td},\mathsf{x}}(a) = 0$, then $\tilde{\pi}'_0 \in \{0,1\}^*$ and $\Phi_{\mathsf{crs},\mathsf{x},0}(\tilde{\pi}'_0) = 1$. Since $\widetilde{\Pi}^{\mathsf{Com}}$ is a Σ-protocol and $\mathcal{R}_{\Sigma}(\widetilde{\Pi}^{\mathsf{Com}})$ is unique output (Lemma 3.12), then there exist no $\tilde{z}_1 \in \{0,1\}^*$ such that $\Phi_{\mathsf{crs},\mathsf{x},1}(\tilde{z}_1) = 1$ and, therefore, $P(a) = 0$ with the required probability. This completes the proof. □

Combining Lemmas 3.12 and 3.14, we have so far proven Theorem 3.10 for the special case of $m = 1$. To derive the theorem for the general case, consider the protocol $\widetilde{\Pi}^{\mathsf{Com}}$ that consists of m parallel repetitions of $\widetilde{\Pi}_1^{\mathsf{Com}}$. Parallel repetition preserves honest-verifier zero knowledge and the Σ-protocol property ($\mathcal{R}_{\Sigma}$ being unique-output) and, consequently, amplifies soundness to 2^{-m}. Further, if $\widetilde{\Pi}_1^{\mathsf{Com}}$ is a trapdoor Σ-protocol with $\mathsf{tdSetup}$ and $\mathsf{BadChallenge}$, then $\widetilde{\Pi}^{\mathsf{Com}}$ is a trapdoor Σ-protocol with $\mathsf{tdSetup}$ and $\mathsf{BadChallenge}^m$, where $\mathsf{BadChallenge}^m(\mathsf{crs}, \mathsf{td}, \mathsf{x}, a_1, \ldots, a_m)$ computes $e_i = \mathsf{BadChallenge}(\mathsf{crs}, \mathsf{td}, \mathsf{x}, a_i)$ for all $i \in [m]$ then outputs $(e_1, \ldots, e_m)$ if $\forall i\ e_i \in \{0,1\}$ and outputs $\perp$ otherwise. By Definition 3.7, if $\mathsf{BadChallenge}$ has ϵ'-probabilistic $6cc'$-degree representation, then so does $\mathsf{BadChallenge}^m$.

Hence, the proof of Theorem 3.10 is complete.

4 CI through Probabilistic Representation

In this section, we show that if a function class $\mathcal{F}$ has a probabilistic representation by a potentially simpler class $\mathcal{C}$ (see Definition 3.7) then CI for relations searchable by $\mathcal{F}$ can be reduced to CI for a class of relations that are "approximated" by $\mathcal{C}$. This is the first step we make towards constructing CI hash, as required by Corollary 3.11, from standard assumptions.

4.1 Approximable Relations and CI-Apx

We start by defining the notion of *approximable relations* and a related special case of correlation intractability, CI-Apx.

Definition 4.1 (CI-Apx)**.** *Let $\mathcal{C} = \{\mathcal{C}_\lambda : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}\}$ be a function class and let $0 < \epsilon < 1$. For every $C \in \mathcal{C}$, we define the relation* ϵ-approximable by C *as follows*

$$\mathcal{R}^{\epsilon}_{C} = \{(x, y) \in \{0,1\}^n \times \{0,1\}^m \mid \Delta(y, C(x)) \leq \epsilon m\}$$

A hash family that is CI for all relations $\{\mathcal{R}^{\epsilon}_{C} \mid C \in \mathcal{C}\}$ is said to be CI-Apx$_\epsilon$ *for $\mathcal{C}$.*

4.2 From CI-Apx for $\mathcal{C}$ to CI for $\mathcal{F}$

We now state and prove the following general theorem.

Theorem 4.2. *Let $\mathcal{F}$ be a function class that has an ϵ-probabilistic representation by $\mathcal{C}$. If $\mathcal{H}$ is* CI-Apx$_{2\epsilon}$ *hash for $\mathcal{C}$, then $\mathcal{H}$ is CI for relations searchable by $\mathcal{F}$ (i.e. ϵ-probabilistically searchable by $\mathcal{C}$).*

4.2.1 Proof of Theorem 4.2.

Suppose $\mathcal{R}$ is searchable by $\mathcal{F} : \{0,1\}^n \to \{0,1\}^m$. Fix some $R \in \mathcal{R}$ and consider its corresponding search function $f \in \mathcal{F}$. Let $\mathfrak{C}_f$ be the ϵ-probabilistic representation of f by $\mathcal{C}$.

We start by defining a game $\mathsf{Game}_0(\mathcal{A})$ against an adversary $\mathcal{A}$ as follows.

$\mathsf{Game}_0(\mathcal{A})$:

1. $\mathsf{k} \xleftarrow{\$} \mathsf{Sample}(1^\lambda)$.
2. $x \leftarrow \mathcal{A}(\mathsf{k})$.
3. Output 1 if and only if $f(x) \neq \perp$ and $\mathsf{Hash}_\mathsf{k}(x) = f(x)$.

It is clear that the probability of an adversary $\mathcal{A}$ to win Game_0 upper bounds the probability he breaks the correlation intractability of $\mathcal{H}$ for R (immediate from Definition 3.2). Our goal, then, is to show that for any PPT adversary $\mathcal{A}$, there exists a negligible function negl such that $\Pr[\mathsf{Game}_0(\mathcal{A}) = 1] < \mathsf{negl}(\lambda)$.

We now reduce Game_0 to Game_1, which is defined below.

$\mathsf{Game}_1(\mathcal{A})$:

1. $C \xleftarrow{\$} \mathfrak{C}_f$.

2. $\mathsf{k} \xleftarrow{\$} \mathsf{Sample}(1^\lambda)$.
3. $x \leftarrow \mathcal{A}(\mathsf{k})$.
4. Output 1 if and only if $\Delta(\mathsf{Hash}_\mathsf{k}(x), C(x)) \leq 2\epsilon m$.

Lemma 4.3. *For any (possibly unbounded) adversary $\mathcal{A}$, there exists a negligible function* negl*, such that*

$$\Pr[\mathsf{Game}_0(\mathcal{A}) = 1] \leq \Pr[\mathsf{Game}_1(\mathcal{A}) = 1] + \mathsf{negl}(\lambda)$$

Proof. The proof is derived from the fact that C in Game_1 is sampled independently of the adversary's choice x and from Lemma 3.8, as follows.

$$\begin{aligned}
\Pr[\mathsf{Game}_0(\mathcal{A}) = 1] &= \Pr[f(x) \neq \bot \wedge \mathsf{Hash}_\mathsf{k}(x) = f(x)] \\
&\leq \Pr_{C \xleftarrow{\$} \mathfrak{C}_f}[f(x) \neq \bot \wedge \Delta(f(x), C(x)) > 2\epsilon m] \\
&+ \Pr_{C \xleftarrow{\$} \mathfrak{C}_f}[f(x) \neq \bot \wedge \Delta(\mathsf{Hash}_\mathsf{k}(x), f(x)) \leq 2\epsilon m] \\
&\leq \Pr[\mathsf{Game}_1(\mathcal{A}) = 1] + \mathsf{negl}(\lambda)
\end{aligned}$$

□

To complete the proof of Theorem 4.2, we show that Game_1 is hard to win with non-negligible probability, based on the correlation intractability of $\mathcal{H}$ for relations 2ϵ-approximable $\mathcal{C}$.

Lemma 4.4. *If $\mathcal{H}$ is* CI-Apx$_{2\epsilon}$ *for $\mathcal{C}$ then, for any $f \in \mathcal{F}$ and any PPT adversary $\mathcal{A}$, there exists a negligible function such that*

$$\Pr[\mathsf{Game}_1(\mathcal{A}) = 1] < \mathsf{negl}(\lambda)$$

Proof. Assume towards contradiction there exists $f \in \mathcal{F}$ and $\mathcal{A}$ for which the above does not hold, namely $\Pr[\mathsf{Game}_1(\mathcal{A}) = 1] > 1/\mathsf{poly}(\lambda)$. Then, there exists some fixed $C \in \mathfrak{C}_f$ such that $\Pr[\mathsf{Game}_1^C(\mathcal{A}) = 1] > 1/\mathsf{poly}(\lambda)$, where Game_1^C is defined as Game_1 with C being fixed (rather than sampled from $\mathfrak{C}_f$). From definition, such an adversary breaks the CI-Apx$_{2\epsilon}$ of $\mathcal{H}$ for C. □

We conclude the proof of the theorem by combining Lemmas 4.3 and 4.4.

5 CI-Apx from Trapdoor Hash

Having shown in the previous section that CI-Apx is a useful notion to obtain CI for a function class that has a simple probabilistic representation, we now show how to construct, from rate-1 trapdoor hash for any function class $\mathcal{C}$ [DGI+19], an CI-Apx hash for $\mathcal{C}$. In fact, in our proof of CI, we require that the underlying TDH scheme satisfies the following stronger notion of correctness.

Definition 5.1 (Enhanced Correctness for TDH)**.** *We say that a (rate-1) trapdoor hash scheme* TDH *for $\mathcal{C} = \{\mathcal{C}_n : \{0,1\}^n \to \{0,1\}\}$ has* enhanced $(1-\tau)$-correctness *for $\tau := \tau(\lambda) < 1$ if it satisfies the following property:*

- ***Enhanced Correctness:*** *There exists a negligible function* $\mathsf{negl}(\lambda)$ *such that the following holds for any* $\lambda, n, \in \mathbb{N}$*, any* $\mathsf{h} \in \{0,1\}^{\eta(\lambda)}$*, and any function* $C \in \mathcal{C}_n$*,*

$$\Pr[\forall x \in \{0,1\}^n : \mathsf{H}(\mathsf{hk}, x) = \mathsf{h},\ \mathsf{e} + \mathsf{e}' = C(x) \mod 2] \geq 1 - \tau - \mathsf{negl}(\lambda)$$

where $\mathsf{hk} \leftarrow \mathsf{S}(1^\lambda, 1^n)$*,* $(\mathsf{ek}, \mathsf{td}) \leftarrow \mathsf{G}(\mathsf{hk}, C)$*,* $\mathsf{e} \leftarrow \mathsf{E}(\mathsf{ek}, x)$ *and* $\mathsf{e}' \leftarrow \mathsf{D}(\mathsf{td}, \mathsf{h})$*.*

Formally, we get the following result.

Theorem 5.2. *Assume there exists rate-1 trapdoor hash scheme* TDH *for* $\mathcal{C} = \{\mathcal{C}_n : \{0,1\}^n \to \{0,1\}^m\}$ *with enhanced* $(1-\tau)$*-correctness where the hash length is* $\eta := \eta(\lambda)$*. Then, for any* ϵ *s.t.* $\epsilon + \tau < \epsilon_0$ *(for some fixed universal constant* ϵ_0*), there exists a polynomial* $m_{\epsilon,\eta,\tau}(\lambda) = O((\eta+\lambda)/\tau + \log(1/\epsilon))$ *such that, for every polynomial* $m > m_\epsilon$*, there exists a CI-Apx*$_\epsilon$ *hash family for* $\mathcal{C}$ *with output length* $m(\lambda)$*.*[6]

Recalling Corollary 3.11, and using the result from Section 4, obtaining CI-Apx for constant-degree functions is sufficient for our purpose of constructing NIZK. To instantiate Theorem 5.2 for constant-degree functions from standard assumption, we use the following result of Döttling et al. [DGI+19].

Theorem 5.3 (Trapdoor Hash from Standard Assumptions [DGI+19])**.** *For any constant* $c \in \mathbb{N}$ *and arbitrarily small* $\tau := \tau(\lambda) = 1/\mathsf{poly}(\lambda)$*, there exists a rate-1 trapdoor hash scheme, for c-degree polynomials over* $\mathbb{Z}_2$*, with enhanced (1-*τ*)-correctness and function privacy under the DDH/QR/DCR/LWE assumption*[7]*.*

We note some gaps between the result from [DGI+19] and the theorem above. First, the aforementioned work does not contain a DDH-based construction for constant-degree functions but rather only for "index functions". Second, all known constructions are not proven to have enhanced correctness. In Appendix A, we show how to close these gaps by simple adjustments to the constructions and proofs from [DGI+19].

Combining Theorems 5.2 and 5.3, we obtain.

Corollary 5.4. *Let* $c \in \mathbb{N}$*. There exists a constant* $\epsilon > 0$ *such that, for any sufficiently large polynomial* $m := m(\lambda)$*, there exists a programmable correlation intractable hash family with output length* m *for all relations* ϵ*-approximable by c-degree polynomials over* $\mathbb{Z}_2$*.*

5.1 The Hash Family

We now present our construction of CI-Apx from rate-1 TDH. We note that we do not use the full power of a TDH. Specifically, we do not require that the decoding algorithm is efficient and, further, we do not use input privacy (as defined in [DGI+19]).

Construction 5.1 (Correlation Intractability from TDH)**.** *Let* $n := n(\lambda)$ *and* $m := m(\lambda)$ *be polynomials in the security parameter, and let* $\epsilon := \epsilon(\lambda) < 0.32$*. Let* $\mathcal{C} : \{0,1\}^n \to \{0,1\}$ *be a function class and let* $\mathsf{TDH} = (\mathsf{S}, \mathsf{G}, \mathsf{H}, \mathsf{E}, \mathsf{D})$ *be a rate-1 trapdoor hash scheme for* $\mathcal{C}$*. Our construction of CI-Apx*$_\epsilon$ *hash for* $\mathcal{C}$ *consists of the following algorithms.*

[6] In fact, as implicitly implied by the proof of the theorem, our construction satisfies the stronger notion of somewhere statistical CI for the corresponding hamming-ball relations [CCH+19]. However, applying Theorem 4.2 on the construction does not preserve this property.

[7] The error probability in the QR, DCR, and LWE constructions is even negligible.

- $\mathsf{Sample}(1^\lambda)$: *Sample* $\mathsf{hk} \xleftarrow{\$} \mathsf{S}(1^\lambda)$ *and, for every* $i \in [m]$, $(\mathsf{ek}_i, \mathsf{td}_i) \xleftarrow{\$} \mathsf{G}(\mathsf{hk}, C_0)$ *for an arbitrary fixed* $C_0 \in \mathcal{C}$*, and a uniformly random* $r \xleftarrow{\$} \{0,1\}^m$*, then output*

$$\mathsf{k} = ((\mathsf{ek}_1, \ldots, \mathsf{ek}_m), r)$$

- $\mathsf{Hash}(\mathsf{k}, x)$: *The hash of an input* $x \in \{0,1\}^n$ *under key* $\mathsf{k} = ((\mathsf{ek}_i)_{i\in[m]}, r)$ *is computed as follows*

$$\mathsf{h} = \mathsf{E}((\mathsf{ek}_1, \ldots, \mathsf{ek}_m), x) + r \mod 2$$

5.2 Proof of Theorem 5.2

Programmability of the construction is trivial and, therefore, we focus on proving CI.

Fix some $C = (C_1, \ldots, C_m) \in \mathcal{C}^m$ and consider the relation ϵ-probabilistically searchable by C, R^ϵ_C. The advantage of an adversary $\mathcal{A}$ in breaking the CI for R^ϵ_C is demonstrated in his advantage in winning in the following game.

$\mathsf{Game}_0(\mathcal{A})$:

1. $\mathsf{k} \xleftarrow{\$} \mathsf{Sample}(1^\lambda)$.
2. $x \leftarrow \mathcal{A}(\mathsf{k})$.
3. Output 1 if and only if $\Delta(\mathsf{Hash}_\mathsf{k}(x), C(x)) \leq 2\epsilon m$.

To show $\Pr[\mathsf{Game}_0(\mathcal{A}) = 1] < \mathsf{negl}(\lambda)$, we define a different game, Game_1, in which we switch the encoding keys $(\mathsf{ek}_1, \ldots, \mathsf{ek}_m)$ in k to encoding keys corresponding to the functions $C_1, \ldots, C_m$ (rather than C_0).

$\mathsf{Game}_1(\mathcal{A})$:

1. Sample $\mathsf{hk} \leftarrow \mathsf{S}(1^\lambda, 1^n)$ and $(\mathsf{ek}'_i, \mathsf{td}'_i) \leftarrow \mathsf{G}(\mathsf{hk}, C_i)$ for every $i \in [m]$. Sample a uniform $r \xleftarrow{\$} \{0,1\}^m$, then set $\mathsf{k} = ((\mathsf{ek}'_1, \ldots, \mathsf{ek}'_m), r)$.
2. $x \leftarrow \mathcal{A}(\mathsf{k})$.
3. Output 1 if and only if $\Delta(\mathsf{Hash}_\mathsf{k}(x), C(x)) \leq 2\epsilon m$.

We claim that, based on the function privacy of the underlying trapdoor hash, we may reduce Game_0 to Game_1.

Lemma 5.5. *Under the function privacy of* TDH*, for any PPT adversary* $\mathcal{A}$*, there exists a negligible function* negl *such that*

$$\Pr[\mathsf{Game}_0(\mathcal{A}) = 1] \leq \Pr[\mathsf{Game}_1(\mathcal{A}) = 1] + \mathsf{negl}(\lambda)$$

Proof. Assume towards contradiction that there exists an adversary $\mathcal{A}$ for which the above does not hold.

We use $\mathcal{A}$ to construct an adversary $\mathcal{A}_\mathsf{TDH}$ that distinguishes between

$$(\mathsf{hk}, (\mathsf{ek}_1, \ldots, \mathsf{ek}_m)) \quad \text{and} \quad (\mathsf{hk}, (\mathsf{ek}'_1, \ldots, \mathsf{ek}'_m))$$

, where $\mathsf{hk} \leftarrow \mathsf{S}(1^\lambda, 1^n)$, $\mathsf{ek}_i \leftarrow \mathsf{G}(\mathsf{hk}, C_0)$ and $\mathsf{ek}'_i \leftarrow \mathsf{G}(\mathsf{hk}, C_{f_i})$ (for every $i \in [m]$), with non-negligible advantage. Such an adversary breaks the function privacy of TDH via a standard hybrid argument.

On input $(ek_1, \ldots, ek_m, C)$, $\mathcal{A}_{\mathsf{TDH}}$ simply calls $x \leftarrow \mathcal{A}((ek_1, \ldots, ek_m), r))$, and outputs 1 iff $\Delta(\mathsf{Hash}_\mathsf{k}(x), C(x)) \leq 2\epsilon m$. It holds that

$$\begin{aligned}
&|\Pr[\mathcal{A}_{\mathsf{TDH}}(\mathsf{hk}, (\mathsf{ek}_1, \ldots, \mathsf{ek}_m)) = 1] - \Pr[\mathcal{A}_{\mathsf{TDH}}(\mathsf{hk}, (\mathsf{ek}'_1, \ldots, \mathsf{ek}'_m)) = 1]| \\
&= |\Pr[\mathsf{Game}_1(\mathcal{A}) = 1] - \Pr[\mathsf{Game}_2(\mathcal{A}) = 1]| \\
&\geq 1/\mathsf{poly}(\lambda)
\end{aligned}$$

□

Lastly, we show that Game_1 is statistically hard to win. This, together with Lemma 5.6, implies Theorem 5.2.

Lemma 5.6. *For any (possibly unbounded) adversary $\mathcal{A}$, there exists a negligible function negl such that*

$$\Pr[\mathsf{Game}_1(\mathcal{A}) = 1] < \mathsf{negl}(\lambda)$$

Proof. The impossibility is based on a simple counting argument. More specifically, it suffices to show there exists a negligible function negl such that

$$\Pr_{\mathsf{k}}[\exists x : \ \Delta(\mathsf{Hash}_\mathsf{k}(x), C(x)) \leq 2\epsilon m] < \mathsf{negl}(\lambda)$$

where k is sampled as in Game_2. We denote the above event by

$$\mathsf{Bad} = [\exists x : \ \Delta(\mathsf{Hash}_\mathsf{k}(x), C(x)) \leq \epsilon m]$$

and observe that

$$\begin{aligned}
\Pr[\mathsf{Bad}] &= \Pr_{\mathsf{k}}[\exists x, z \in \{0,1\}^m : \ |z| \leq 2\epsilon m \ \wedge \ C(x) + z = \mathsf{Hash}_\mathsf{k}(x) \mod 2] \\
&= \Pr_{r}[\exists x, z \in \{0,1\}^m : \ |z| \leq 2\epsilon m \wedge \ C(x) + z + \mathsf{E}(\mathsf{ek}, x) = r \mod 2]
\end{aligned}$$

From the enhanced $(1-\tau)$-correctness of TDH and using Chernoff bound, it holds that for every $\mathsf{h} \in \{0,1\}^\eta$,

$$\Pr[\forall x : \mathsf{H}(\mathsf{hk}, x) = \mathsf{h}, \quad \Delta(\mathsf{E}(\mathsf{ek}, x) + C(x), \mathsf{D}(\mathsf{td}, \mathsf{h})) > 2\tau m] \leq e^{-\tau m/3}$$

Applying union bound over all $\mathsf{h} \in \{0,1\}^\eta$, we get that

$$\Pr[\forall x, \quad \Delta(\mathsf{E}(\mathsf{ek}, x) + C(x), \mathsf{D}(\mathsf{td}, \mathsf{H}(\mathsf{hk}, x))) > 2\tau m] < e^{\eta - \tau m} = \mathsf{negl}(\lambda)$$

Therefore, letting $\mathsf{h}_x = \mathsf{H}(\mathsf{hk}, x)$, we get

$$\begin{aligned}
\Pr[\mathsf{Bad}] < &\Pr_{r}[\exists x, z' \in \{0,1\}^m : \ |z'| \leq 2(\epsilon + \tau)m \ \wedge \ \mathsf{D}(\mathsf{td}, \mathsf{h}_x) + z' = r \mod 2] \\
&+ \mathsf{negl}(\lambda)
\end{aligned}$$

where $r \xleftarrow{\$} \{0,1\}^m$. Define $\epsilon' := 2(\epsilon + \tau)$ and, for any fixed hk and td, let

$$Y = \{y = \mathsf{D}(\mathsf{td}, \mathsf{h}_x) + z' \mod 2 \mid x \in \{0,1\}^n, z' \in \{0,1\}^m \ \text{ s.t. } \ |z'| \leq \epsilon' m\}$$

and notice that the above implies $\Pr[\mathsf{Bad}] < \Pr_r[r \in Y] + \mathsf{negl}(\lambda) = 2^{-m}|Y| + \mathsf{negl}(\lambda)$. Thus, it suffices to show that $2^{-m}|Y|$ is negligible.

Clearly, $\mathsf{D}(\mathsf{td}, \mathsf{h}_x)$ can take at most 2^η values over all possible x. Further, we can bound

$$|\{z' \in \{0,1\}^m \mid |z'| \leq \epsilon' m\}| = \sum_{i=1}^{\epsilon' m} \binom{m}{i} \leq \sum_{i=1}^{\epsilon' m} \left(\frac{me}{i}\right)^i \leq (e/\epsilon')^{\epsilon' m + 1}$$

and consequently, if ϵ' is a (universally) sufficiently small constant, and $m \geq (\lambda + \eta + \log(e/\epsilon'))/(1 - \epsilon' \log(e/\epsilon')) = O((\eta + \lambda)/\tau + \log(1/\epsilon))$, we get

$$2^{-m}|Y| \leq 2^{-m}(e/\epsilon)^{\epsilon' m + 1} 2^\eta < 2^{(\epsilon' \log(e/\epsilon') - 1)m + \log(e/\epsilon') + \eta} < 2^{-\lambda}$$

and hence we finish. □

6 Commitment with Linear Approximate Extraction

In this section we prove that, under the standard decisional LPN assumption, there exists an extractable commitment scheme that satisfies the conditions set by Corollary 3.11, and can be essentially used, together with a suitable CI hash, to obtain NIZK for NP.

Theorem 6.1. *Let $0 < c < 1$ be an arbitrary constant and $\ell := \ell(\lambda)$ be an arbitrarily large polynomial. There exists, under the $(n, 2n, n^{-(1+c)/2})$-DLPN assumption, a statistically-binding extractable commitment scheme where, for any td, the function $f_{\mathsf{td}} = \mathsf{Extract}(\mathsf{td}, x)$ has efficient $(1/\ell)$-probabilistic representation by linear functions.*

6.1 The Commitment Scheme

Our construction is heavily inspired by the public key encryption of Damgård and Park [DP12].

Construction 6.1 (Extractable Bit Commitment from LPN)**.** *Let $k := k(\lambda)$, $n := n(\lambda)$ be sufficiently large polynomials such that $k = \Omega(n)$ and let $m = 2n$. Let $\tau := \tau(\lambda) = n^{-(1+c)/2}$ for some constant $0 < c < 1$ and define $\epsilon := \epsilon(\lambda) = 16n^{-c}$ accordingly. Our construction of a lossy extractable bit commitment scheme consists of the following algorithms:*

- $\mathsf{Gen}(1^\lambda)$*: Sample* $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_2^{m \times n}$, $\mathbf{S} \xleftarrow{\$} \mathbb{Z}_2^{n \times k}$ *and* $\mathbf{E} \xleftarrow{\$} \mathsf{Ber}_\tau^{m \times k}$*, and set* $\mathbf{B} = \mathbf{AS} + \mathbf{E}$*, then output*

$$\mathsf{k} = (\mathbf{A}, \mathbf{B}) \qquad\qquad \mathsf{td} = \mathbf{S}$$

- $\mathsf{Commit}(\mathsf{pk}, \mathsf{x}; r)$*: Parse r as* $\mathbf{r} \xleftarrow{\$} \mathsf{Ber}_\tau^m$ *and let* $\mathbf{x} = \mathsf{x}^\ell \in \mathbb{Z}_2^k$*. Compute* $\mathbf{u} = \mathbf{rA}$ *and* $\mathbf{c} = \mathbf{rB} + \mathbf{x}$*, then output*

$$\mathsf{com} = (\mathbf{u}, \mathbf{c}) \tag{4}$$

- $\mathsf{Verify}(\mathsf{pk}, \mathsf{com}, \mathsf{x}, r)$*: Parse r as* $\mathbf{r} \in \mathbb{Z}_2^m$*, then output 1 if and only if*

$$|\mathbf{r}| \leq 2\tau m \ \wedge \ \mathsf{Commit}(\mathsf{pk}, \mathsf{x}; r) = \mathsf{com}$$

- $\mathsf{Extract}(\mathsf{td}, \mathsf{com})$*: Parse* $\mathsf{td} = \mathbf{S} \in \mathbb{Z}_2^{n \times k}$ *and* com *as in Equation (4), then output*

$$\mathsf{x}' = \mathsf{Majority}_\epsilon(\mathbf{uS} + \mathbf{c})$$

where $\mathsf{Majority}_\epsilon : \{0,1\}^k \to \{0,1\}$ *is the* majority with ϵ-promise *function, which is defined as follows:*

$$\mathsf{Majority}_\epsilon(\mathbf{v}) = \begin{cases} 0 & |\mathbf{v}| \leq \epsilon k \\ 1 & |\mathbf{v}| \geq (1-\epsilon)k \\ \perp & \textit{otherwise} \end{cases}$$

6.2 Proof of Theorem 6.1

The correctness of the construction immediately follows from applying Chernoff bound on the weight of an honestly sampled randomness $\mathbf{r}$. A proof that the commitment scheme is hiding, based on the $(n, 2n, n^{-(1+c)/2})$-DLPN assumption is identical to the proof of security for the public key encryption from [DP12]. We complete the proof of Theorem 6.1 through a series of lemmas, that show that the scheme has correct extraction and, further, that the extraction function has a probabilistic linear representation.

Lemma 6.2. *Let* $k, n, m = 2n$ *and* $\tau = n^{-(1+c)/2}$ *be the parameters from Construction 6.1, and define* $\epsilon := \epsilon(\lambda) = 16n^{-c}$ *accordingly. Then, there exists a negligible function* negl *such that the following holds for* $\mathbf{E} \xleftarrow{\$} \mathsf{Ber}_\tau^{m \times k}$

$$\Pr[\exists \mathbf{r} \ s.t. \ \ |\mathbf{r}| \leq 4\tau m \ \wedge \ |\mathbf{rE}| > \epsilon k] < \mathsf{negl}(\lambda)$$

Proof. Denote by $\mathbf{e}_1, \ldots, \mathbf{e}_m \in \mathbb{Z}_2^k$ the rows of $\mathbf{E}$. We first tail-bound the weight of each row using Chernoff and get $\Pr[|\mathbf{e}_i| > 2\tau k] < 2^{-\Omega(\tau k)}$. Using union bound, we imply that $\Pr[\exists i \ \text{ s.t. } \ |\mathbf{e}_i| > 2\tau k] < m \cdot 2^{-\Omega(\tau k)}$, which is negligible in τk and, therefore, in λ (assuming $k = \Omega(n)$). Assuming this does not happen, it is easy to see that, for any $\mathbf{r} \in \{0,1\}^m$ with $|\mathbf{r}| \leq 2\tau m$, it holds that $|\mathbf{rE}| \leq (4\tau m)(2\tau k) = 8\tau^2 mk = \epsilon k$. □

Lemma 6.3 (Extraction)**.** *The bit commitment scheme from Construction 6.1 has correct extraction and, therefore, is also statistically binding (due to Remark 2.5).*

Proof. For any $\mathsf{x} \in \{0,1\}$, it holds that

$$\mathsf{Extract}(\mathsf{td}, \mathsf{Commit}(\mathsf{pk}, \mathsf{x}; r)) = \mathsf{Majority}_\epsilon(\mathbf{uS} + \mathbf{c}) = \mathsf{Majority}_\epsilon(\mathbf{rE} + \mathbf{x})$$

where $\mathbf{x} = x^k$, $\mathbf{E} \xleftarrow{\$} \mathsf{Ber}_\tau^{m \times k}$ and $\mathbf{r} \xleftarrow{\$} \mathsf{Ber}_\tau^m$. Using Chernoff, we bound $\Pr[|\mathbf{r} \leq 4\tau m|] > 1 - \mathsf{negl}(\lambda)$, and therefore, by applying Lemma 6.2, we get that $\Pr[|\mathbf{rE}| \leq \epsilon k] > 1 - \mathsf{negl}(\lambda)$ for $\epsilon = 16n^{-c}$. Conditioning on such an event it holds that $\mathsf{Majority}_\epsilon(\mathbf{rE} + \mathbf{x}) = \mathsf{x}$, and therefore extraction is correct with all but negligible probability. □

Lemma 6.4. *Let* $\ell := \ell(\lambda)$ *be an arbitrarily large polynomial and set* $\epsilon(\lambda) = 1/\ell(\lambda)$*. Let* $\mathsf{Com} = (\mathsf{Gen}, \mathsf{Commit}, \mathsf{Verify}, \mathsf{Extract})$ *be the bit commitment scheme from Construction 6.1 with a sufficiently large security parameter* n *such that* $16n^{-c} \leq \epsilon$*. Then, for any* td*, the function* $f_{\mathsf{td}}(x)$ *has an efficient* ϵ*-probabilistic representation by linear functions.*

Proof. For simplicity of exposition, we describe the distribution $\mathfrak{L}_{\mathsf{td}}$ as a randomized algorithm which implicitly defines a distribution over linear functions in a straight-forward manner.

$\mathfrak{L}_{\mathsf{td}}(a)$:

1. Sample a random column from the secret $\mathbf{S}$, and denote it by $\mathbf{s} \in \mathbb{Z}_2^n$.
2. Parse $a = (\mathbf{u}, \mathbf{c})$ then output $x = \mathbf{us} + \mathbf{c}$.

It is clear that the above algorithm can be described as a randomized linear function in a. Further, notice that if $\mathsf{Extract}(\mathsf{td}, a) \neq \perp$ then $|\mathbf{uS} + \mathbf{c}| \leq \epsilon k$ or $\geq (1-\epsilon)k$ and, therefore, $\Pr[\mathbf{us} + \mathbf{c} \neq \mathsf{Majority}_\epsilon(\mathbf{uS} + \mathbf{c})] \leq \epsilon$. This completes the proof. □

References

[Ale03] Michael Alekhnovich. More on average case vs approximation complexity. In *44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings*, pages 298–307. IEEE Computer Society, 2003.

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

[BFJ+19] Saikrishna Badrinarayan, Rex Fernando, Aayush Jain, Dakshita Khurana, and Amit Sahai. Statistical zap arguments. Cryptology ePrint Archive, Report 2019/780, 2019.

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *20th Annual ACM Symposium on Theory of Computing*, pages 103–112, Chicago, IL, USA, May 2–4, 1988. ACM Press.

[BGI16] Elette Boyle, Niv Gilboa, and Yuval Ishai. Breaking the circuit size barrier for secure computation under DDH. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 509–539, Santa Barbara, CA, USA, August 14–18, 2016. Springer, Heidelberg, Germany.

[BKM06] Adam Bender, Jonathan Katz, and Ruggero Morselli. Ring signatures: Stronger definitions, and constructions without random oracles. In Shai Halevi and Tal Rabin, editors, *TCC 2006: 3rd Theory of Cryptography Conference*, volume 3876 of *Lecture Notes in Computer Science*, pages 60–79, New York, NY, USA, March 4–7, 2006. Springer, Heidelberg, Germany.

[Blu87] Manuel Blum. How to prove a theorem so no one else can claim it. In *In: Proceedings of the International Congress of Mathematicians*, pages 1444–1451, 1987.

[BMW03] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 614–629, Warsaw, Poland, May 4–8, 2003. Springer, Heidelberg, Germany.

[CCH+19] Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, Ron D. Rothblum, and Daniel Wichs. Fiat-shamir: From practice to theory. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, page 1082–1090, New York, NY, USA, 2019. Association for Computing Machinery.

[CCRR18] Ran Canetti, Yilei Chen, Leonid Reyzin, and Ron D. Rothblum. Fiat-Shamir and correlation intractability from strong KDM-secure encryption. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part I*, volume 10820 of *Lecture Notes in Computer Science*, pages 91–122, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Heidelberg, Germany.

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

[DDN91] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography (extended abstract). In *23rd Annual ACM Symposium on Theory of Computing*, pages 542–552, New Orleans, LA, USA, May 6–8, 1991. ACM Press.

[DGI+19] Nico Döttling, Sanjam Garg, Yuval Ishai, Giulio Malavolta, Tamer Mour, and Rafail Ostrovsky. Trapdoor hash functions and their applications. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 3–32, Cham, 2019. Springer International Publishing.

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

[DP12] Ivan Damgård and Sunoo Park. How practical is public-key encryption based on LPN and ring-LPN? Cryptology ePrint Archive, Report 2012/699, 2012. http://eprint.iacr.org/2012/699.

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

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology – CRYPTO'86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194, Santa Barbara, CA, USA, August 1987. Springer, Heidelberg, Germany.

[GGM84] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. In *Foundations of Computer Science, 1984. 25th Annual Symposium on*, pages 464–479. IEEE, 1984.

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *17th Annual ACM Symposium on Theory of Computing*, pages 291–304, Providence, RI, USA, May 6–8, 1985. ACM Press.

[GOS12] Jens Groth, Rafail Ostrovsky, and Amit Sahai. New techniques for noninteractive zero-knowledge. *J. ACM*, 59(3), June 2012.

[HL18] Justin Holmgren and Alex Lombardi. Cryptographic hashing from strong one-way functions (or: One-way product functions and their applications). pages 850–858, 10 2018.

[JJ19] Abhishek Jain and Zhengzhong Jin. Statistical zap arguments from quasi-polynomial lwe. Cryptology ePrint Archive, Report 2019/839, 2019.

[KRR17] Yael Tauman Kalai, Guy N. Rothblum, and Ron D. Rothblum. From obfuscation to the security of Fiat-Shamir for proofs. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part II*, volume 10402 of *Lecture Notes in Computer Science*, pages 224–251, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Heidelberg, Germany.

[LVW19] Alex Lombardi, Vinod Vaikuntanathan, and Daniel Wichs. 2-message publicly verifiable wi from (subexponential) lwe. Cryptology ePrint Archive, Report 2019/808, 2019. https://eprint.iacr.org/2019/808.

[NY90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd Annual ACM Symposium on Theory of Computing*, pages 427–437, Baltimore, MD, USA, May 14–16, 1990. ACM Press.

[PS19] Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for np from (plain) learning with errors. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 89–114, Cham, 2019. Springer International Publishing.

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

A Trapdoor Hash for Linear Functions from DDH

In this appendix, we confirm Theorem 5.3 and show that, building on the corresponding construction from [DGI+19], we can obtain a rate-1 trapdoor hash scheme for linear functions with enhanced correctness from DDH. A proof for the constructions from the other standard assumptions (QR,DCR and LWE) follow similarly.

A.1 The Decisional Diffie-Hellman Assumption

We give the formal definition of the decisional version of the *Diffie-Hellman (DDH)* assumption [DH76].

Definition A.1 (Decisional Diffie-Hellman (DDH) assumption)**.** *A (prime-order)* group generator *is an algorithm $\mathcal{G}$ that takes as an input a security parameter 1^λ and outputs $(\mathbb{G}, p, g)$, where $\mathbb{G}$ is the description of a multiplicative cyclic group, p is the order of the group which is always a prime number, and g is a generator of the group. We say that $\mathcal{G}$* satisfies the DDH assumption *(or is* DDH-hard*) if for any PPT adversary, $\mathcal{A}$, there exists a negligible function $\mathsf{negl} : \mathbb{N} \to \mathbb{R}$ such that*

$$|\Pr[\mathcal{A}((\mathbb{G}, p, g), (g^{a_1}, g^{a_2}, g^{a_1 a_2})) = 1] - \Pr[\mathcal{A}((\mathbb{G}, p, g), (g^{a_1}, g^{a_2}, g^{a_3})) = 1]| < \mathsf{negl}(\lambda)$$

where $(\mathbb{G}, p, g) \xleftarrow{\$} \mathcal{G}(1^\lambda)$ *and* $a_1, a_2, a_3 \xleftarrow{\$} \mathbb{Z}_p$.

A.2 Parity Encoding over Discrete Log Groups

Let $\mathbb{G}$ be a multiplicative cyclic group of prime order p, and let $g \in \mathbb{G}$ be a generator. We recall the variant of the "distributed discrete log" function [BGI16], which was used in the DDH-based TDH construction from [DGI+19]. The subroutine $\mathsf{Parity}_{\mathbb{G},g}$ takes as input an element $h \in \mathbb{G}$, bounds on the failure probability $\tau > 0$ and on the input range $n \in \mathbb{N}$, and a pseudo-random function [GGM84] $\phi_K : \mathbb{G} \to \{0,1\}^{\lceil \log(2n/\tau) \rceil}$. At a high level, the function outputs the parity of the "distance" between h and the next zero of ϕ_K, and, intuitively, defines a randomized parity encoding for close group elements (those within range defined by n).

$\mathsf{Parity}_{\mathbb{G},g}(h, \tau, n, K)$:

1. Define $T := [2n \log_e(2/\tau)]/\tau$, and set $i := 0$.
2. While $i \leq T$:
   2.1. If $\phi_K(h \cdot g^i) = 0^{\lceil \log(2n/\tau) \rceil}$ then output $\mathsf{LSB}(i)$, else set $i := i + 1$.
3. Output $\mathsf{LSB}(i)$.

where LSB returns the least significant bit of a certain integer.

The function $\mathsf{Parity}_{\mathbb{G},g}(h, \tau, n, K)$ is computable in time polynomial in $|\mathbb{G}|$, n, $|K|$ and $1/\tau$. We hereby prove a useful property of the parity encoding. The original TDH construction from [DGI+19] relies on a slightly weaker statement, which was proven in [BGI16]. In order to obtain TDH for linear functions, rather than only for index functions, the following proposition has to be used. The proof is similar to the proof in the aforementioned work.

Proposition A.2. *Let* $\mathbb{G}$ *be a multiplicative cyclic group of prime order* p*, and let* $g \in \mathbb{G}$*,* $\tau > 0$*,* $n \in \mathbb{N}$ *with* $[2n \log_e(2/\tau)]/\tau < p$*. Let* $\phi_K : \mathbb{G} \to \{0,1\}^{\lceil \log(2n/\tau) \rceil}$ *be a pseudo-random function with a uniformly random key* $K \xleftarrow{\$} \{0,1\}^\lambda$*. Then, for any* $h \in \mathbb{G}$*, it holds that*

$$\Pr[\forall 0 \leq x \leq n, \quad \mathsf{Parity}_{\mathbb{G},g}(h, \tau, n, K) + \mathsf{Parity}_{\mathbb{G},g}(hg^x, \tau, n, K) = \mathsf{LSB}(x) \mod 2] \geq 1 - \tau - \mathsf{negl}(\lambda)$$

where the probability is taken over the choice of K*.*

Proof. Relying on the security of the PRF, we may replace the calls to ϕ_K in Parity by calls to a truly random function ϕ, at the cost of a negligible error probability.

The proof is completed, similarly to the proof of Proposition 3.2 from [BGI16], by the following simple case analysis.

(i) $\phi(h \cdot g^i) = \mathbf{0}$ for some $0 \leq i \leq n - 1$.

This implies that for $i < x \leq n$ we may get error.

The probability of such event is bound by $1 - (1 - (\tau/2n))^n \leq \tau/2$ (as pointed in [BGI16]).

(ii) $\phi(h \cdot g^i) = \mathbf{0}$ for no $0 \leq i \leq n - 1$ but for some $n \leq i \leq T$.

In which case, we never get error.

(iii) $\phi(h \cdot g^i) = \mathbf{0}$ for no $0 \le i \le T$.

This implies that Parity halts at $i = T + 1$, and we get error for some $0 \le x \le n$.

This happens with probability $(1 - (\tau/2n))^{2n \log_e(2/\tau)/\tau} < \tau/2$.

Bounding the probability of cases (i) and (iii) is sufficient since the three cases are a partition of the probability space over the choice of ϕ.

□

A.3 The Construction

We now present a modified variant of the DDH-based TDH construction from [DGI+19] (differences emphasized in red). The construction presented below supports encoding linear functions and not only "index functions" as it is the case in the original construction. On the other hand, this construction has two-sided error (compared to one-sided error in the original). Since we do not require input privacy for our construction of CI hash, we present a TDH where the hash and decoding are deterministic. This can be easily transformed to an input-secure randomized scheme, following ideas in [DGI+19].

The proof for function-privacy of the construction is identical to the proof of the scheme from [DGI+19]. A proof that the scheme has enhanced correctness, as required for correlation intractability, is provided after the construction.

We define linear functions $\mathbb{Z}_2^n \to \mathbb{Z}_2$ through vectors in $\mathbb{Z}_2^n$: Every vector $\mathbf{v} \in \mathbb{Z}_2^n$ is associated with the linear function $f_{\mathbf{v}}(x) = \mathbf{v}^T x$.

Construction A.1 (Rate-1 TDH for Linear Functions from DDH)**.** *Let $\tau := \tau(\lambda) > 1/\mathsf{poly}(\lambda)$ for some polynomial* poly*. The (simplified) rate-1 DDH-based* TDH *scheme for linear functions consists of the following algorithms.*

- $\mathsf{S}(1^\lambda, 1^n)$:

  1. *Sample* $(\mathbb{G}, p, g) \xleftarrow{\$} \mathcal{G}$
  2. *Sample a matrix*
  $$\mathbf{A} := \begin{pmatrix} g_{1,0}, g_{2,0}, \ldots, g_{n,0} \\ g_{1,1}, g_{2,1}, \ldots, g_{n,1} \end{pmatrix} \xleftarrow{\$} \mathbb{G}^{2 \times n}$$
  3. *Output*
  $$\mathsf{hk} := ((\mathbb{G}, p, g), \mathbf{A}) \tag{5}$$

- $\mathsf{G}(\mathsf{hk}, f_{\mathbf{v}})$: *parse* hk *as in Equation 5 and proceed as follows.*

  1. *Sample* $K \xleftarrow{\$} \{0,1\}^\lambda$ *and* $s \xleftarrow{\$} \mathbb{Z}_p$.
  2. *Set*
  $$u := g^s$$
  and
  $$\mathbf{B} := \begin{pmatrix} u_{1,0}, u_{2,0}, \ldots, u_{n,0} \\ u_{1,1}, u_{2,1}, \ldots, u_{n,1} \end{pmatrix} \qquad \text{where} \qquad u_{j,b} := \begin{cases} g_{j,b}^s \cdot g & \text{if } b = 1 \ \wedge \ \mathbf{v}_j = 1 \\ g_{j,b}^s & \text{otherwise} \end{cases}$$

3. Output

$$\mathsf{ek} := (u, \mathbf{B}, K) \qquad\qquad \mathsf{td} := (s, K) \tag{6}$$

- $\mathsf{H}(\mathsf{hk}, \mathsf{x})$: *parse* hk *as in Equation 5 and* $\mathbf{A} = (g_{j,b})_{j\in[n],b\in\{0,1\}}$*, and output*

$$\mathsf{h} := \prod_{j=1}^{n} g_{j,\mathsf{x}[j]} \tag{7}$$

- $\mathsf{E}(\mathsf{ek}, \mathsf{x})$*: parse* ek *as* $(u, \mathbf{B})$ *and* $\mathbf{B} = (u_{j,b})_{j\in[n],b\in\{0,1\}}$*, and output*

$$\mathsf{e} := \mathsf{Parity}_{\mathbb{G},g}\left(\prod_{j=1}^{n} u_{j,\mathsf{x}[j]}, \tau, n, K\right)$$

- $\mathsf{D}(\mathsf{td}, \mathsf{h})$: *parse* $\mathsf{h} \in \mathbb{G}$ *and* td *as in Equation 6, and output*

$$\mathsf{e}' := \mathsf{Parity}_{\mathbb{G},g}(\mathsf{h}^s, \tau, n, K) \tag{8}$$

A.4 Proof of Enhanced Correctness

Theorem A.3. *The trapdoor hash scheme from Construction A.1 has enhanced* $(1-\tau)$*-correctness.*

Proof. The enhanced correctness of the scheme is derived directly from Proposition A.2 as follows.

$$\begin{aligned}
&\Pr[\exists \mathsf{x} : \mathsf{H}(\mathsf{hk}, \mathsf{x}) = \mathsf{h},\ \mathsf{e} + \mathsf{e}' \neq f_{\mathbf{v}}(\mathsf{x})] \\
&= \Pr[\exists \mathsf{x} : \mathsf{H}(\mathsf{hk}, \mathsf{x}) = \mathsf{h},\ \mathsf{Parity}(\mathsf{h}^s g^{\sum \mathbf{v}_i \mathsf{x}_i}, \tau, n, K) + \mathsf{Parity}(\mathsf{h}^s, \tau, n, K) \neq f_{\mathbf{v}}(\mathsf{x}) \mod 2] \\
&\leq \Pr[\exists x \in [n],\ \mathsf{Parity}(\mathsf{h}^s g^x, \tau, n, K) + \mathsf{Parity}(\mathsf{h}^s, \tau, n, K) \neq \mathsf{LSB}(x) \mod 2] \\
&< \tau + \mathsf{negl}(\lambda)
\end{aligned}$$

□