# Non-Interactive Zero-Knowledge from LPN and MQ

Quang Dao<sup>\*</sup>

Aayush Jain<sup>+</sup>

Zhengzhong Jin<sup>‡</sup>

August 8, 2024

#### Abstract

We give the first construction of non-interactive zero-knowledge (NIZK) arguments from post-quantum assumptions other than Learning with Errors. In particular, we achieve NIZK under the polynomial hardness of the Learning Parity with Noise (LPN) assumption, and the exponential hardness of solving random under-determined multivariate quadratic equations (MQ). We also construct NIZK satisfying statistical zero-knowledge assuming a new variant of LPN, Dense-Sparse LPN, introduced by Dao and Jain (CRYPTO 2024), together with exponentially-hard MQ.

The main technical ingredient of our construction is an extremely natural (but only in hindsight!) construction of correlation-intractable (CI) hash functions from MQ, for a NIZK-friendly sub-class of constant-degree polynomials that we call concatenated constant-degree polynomials. Under exponential security, this hash function also satisfies the stronger notion of approximate CI for concatenated constant-degree polynomials. The NIZK construction then follows from a prior blueprint of Brakerski-Koppula-Mour (CRYPTO 2020). In addition, we show how to construct (approximate) CI hashing for degree-*d* functions from the (exponential) hardness of solving random degree-*d* equations, a natural generalization of MQ. To realize NIZK with statistical zero-knowledge, we design a lossy public-key encryption scheme with approximate linear decryption and inverse-polynomial decryption error from Dense-Sparse LPN. These constructions may be of independent interest.

Our work therefore gives a new way to leverage MQ with uniformly random equations, which has found little cryptographic applications to date. Indeed, most applications in the context of encryption and signature schemes make use of structured variants of MQ, where the polynomials are not truly random but posses a hidden planted structure. We believe that the MQ assumption may plausibly find future use in the designing other advanced proof systems.

<sup>\*</sup>Carnegie Mellon University. Email: qvd@andrew.cmu.edu

<sup>&</sup>lt;sup>†</sup>Carnegie Mellon University. Email: aayushja@andrew.cmu.edu

<sup>&</sup>lt;sup>‡</sup>Northeastern University. Email: zh.jin@northeastern.edu

# Contents

1	Introduction	1			
	1.1 Our Results	2			
	1.2 Discussion and Related Works	5			
	1.3 Open Questions	6			
2	Technical Overview	6			
3	Preliminaries	9			
	3.1 Probability Lemmas	10			
	3.2 Relations	11			
	3.3 Correlation Intractability	12			
	3.4 Non-Interactive Zero-Knowledge	13			
	3.5 Public-Key Encryption	14			
	3.6 Commit-then-Open Protocols	14			
4	Assumptions	16			
	4.1 Multivariate Cryptography	16			
	4.2 Learning Parity with Noise	19			
5	NIZK and CI Hash Constructions	20			
	5.1 Correlation Intractability from Approximate MPS	21			
	5.2 NIZK Constructions	23			
6	6 Lossy PKE from Dense-Sparse LPN				
7	References	29			

### 1 Introduction

Zero-knowledge (ZK) proofs [GMR85] have played a central role in the theory and practice of cryptography for almost four decades. At a high level, a zero-knowledge proof enables an efficient prover to convince a verifier about the validity of a statement without disclosing any other information. While ZK proofs have found countless applications in cryptography, they are most useful in their non-interactive variant, where both the prover and verifier gets access to a *common reference string* generated by a trusted party, and the proof consists of a *single* message from the prover to the verifier. This notion is known as *non-interactive zero knowledge* (NIZK) [DMP88]. NIZKs are widely used in various applications, including CCA-secure encryption [NY90, DDN91], signature schemes [BMW03, BKM06], blockchains [BCG<sup>+</sup>14], and more.

Constructions of NIZK have had a curious history. While NIZKs can be easily built in the random oracle model using the Fiat-Shamir Heuristic [FS87], only a handful number of assumptions are known to give rise to NIZKs in the standard model with reduction-based security proofs. Since the seminal work building the first NIZK [DMP88], there has been tremendous efforts in designing new NIZKs from a variety of assumptions [BFM88, FLS90, BY93, CHK03, GOS06b, GOS06a, GR13, SW14, CL18, CCRR18, CCH<sup>+</sup>19, PS19, CKU20, BKM20, JJ21, GLS22, CJJQ23, CW23]. However, up until six years ago, we only knew NIZKs from Factoring and Bilinear-group-based assumptions [BFM88, FLS90, BY93, CHK03, GOS06b, GOS06a, GR13, SW14, CL18, CCRR18].

This changed drastically with recent progress on instantiating the Fiat-Shamir Heuristic via *correlation-intractable* hash functions [CCH<sup>+</sup>19]. Informally, a hash function  $\mathcal{H}$  is correlation intractable with respect to a relation R if it is computationally difficult to find any input x such that  $(x, \mathcal{H}(x)) \in R$ . These recent breakthroughs culminated in the first constructions of post-quantum NIZK from Learning with Errors (LWE) [CCH<sup>+</sup>19, PS19], a central assumption [Reg05] in post-quantum cryptography that has enabled a vast array of other advanced primitives [BV11, GSW13, GVW13, GVW15, GKW17, WZ17, GKW18, BCM<sup>+</sup>18, Mah18b, Mah18a, Bra18].

Correlation-intractability has proven to be a game-changer not only in the design of NIZKs but also in the construction of other advanced proof systems such as batch arguments (BARGs) for NP [CJJ21, WW22, DGKV22, PP22, GSWW22, KLVW23], succinct non-interactive arguments (SNARGs) for P [CJJ22, HJKS22, CGJ<sup>+</sup>23] and other subclasses of NP [JKKZ21, KLV23, BBK<sup>+</sup>23, NWW23], or incrementally verifiable computation [WW22, DGKV22, PP22]. In the context of NIZKs, the community has designed new techniques that enabled constructions of NIZK from new sets of well-studied assumptions such as DDH and LPN [BKM20], and eventually just (subexponentially hard) DDH [JJ21].

Unfortunately, constructing NIZKs from purely post-quantum assumptions (other than LWE) has remained elusive. There has been almost no progress on this goal since the original constructions from LWE in [CCH<sup>+</sup>19, PS19]. This brings us to the central question for this paper:

**Question.** *Can we build NIZKs from well-studied post-quantum assumptions other than Learning with Errors (and its variants)?* 

To make progress on this question along the correlation intractability (CI) framework, it is crucial to explore constructions of CI hash functions from other post-quantum assumptions, for a function class that proves sufficient for NIZK. This leads us to the second question:

**Question.** *Can we build CI hashing for function classes expressive enough to imply NIZK from post-quantum assumptions other than LWE?* 

**Why other post-quantum assumptions?** We believe that it is important to base NIZKs on other post-quantum assumptions for the following reasons.

First, the state of affairs in terms of variety of assumptions implying advanced post-quantum cryptography is highly unsatisfactory. While LWE is known to imply many advanced primitives [BV11, GSW13, GVW13, GVW15, GKW17, WZ17, GKW18, BCM<sup>+</sup>18, Mah18b, Mah18a, Bra18], it remains the only known post-quantum assumption, or even the only known assumption, for most of these advanced applications. This motivates the search for alternate constructions of advanced primitives from other post-quantum assumptions. In this light, NIZK is one such primitive that still remains a challenging test of expressivity for assumptions. NIZK constructions have always been surprisingly difficult to realize, and often once NIZK is built from an assumption, we have also found ways to use the same assumption for designing other advanced cryptography, e.g. [CJJ21, CJJ22, HJKS22, CGJ<sup>+</sup>23, GS08, GOS06b].

Second, while LWE is the only known post-quantum assumption implying NIZK (or even much stronger primitives like fully homomorphic encryption), it turns out not to be as useful for building other primitives such as indistinguishability obfuscation (iO) [BGI+01], for which we know of constructions from other assumptions (that are not all post-quantum secure) [JLS21, JLS22]. Thus, we believe that new techniques for leveraging other well-studied post-quantum assumptions to build NIZKs could be helpful for other post-quantum, or quantum, cryptography applications for which LWE might not be the best assumption (besides iO, another such example is public-key quantum money [LMZ23]).

### 1.1 Our Results

We answer the above questions by constructing NIZK from two well-studied post-quantum assumptions: the polynomial hardness of the Learning Parity with Noise (LPN) [BFKL94] assumption, in a regime slightly stronger than that implying public-key encryption, and the *exponential* hardness<sup>1</sup> of solving random underdetermined multivariate quadratic (MQ) [OSS84] equations. These are arguably *the* central assumptions in code-based and multivariate-based cryptography, respectively, and have been subject to intense cryptanalysis over several decades.

Learning Parity with Noise [BFKL94] posits the hardness of decoding codewords from a random linear code that has been corrupted with a random sparse noise. More formally, the LPN<sub> $n,m,\eta$ </sub> assumption states that

$$(\mathbf{A}, \mathbf{s} \cdot \mathbf{A} + \mathbf{e})$$
 is indistinguishable from  $(\mathbf{A}, \mathbf{u})$ , (1)

for a random matrix  $\mathbf{A} \in \mathbb{F}_2^{n \times m}$ , random  $\mathbf{s} \in \mathbb{F}_2^{1 \times n}$ , random  $\mathbf{u} \in \mathbb{F}_2^{1 \times m}$ , and the sparse noise vector  $\mathbf{e} \in \mathbb{F}_2^{1 \times m}$  is Bernoulli distributed with some probability  $\eta$ . We use LPN with noise rate  $\eta = n^{-1/2-\delta}$  for any  $\delta > 0$ , which is slightly lower than the noise rate  $n^{-1/2}$  sufficient for public-key encryption [Ale03]. In this regime, the best known attacks (see [BCGI18, BCG<sup>+</sup>20] for a survey of attacks) run in time  $2^{\tilde{O}(n^{1/2-\delta})}$ .

The Multivariate Quadratic problem has an even longer history of study, whose first usage in cryptography can be found in e.g. [OSS84, MI88]. Specifically, the MQ<sub>*n*,*m*</sub> assumption (over  $\mathbb{F}_2$ ) states that it is computationally difficult to find a solution to a system of *m* random quadratic

<sup>&</sup>lt;sup>1</sup>In this work, we say that an assumption satisfies exponential hardness if there exists some (however small) constant  $\tau > 0$  so that any polynomial-time adversary has at most  $2^{-\tau m}$  chance of breaking the assumption on input length m.

equations in *n* variables:

$$\begin{cases} q_1(\mathbf{X}) := \sum_{i,j=1}^n a_{i,j}^{(1)} X_i X_j + \sum_{i=1}^n b_i^{(1)} X_i + c^{(1)} = 0, \\ \vdots \\ q_m(\mathbf{X}) := \sum_{i,j=1}^n a_{i,j}^{(m)} X_i X_j + \sum_{i=1}^n b_i^{(m)} X_i + c^{(m)} = 0. \end{cases}$$
(2)

The MQ assumption is considered in two regimes: underdetermined (when m < n), and overdetermined (when m > n).<sup>2</sup> Arguably the *central target for algebraic cryptanalysis*, a tremendous body of works [Pat97, KS98, KS99, KPG99, CKPS00, CGMT02, TW12, MHT13, CHMT14] have studied MQ in both parameter regimes. The best known attacks against MQ runs in exponential time when  $m \approx n$ , with a gradual decrease in runtime to polynomial time when  $m = \Omega(n^2)$  [KS99] or  $m = O(\sqrt{n})$  [KPG99, CGMT02, TW12, MHT13, CHMT14]. In this work, we rely on the *exponential* hardness of underdetermined MQ<sub>n,m</sub>, where  $m = n^{1-\epsilon}$  for an arbitrarily small constant  $\epsilon > 0$ . Hardness in this regime is well-supported by the best known attacks (see Section 4.1 for details).

**Theorem 1.1** (Informal). *There exists NIZK for* NP *in the common random string model, assuming the following:* 

- no polynomial-time algorithm succeeds in solving LPN<sub>*n*,*m*, $\eta$ </sub> with noticeable probability, where m = poly(n) and  $\eta = O(n^{-1/2-\delta})$ , for some  $\delta > 0$ ,
- no polynomial-time algorithm succeeds in solving  $MQ_{n,m}$  with  $\Omega(2^{-\tau m})$  probability, where  $m = n^{1-\rho}$ , for some  $\rho, \tau > 0$ .

**On Exponential Security.** Our notion of exponential security for MQ requires that a polynomialtime adversary has exponentially-small success probability. We could also consider a different notion of exponential security where an exponential-time adversary has negligible success probability. While contrived examples exist where the latter do not imply the former,<sup>3</sup> for most natural problems (including MQ) it is expected that the former notion is weaker. Our exponential hardness assumption is also qualitatively weaker than the "almost optimal security" notion of earlier CI hash constructions [CCRR18, HL18, CCH<sup>+</sup>18], which requires poly-time adversary to have roughly poly( $\lambda$ )/2<sup> $\lambda$ </sup> success probability.

We can also relax the exponential hardness requirement by relying on the *polynomial* hardness of a new *Approximate MQ* assumption, which only requires a polynomial-time adversary to solve *any constant* (say 99%) of the random quadratic equations. We believe that this assumption is very natural, and its hardness is implied by the exponential hardness of MQ (see Section 4.1 for details). **Statistical Zero-Knowledge.** Our NIZK from Theorem 1.1 satisfies adaptive computational soundness and adaptive computational zero-knowledge. We may upgrade our construction to satisfy *statistical* zero-knowledge (which requires weakening soundness to only be non-adaptive [Pas13]) by leveraging a recently introduced variant of LPN, called *Dense-Sparse LPN* [DJ24]. At a high level, the DS-LPN<sub>*n.m.k.n*</sub> assumption states that

$$(\mathbf{A}, \mathbf{s} \cdot \mathbf{A} + \mathbf{e})$$
 is indistinguishable from  $(\mathbf{A}, \mathbf{u})$ , (3)

<sup>&</sup>lt;sup>2</sup>In the overdetermined case, since a solution is often not available for random MQ equations, Equation (2) is modified to be in the *planted* regime, where the right-hand side is not all-zero but is the evaluation  $(q_1(\mathbf{x}), \ldots, q_m(\mathbf{x}))$  for a random  $\mathbf{x} \in \mathbb{F}_q^n$ . We will only focus on the underdetermined regime, where a solution will exist with overwhelming probability (if  $m \ll n$ ).

<sup>&</sup>lt;sup>3</sup>An example is as follows: define a problem that is unsolvable with  $1 - 2^{-\lambda}$  probability, and trivially solvable with  $1/2^{-\lambda}$  probability.

Assumptions	CRS	SND	ZK	Post-Quantum
Factoring [BFM88, FLS90, BY96]	random	S	C	no
Bilinear Mans [CHK07, COS06b]	random	С	S	no
Dimear waps [er nov, Gesoob]	structured	S	С	
Bilinear Mans [COS06a]	structured	C	S	no
Diffical Waps [GOStoa]	random	S	C	
earning with Errors [CCH <sup>+</sup> 10 PS10]	random	С	S	yes
	structured	S	С	
DDH + LPN [BKM20]	random	С	С	no
sub-exponential DDH [JJ21]	random	C	S	no
LPN + exponential MQ (Ours)	random	C	С	yes
DS-LPN + exponential MQ (Ours)	structured	C	S	yes

Figure 1: Known constructions of NIZK from concrete/well-studied assumptions. In the above, assumptions are polynomially-secure unless stated otherwise. We abbreviate the following: CRS = (type of) common reference string; SND = soundness guarantee; ZK = zero-knowledge guarantee; S = statistical; C = computational.

for a random  $\mathbf{s} \in \mathbb{F}_2^{1 \times n}$ , random  $\mathbf{u} \in \mathbb{F}_2^{1 \times m}$ , a Bernoulli-distributed noise  $\mathbf{e} \in \mathbb{F}_2^{1 \times m}$  with error probability  $\eta$ , and a *structured* matrix  $\mathbf{A} = \mathbf{TM} \in \mathbb{F}_2^{n \times m}$  that is the product of a random (dense) matrix  $\mathbf{T} \in \mathbb{F}_2^{n \times 2n}$  and a random *k*-sparse matrix  $\mathbf{M} \in \mathbb{F}_2^{2n \times m}$  (with only *k* non-zero entries per column). In [DJ24], the authors showed that for any constant  $k \geq 3$ , a small enough constant  $\delta > 0$ and some function  $1 < \rho(\delta) < k/2$ , Dense-Sparse LPN with  $\eta = O(1/n^{1-\delta})$  and  $m = n^{\rho(\delta)}$  implies lossy trapdoor functions [PW08], and is plausibly secure against subexponential time adversaries.

Using Dense-Sparse LPN, we construct a *lossy public-key encryption* [PVW08, BHY09, HLOV11] scheme with  $1/\operatorname{poly}(\lambda)$ -approximate linear decryption,<sup>4</sup> which suffices for building NIZK with statistical zero-knowledge. Our construction crucially relies on the inverse-polynomial noise rate; we do not know of any lossy PKE construction (with  $1/\operatorname{poly}(\lambda)$ -approximate linear decryption) from LPN with *any* error probability  $\epsilon$ , even in the quasi-polynomial time broken regime of  $\epsilon = O(\log^2 n/n)$  [BLVW19]. See Section 4.2 for details on this assumption, and Section 6 for our construction.

**Theorem 1.2** (Informal). There exists NIZK for NP in the common reference string model with statistical zero-knowledge, assuming exponential hardness of  $MQ_{n,m^{1-\rho}}$  (as in Theorem 1.1) together with polynomial hardness of DS-LPN<sub>n,m,k,\epsilon</sub> for some constant  $k \ge 3$ ,  $\epsilon = O(n^{1-\delta})$ , and  $m = n^{\rho(\delta)}$ , where  $\delta > 0$  is a small enough constant and  $1 < \rho(\delta) < k/2$ .

**Correlation-Intractable Hash Functions.** The main enabler behind our NIZK construction, as in a number of previous constructions, is a correlation-intractable hash function family for a class

<sup>&</sup>lt;sup>4</sup>This does not follow directly from generic transformations from lossy trapdoor functions, since the decryption function may no longer be approximately linear.

of relations expressive enough for NIZK. In our case, these are function relations (x, f(x)), for all functions f that are *well-approximated* by a suitable sub-class of all *constant-degree polynomials*.

We generalize the  $MQ_{n,m}$  assumption to posit the hardness of solving random polynomial systems for some *constant degree* d > 2; we call this assumption  $MPS_{n,m,d}$ .<sup>5</sup> We show that the  $MPS_{n,m,d}$  assumption naturally implies a CI hash function for the class of degree-d polynomials. Under exponential hardness of  $MPS_{n,m,d}$ , the same hash function is CI for a larger class of functions that only need to be well-approximated by degree-d polynomials. Finally, we show how to base CI hashing with (almost) the same requirement on just  $MQ_{n,m}$ , weakening the approximation function class to *concatenated* constant-degree polynomials. These are polynomial tuples of the form

$$\begin{pmatrix} P_{1}(\mathbf{x}) \\ \vdots \\ P_{m}(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} P_{1,1}(\mathbf{x}_{1}) \| \dots \| P_{1,\ell}(\mathbf{x}_{\ell}) \\ \vdots \\ P_{m,1}(\mathbf{x}_{1}) \| \dots \| P_{m,\ell}(\mathbf{x}_{\ell}) \end{pmatrix} \in \mathbb{F}_{2}^{m\ell} \quad \text{for some } \ell \in \mathbb{N},$$
(4)

where  $\mathbf{x} = \mathbf{x}_1 \| \dots \| \mathbf{x}_\ell \in \mathbb{F}_2^{n\ell}$  and " $\|$ " denotes string concatenation.

**Theorem 1.3** (Informal). For every small enough constant  $\epsilon > 0$  and constant degree  $d \ge 2$ , there exists correlation-intractable hash functions for the following function classes:

- Functions  $\epsilon$ -approximable by degree-d polynomials, from exponential hardness of  $MPS_{n,m,d}$ , where  $m = n^{1-\rho}$  for any  $0 < \rho < 1/2$ ,
- Functions ε-approximable by ℓ-concatenation of degree-d polynomials, from exponential hardness of MQ<sub>n,m</sub>, where m = n<sup>1-ρ</sup> for any 0 < ρ < 1/2 and ℓ is large enough.</li>

We expect that for MPS of degree d > 2, the compression can be made even lower than  $m = \omega(n^{1/2})$ ; we leave determining the right hardness threshold for general degree d to future work.

### 1.2 Discussion and Related Works

We note that our construction is a rare departure from typical applications of MQ-based assumptions to design primitives such as signatures [Pat97, KPG99, PCG01, DS05, BP17, Beu22b] and encryption schemes [MI88, Pat96, PGC98, HLY12, TDTD13]. To the best of our knowledge, all known constructions of signatures and encryption schemes from MQ-based systems (in the standard model) make use of *structured* variants of MQ, where the polynomials are not truly randomly chosen. In many cases, this planted structure is not sufficiently hiding, leading to countless attacks against various MQ-based schemes over the years [Pat95, KS98, FJ03, WBP05, DFSS07, DDY<sup>+</sup>08, BFP11, AFF<sup>+</sup>14, DDS<sup>+</sup>20, Beu21, Beu22a]. In our case, the polynomials are identically random, giving rise to the first advanced cryptographic primitive from random MQ systems (together with LPN).

We also note some prior applications of truly random MQ: as a PRG [BGP06, LLY08] (in the overdetermined regime), a low-complexity universal one-way hash function [AHI+17], and in the context of pseudorandom correlation generators [BCG+19]. To date, it is not known how to construct either collision-resistant hashing or public-key encryption from random MQ. A prior work [DY08] suggested that the same hash function as ours (for MPS with degree 3 and above) is collision-resistant, though their argument was heuristic and no provable reductions were presented.

<sup>&</sup>lt;sup>5</sup>Short for Multivariate Polynomial Solving.

Moreover, both LPN and MQ in our parameter regime are not known to imply many advanced cryptographic primitives. Indeed, LPN with error probability  $O(n^{-0.5-\rho})$ , for an arbitrarily small  $\rho > 0$ , is currently only known to imply basic Cryptomania primitives such as CPA and CCA-secure public-key encryption [Ale03, DMN12, KMP14, YZ16], along with UC-secure oblivious transfer [DGH<sup>+</sup>20]. This is a stark contrast from prior NIZK constructions from standard assumptions, where the underlying assumptions can be shown to imply highly structured primitives such as trapdoor permutations (in the case of Factoring) or additively homomorphic encryption (for all other assumptions implying NIZK). In this sense, we achieve NIZK under arguably the *weakest* standard assumptions to date.

Our work thus once again proves the power of devising new ways to use combination of assumptions. Similar effect was observed in the works [JLS21, JLS22] which constructed indistinguishability obfuscation leveraging the subexponential security of Bilinear Maps, together with Minicrypt-like assumptions of PRGs in NC<sup>0</sup> and LPN with noise probability  $n^{-\delta}$  for arbitrarily small constant  $\delta > 0$ . There are also similar recent examples in other contexts (such as [AY20, LLL22]).

### 1.3 **Open Questions**

We believe that our work opens up many exciting avenues for future work. First, our work motivates the study of random MQ for the design of other post-quantum advanced cryptographic primitives (potentially in conjunction with other assumptions). Second, it is plausible that one might be able to extend the techniques in this work to realize correlation intractability for the class of TC<sup>0</sup> circuits, which might lead to realizing more advanced proof systems such as BARGs and SNARGs. Finally, we motivate studying the complexity of solving underdetermined higherdegree equations, which has not received as much attention compared to the quadratic case.

## 2 Technical Overview

To obtain our NIZK constructions, we follow the paradigm for building NIZKs via Fiat-Shamir and correlation-intractable hashing (CIH) as described in [CCH<sup>+</sup>19]. Our main technical contribution is a new construction of CIH from the MQ (or more generally MPS) assumption, for a class of functions expressive enough to give rise to NIZK. As such, we will start our technical overview by talking about our CIH constructions, with the motivation for NIZK in mind. Next, we will recall the NIZK template via Fiat-Shamir and CIH in [CCH<sup>+</sup>19, BKM20], and show why our CIH is sufficient for instantiating this template. Finally, we briefly mention our lossy PKE construction which suffices for statistical zero-knowledge.

**CIH for Constant Degree Polynomials.** A hash function Hash is *correlation-intractable* for a function class  $\mathcal{F}$  if for any function  $f \in \mathcal{F}$ , it is hard to efficiently find an input x such that Hash(hk, x) =  $f(\mathbf{x})$ , where the key hk is honestly generated and given to the adversary. We start by describing how we can leverage the hardness of solving underdetermined random degree-*d* multivariate equations over  $\mathbb{F}_2$  to realize CIH for constant degree-*d* polynomials.

Our CIH construction is extremely simple (but only in hindsight!). To build a CIH with input length n and output length m < n, we first sample m polynomials of degree d,<sup>6</sup> denoted

<sup>&</sup>lt;sup>6</sup>By this, we mean polynomials with terms of every degree from *d* to 0. We denote by Poly(n, d) the set of such polynomials (over  $\mathbb{F}_2$ ).

 $P_1, \ldots, P_m \in \mathsf{Poly}(n, d)$ , with uniformly random coefficients, and set the CIH hash key hk =  $(P_i)_{i \in [m]}$ . The hash on an input  $\mathbf{x} \in \mathbb{F}_2^n$  is simply polynomial evaluation:

$$\mathsf{Hash}(\mathsf{hk}, \mathbf{x}) = (P_1(\mathbf{x}), \dots, P_m(\mathbf{x})) \in \mathbb{F}_2^m.$$
(5)

We now sketch the proof of correlation intractability for Hash, assuming the hardness of  $MPS_{n,m,d}$ . For any fixed degree-*d* polynomial function  $(g_1, \ldots, g_m)$ , we can view the sampling of  $(P_1, \ldots, P_m)$  as first sampling random degree-*d* polynomials  $(h_1, \ldots, h_m)$  with uniformly random coefficients, and then setting  $P_i = g_i + h_i$  for every  $i \in [m]$ . If an adversary breaks the correlation intractability for  $(g_1, \ldots, g_m)$ , then it must be the case that  $P_i(\mathbf{x}) = g_i(\mathbf{x})$  for every  $i \in [m]$ , or equivalently,  $g_i(\mathbf{x}) + h_i(\mathbf{x}) = g_i(\mathbf{x})$  for every *i*. However, this implies  $h_i(\mathbf{x}) = 0$  for every *i*, which contradicts the hardness of solving a random system of degree-*d* equations.

While the above CIH construction and proof are elegant and immediate, they suffer from the following drawbacks:

- 1. First, the hardness of solving random underdetermined degree-*d* equations (for any  $d \ge 3$ ) is not as well-established as that of solving quadratic equations, namely  $MQ_{n,m}$ . We would like to build CIH for degree-*d* polynomials, where *d* may be an arbitrarily large constant, from the hardness of  $MQ_{n,m}$ .
- 2. Second, CIH for constant-degree polynomials is *not* sufficient for obtaining NIZK. Looking ahead, we will need to build CIH for a larger class of functions that are *well-approximated* by constant-degree polynomials.

We say that a function  $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$  is  $\epsilon$ -approximable by degree-d polynomials if there exists some distributions  $\mathfrak{G}_1, \ldots, \mathfrak{G}_m \subseteq \mathsf{Poly}(n, d)$  such that for  $g_1, \ldots, g_m$  independently sampled from these distributions and for any input  $\mathbf{x}$ , it holds that  $f(\mathbf{x})_i = g_i(\mathbf{x})$  for at least  $(1 - \epsilon)$ -fraction of the output length with overwhelming probability. In other words, we have

$$\Pr\left[\Delta\left(f(\mathbf{x}), (g_1(\mathbf{x}), \dots, g_m(\mathbf{x}))\right) \le \epsilon m\right] \le \mathsf{negl}(m),\tag{6}$$

where  $\Delta(\cdot, \cdot)$  denotes the Hamming distance. We want to achieve CIH for this class of functions that are  $\epsilon$ -approximable by degree-*d* polynomials, for a constant  $\epsilon > 0$  that can be arbitrarily small. We call this an *approximate* CI hash.

It turns out that the hash construction in Equation (5) *already* satisfies this notion, with a loss in security that depends on  $\epsilon$ . The idea is simple: since we know that the Hamming difference between  $f(\mathbf{x})$  and  $(g_1(\mathbf{x}), \ldots, g_m(\mathbf{x}))$  is small, we may simply *guess* this difference, written as a vector  $\mathbf{e} \in \mathbb{F}_2^m$ . If our guess  $\mathbf{e}$  is correct, then we may continue the proof strategy above with the following modification. We now view sampling  $(P_1, \ldots, P_m)$  as first sampling degree-*d* polynomials  $\mathbf{h} = (h_1, \ldots, h_m)$ , then setting  $P_i = g_i + h_i + e_i$  for all  $i \in [m]$  (with  $e_i$  added to the constant term). Violating the CI property for *f* now translates to having

h(x) = f(x) + g(x) + e, which is equal to 0 if our guess e is correct.

As a consequence, our reduction loses a factor  $\Gamma$  which is equal to the size of all possible error vectors  $\mathbf{e} \in \mathbb{F}_2^m$ . Since  $\mathbf{e}$  is at most  $(\epsilon m)$ -sparse, this factor is  $\Gamma \sim 2^{H(\epsilon)m} = 2^{\Omega(m)}$ , where  $H(\cdot)$  is the binary entropy function. This is why we need to assume exponentially-small success probability of breaking MPS<sub>*n,m,d*</sub>.

**Basing CIH from MQ.** Going back to the first problem, namely that of basing degree-*d* (approximate) CIH solely on MQ, we may try the following degree-reduction approach via tensoring. For instance, we may represent any degree-4 polynomial in  $\mathbf{x} \in \mathbb{F}_2^n$  as a degree-2 polynomial in  $\mathbf{x} \otimes \mathbf{x} \in \mathbb{F}_2^{n^2}$ . This means that we may instead hand out random degree-2 polynomials  $hk = (Q_1, \ldots, Q_m) \in (Poly(n^2, d))^m$  as the hash key, and evaluate the hash by first tensoring  $\mathbf{x}$  then applying  $\mathbf{Q}$ :

$$\mathsf{Hash}(\mathsf{hk},\mathbf{x}) = (Q_1(\mathbf{x}\otimes\mathbf{x}),\ldots,Q_m(\mathbf{x}\otimes\mathbf{x})) \in \mathbb{F}_2^m.$$

Using a similar argument as above, to prove CIH for a fixed degree-4 function  $\mathbf{g} = (g_1, \dots, g_m)$ , we first sample random quadratic polynomials  $\mathbf{h} = (h_1, \dots, h_m)$ , then set  $\mathbf{Q} = \mathbf{h} + \mathbf{g}$ . Breaking correlation intractability then implies finding  $\mathbf{x}$  such that  $\mathbf{h}(\mathbf{x} \otimes \mathbf{x}) = \mathbf{0}$ , which violates the  $MQ_{n^2,m}$  assumption.

The problem with this approach is that in order for  $MQ_{n^2,m}$  to be difficult, we must have  $m > \sqrt{n^2} = n$ , meaning that the hash function is no longer compressing! To get around this issue, we need to restrict ourselves to achieving (approximate) CIH only for a suitable *sub-class* of constant-degree polynomials. In this work, we consider the following sub-class of *concatenated* polynomials. Our inputs  $\mathbf{x}$ , which we think of as growing in length, are now divided into consecutive chunks that are of a *fixed* length  $p(\lambda)$ ; for simplicity, assume that  $p(\lambda) = \lambda$ . Formally, we write  $\mathbf{x} = \mathbf{x}_1 \| \dots \| \mathbf{x}_\ell \in \mathbb{F}_2^{\lambda,\ell}$ , where each  $\mathbf{x}_i \in \mathbb{F}_2^{\lambda}$ , and  $\ell$  may be chosen to be an arbitrarily large polynomial in  $\lambda$ . Given  $\mathbf{x}$ , we consider only degree-*d* polynomials of the form  $P = P_1 \| \dots \| P_\ell : \mathbb{F}_2^{\lambda\ell} \to \mathbb{F}_2^{\ell}$  such that  $P(\mathbf{x}) = P_1(\mathbf{x}_1) \| \dots \| P_\ell(\mathbf{x}_\ell)$ , and each  $P_j : \mathbb{F}_2^{\lambda} \to \mathbb{F}_2$  is of degree *d*. We denote by  $\text{CPoly}(\ell, d)$  the set of such  $\ell$ -concatenated degree-*d* polynomials.

The tensoring approach now works with this new class of concatenated polynomials. To design an (approximate) CIH for the class  $\text{CPoly}(\ell, d)$  (assuming d is even for simplicity), we may sample random quadratic polynomials hk =  $(Q_1, \ldots, Q_\ell) \in (\text{Poly}(n, 2))^\ell$  as the hash key, where  $n = \lambda^{d/2} \ell$  is the input length. Hashing now works as follows: given an input  $\mathbf{x} = \mathbf{x}_1 \| \ldots \| \mathbf{x}_\ell \in \mathbb{F}_2^{\lambda \ell}$ ,

$$\mathsf{Hash}(\mathsf{hk},\mathbf{x}) = (Q_1(\mathbf{z}),\ldots,Q_\ell(\mathbf{z})) \in \mathbb{F}_2^m, \quad \text{where } \mathbf{z} := \mathbf{x}_1^{\otimes d/2} \| \ldots \| \mathbf{x}_\ell^{\otimes d/2} \|$$

The proof of approximate CI for this hash function is almost exactly the same as our above proof sketch for  $MPS_{n,m,d}$ . What is new here is that the parameters now allow for compression: we have  $n = \lambda^{d/2} \ell$  and  $m = \ell$ . Therefore, we may assume that  $n^{1-\delta} < m < n$  for any  $\delta > 0$  for which  $MQ_{n,m}$  is plausibly exponentially secure, and then set  $\ell$  large enough so that the inequality happens.

A final question remains: is approximate CIH for concatenated constant-degree polynomials sufficient for obtaining NIZK? To answer this question, we will describe the NIZK via Fiat-Shamir and CIH template in [CCH<sup>+</sup>19, BKM20], then note how our CIH notion suffices for NIZK in this template.

NIZK from Approximate CIH for CPoly. The starting point for prior NIZK constructions in the CI framework is a Sigma protocol  $\Sigma$  for a NP-complete language *L* that satisfies some special properties.<sup>7</sup> Denoting the messages of  $\Sigma$  by  $(\alpha, \beta, \gamma)$ , we want the following to hold for  $\Sigma$ :

- 1. The first message  $\alpha = \text{Enc}(\text{pk}, m)$  is an encryption of some underlying content *m*, with pk part of the common reference string;
- 2. The challenge  $\beta \in \{0, 1\}$  is a bit;

<sup>&</sup>lt;sup>7</sup>A concrete example for  $\Sigma$  is the Graph Hamiltonicity protocol [Blu86, FLS90].

3. Given sk as the trapdoor, for every false instance  $x \notin L$ , there exists an efficiently computable BadChal<sub>sk</sub> that on input the first message  $\alpha$ , outputs the *unique* challenge  $\beta^* \in \{0, 1\}$  for which there does not exist  $\gamma$  that would make the  $\Sigma$ -protocol's verifier accept.

The NIZK construction now applies the Fiat-Shamir heuristic [FS87] on this  $\Sigma$  protocol, deriving the challenge as  $\beta$  = Hash(hk,  $\alpha$ ). The soundness of this construction then relies on the hash function being correlation-intractable for an expressive enough function class that captures the BadChal<sub>sk</sub> functions. Thus, it is crucial that BadChal<sub>sk</sub> is *as simple as possible*. In particular, Brakerski-Koppula-Mour showed in [BKM20] that assuming the PKE scheme satisfies approximate linear decryption (which they instantiate from LPN), we can make the BadChal<sub>sk</sub> function to be  $\epsilon$ -approximable by constant degree-d polynomials. This is exactly the function class that we constructed CIH for (in our construction from MPS<sub>*n,m,d*</sub>)!

We now show that in fact  $\epsilon$ -approximation by concatenated polynomials  $\text{CPoly}(\ell, d)$  suffices. This is because we can always do *parallel repetition* on the base NIZK protocol, which has the effect of driving down soundness to be exponentially small. If we repeat the base protocol  $\ell$  times, then a proof  $\pi$  consists of  $(\alpha_1, \ldots, \alpha_\ell, \gamma_1, \ldots, \gamma_\ell)$ , and the overall bad challenge function BadChals<sub>sk</sub> $(\alpha) \rightarrow \beta$  is now a *concatenation* of individual bad challenge functions, i.e., we would compute BadChals<sub>sk</sub> $(\alpha) = BadChal_{sk}(\alpha_1) \| \ldots \| BadChal_{sk}(\alpha_\ell)$ . Therefore, by choosing  $\ell$  to be large enough, we will be able to apply our approximate CIH construction from MQ<sub>*n*,*m*</sub>.

Achieving Statistical Zero-Knowledge. To build NIZKs with statistical zero-knowledge property, we only need to replace the commitment scheme in the Sigma protocol mentioned above with a lossy PKE scheme. This strategy was described in [CCH<sup>+</sup>19], and here we recall the high-level ideas. A lossy PKE has two indistinguishable modes of public keys: normal (where decryption can be carried out), and lossy (where the ciphertext *statistically* loses information about the plaintext). Statistical zero-knowledge then follows from this statistical hiding property: we first switch the lossy key in the CRS to a normal decryption key, then proceed as before.

To make such a strategy work, we need to rely on a lossy PKE that simultaneously has a decryption procedure approximable by linear functions and an inverse-polynomial decryption error probability. Known constructions from LPN with extremely-low noise rate  $O\left(\frac{\log^2 n}{n}\right)$  [BLSV18, BLVW19] do not suffice, because there the decryption error is  $1/2 - 1/\operatorname{poly}(n)$ .

Instead, we build a new lossy PKE based on a recently introduced assumption called the densesparse LPN [DJ24]. On a high-level, this assumption achieves the same lossiness property as LPN with extremely-low noise  $O\left(\frac{\log^2 n}{n}\right)$ , but at a much higher inverse-polynomial noise probability. Once we have this assumption our constructions are structurally very similar to prior constructions of lossy encryptions built from lattices [PVW08]. We refer the reader to Section 6 for details.

### 3 Preliminaries

**Notation.** Let  $\mathbb{N} = \{1, 2, ...\}$  be the natural numbers, and define  $[a, b] := \{a, a + 1, ..., b\}, [n] := [1, n]$ . We denote sampling from a distribution by  $x \leftarrow D$ ; for a finite set S, we write  $x \leftarrow S$  to denote uniformly sampling from S. We denote the security parameter by  $\lambda$ ; our parameters depend on  $\lambda$ , e.g.  $n = n(\lambda)$ , and we often drop the explicit dependence. We write  $negl(\lambda)$  to denote negligible functions in  $\lambda$ .

We abbreviate PPT for probabilistic polynomial-time. Our adversaries are non-uniform PPT algorithms  $\mathcal{A} = \{\mathcal{A}_{\lambda}\}_{\lambda \in \mathbb{N}}$ . Two ensembles of distributions  $\mathcal{D} = \{\mathcal{D}_{\lambda}\}_{\lambda \in \mathbb{N}}$  and  $\mathcal{D}' = \{\mathcal{D}'_{\lambda}\}_{\lambda \in \mathbb{N}}$  are

computationally (resp. statistically) indistinguishable, denoted  $\mathcal{D} \approx_c \mathcal{D}'$  (resp.  $\mathcal{D} \approx_s \mathcal{D}'$ ), if for any non-uniform PPT (resp. unbounded) adversary  $\mathcal{A}$ , there exists a negligible function negl such that  $\mathcal{A}$  can distinguish between the two distributions with probability at most negl( $\lambda$ ).

We denote by  $\mathbb{F}$  a finite field, and  $\mathbb{F}_q$  the finite field with q elements. We will use  $\mathbb{F}_2$  interchangeably with the set  $\{0, 1\}$ . Vector and matrices are written in boldcase, e.g.  $\mathbf{v} \in \mathbb{F}_q^m$  and  $\mathbf{A} \in \mathbb{F}_q^{n \times m}$ . We write  $\Delta(\mathbf{u}, \mathbf{v})$  to denote the Hamming distance (number of different entries) between two vectors  $\mathbf{u}, \mathbf{v} \in \mathbb{F}^n$ , and write  $\|\mathbf{v}\|_0$  to denote the Hamming weight of  $\mathbf{v}$ .

**Bernoulli Distribution.** We denote the Bernoulli distribution over a finite field  $\mathbb{F}_q$  with noise rate  $\epsilon \in (0, 1)$  by  $\text{Ber}(\mathbb{F}_q, \epsilon)$ ; this distribution gives 0 with probability  $1 - \epsilon$ , and a random non-zero element of  $\mathbb{F}_q$  with probability  $\epsilon$ . We omit the field when it is binary ( $\mathbb{F}_q = \mathbb{F}_2$ ).

**Hamming Balls.** We define  $\text{Ball}(n, k, \mathbb{F}_q) = \{\mathbf{x} \in \mathbb{F}_q^n \mid \|\mathbf{x}\|_0 \leq k\}$  to be the Hamming ball of radius k, and omit the field when it is binary ( $\mathbb{F}_q = \mathbb{F}_2$ ). Note that we may uniformly sample from  $\text{Ball}(n, k, \mathbb{F}_q)$  in  $\text{poly}(n, \log q)$  time (for instance, first choose the Hamming weight s of  $\mathbf{x}$  according to the right probability, then uniformly sample  $\mathbf{x}$  of weight s).

We also define the *regular* (binary) Hamming ball

$$\mathsf{Ball}_{\mathsf{reg}}(n,k) := \left\{ \mathbf{x} = \mathbf{x}_1 \| \dots \| \mathbf{x}_k \in \mathbb{F}_2^n \mid \mathbf{x}_i \in \mathbb{F}_2^{n/k} \land \| \mathbf{x}_i \|_0 = 1 \ \forall \ i \in [k] \right\}.$$

There is a simple bijection between  $\mathbb{F}_2^{k \log(n/k)}$  and  $\text{Ball}_{\text{reg}}(n,k)$ , which decomposes  $\mathbf{y} \in \mathbb{F}_2^{k \log(n/k)}$  into chunks  $\mathbf{y}_1 \| \dots \| \mathbf{y}_k$  of length  $\log(n/k)$  each, then turn each chunk into an indicator vector  $\operatorname{ind}(\mathbf{y}_k) \in \mathbb{F}_2^{n/k}$  that has 1 in the  $(\mathbf{y}_k)$ -th entry (interpreted as a number in binary), and 0 otherwise. We call this mapping *sparsification*, and denote it by  $\operatorname{spfy}(\mathbf{y}) := \operatorname{ind}(\mathbf{y}_1) \| \dots \| \operatorname{ind}(\mathbf{y}_k)$ .

**Binary Entropy Function.** The *binary entropy function*  $H : [0,1] \rightarrow [0,1]$  is defined by  $H(x) := -x \log x - (1-x) \log(1-x)$ . It gives an upper bound on the size of the Hamming ball, e.g.,  $|\mathsf{Ball}(n, \epsilon n)| \leq 2^{H(\epsilon)n}$ 

#### 3.1 **Probability Lemmas**

**Lemma 3.1** (Piling-Up Lemma). For any  $\epsilon \in (0, 1)$ , we have that

$$\Pr\left[\sum_{i=1}^{\ell} e_i = 1 \mid e_1, \dots, e_{\ell} \leftarrow \mathsf{Ber}(\epsilon)\right] = \frac{1 - (1 - 2\epsilon)^{\ell}}{2} < \min\left(\epsilon\ell, \frac{1}{2} - 2^{-4\epsilon\ell - 1}\right).$$

**Lemma 3.2** (Chernoff/Hoeffding bound). Let  $X_1, \ldots, X_n \in \{0, 1\}$  be *i.i.d* random variables with mean at most  $\epsilon$ . Then for every  $\kappa \ge 1$ ,

$$\Pr[X_1 + \dots + X_n > (1+\kappa)\epsilon n] \le e^{-2\kappa^2\epsilon n}.$$

**Definition 3.1** (Statistical Difference). *Given two random variables* X, Y *over the same underlying finite set* U, we define their statistical difference to be

$$SD(X,Y) = \frac{1}{2} \sum_{u \in \mathcal{U}} |\Pr[X = u] - \Pr[Y = u]|$$

**Definition 3.2** (Min-entropy). For a random variable X taking range over a finite set X, we define its average min-entropy to be

$$H_{\infty}(X) = -\log \max_{x \in \mathcal{X}} \Pr[X = x].$$

Given two random variables X, Y taking range over finite sets  $\mathcal{X}, \mathcal{Y}$  respectively, and any  $y \in \mathcal{Y}$ , we define the conditional min-entropy of X given Y = y to be

$$H_{\infty}(X \mid Y = y) = -\log \max_{x \in \mathcal{X}} \Pr[X = x \mid Y = y].$$

We define the average conditional min-entropy of X given Y to be

$$H_{\infty}(X \mid Y) = -\log \mathbb{E}_{y \leftarrow \mathcal{Y}} \left[ \max_{x \in \mathcal{X}} \Pr[X = x \mid Y = y] \right] = -\log \mathbb{E}_{y \leftarrow \mathcal{Y}} \left[ 2^{-H_{\infty}(X \mid Y = y)} \right].$$

**Lemma 3.3.** Let X, Y be random variables where Y has range in  $\mathcal{Y}$ . Then we have  $\mathbf{H}_{\infty}(X \mid Y) \geq \mathbf{H}_{\infty}(X) - \log|\mathcal{Y}|$ .

**Leftover Hash Lemma.** A family of hash functions  $\{h : \mathcal{X} \to \mathcal{Y}\}_{h \in \mathcal{H}}$  is called *universal* if for all  $x \neq x' \in \mathcal{X}$ , we have  $\Pr_{h \leftarrow \mathcal{H}}[h(x) = h(x')] \leq \frac{1}{|\mathcal{Y}|}$ . We need the following *generalized* leftover hash lemma from [DRS04].

**Lemma 3.4** (LHL with leakage [DRS04], restated). Assume  $\{h : \mathcal{X} \to \mathcal{Y}\}_{h \in \mathcal{H}}$  is a universal family of hash functions. Then for any random variables X taking range in  $\mathcal{X}$  and Z possibly dependent on X, we have

$$SD((h, h(X), Z), (h, U(\mathcal{Y}), Z)) \le \frac{1}{2} \sqrt{\frac{|\mathcal{Y}|}{2^{H_{\infty}(X|Z)}}}.$$

#### 3.2 Relations

**Definition 3.3** (Searchable Relations). For  $n, m \in \mathbb{N}$ , we say that a relation  $\mathcal{R} \subseteq \{0, 1\}^n \times \{0, 1\}^m$  is searchable by a function class  $\mathcal{F} = \{0, 1\}^n \to \{0, 1\}^m \cup \bot$  if the following two conditions are satisfied:

- It is unique-output, namely that for every x ∈ {0,1}<sup>n</sup> there exists at most one y ∈ {0,1}<sup>m</sup> such that (x, y) ∈ R. We denote R : {0,1}<sup>n</sup> → {0,1}<sup>m</sup> ∪ ⊥ to be the unique (partial) function such that R(x) = y if there exists y such that (x, y) ∈ R, and R(x) = ⊥ otherwise.
- For every  $R \in \mathcal{R}$ , there exists a function  $f_R \in \mathcal{F}$  such that for all  $\mathbf{x}$  with  $R(\mathbf{x}) \neq \bot$ , we have  $(\mathbf{x}, f_R(\mathbf{x})) \in R$ .

**Definition 3.4** (Probabilistic Representation [BKM20]). Let  $n, m \in \mathbb{N}$  and  $\epsilon \in (0, 1)$ . Let  $f : \{0, 1\}^n \to \{0, 1\}^m \cup \bot$  be a function with  $f(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))$ , where  $f_i : \{0, 1\}^n \to \{0, 1\} \cup \bot$  for all  $i \in [m]$ . A (bit-wise)  $\epsilon$ -probabilistic representation of f by a class of functions  $C : \{0, 1\}^n \to \{0, 1\}$  consists of m distributions  $\mathfrak{C}_1, \dots, \mathfrak{C}_m \subseteq C$  that satisfy the following:

$$\forall i \in [m], \ \forall \mathbf{x} \ such \ that \ f(\mathbf{x}) \neq \bot, \quad \Pr_{C_i \leftarrow \mathfrak{C}_i}[f_i(\mathbf{x}) = C_i(\mathbf{x})] > 1 - \epsilon.$$

Using Chernoff bound, we immediately get the corollary (also found in [BKM20]) that f is (point-wise) well-approximated by C.

**Lemma 3.5.** Let  $n = n(\lambda)$ ,  $m = m(\lambda)$ ,  $\epsilon = \epsilon(\lambda) \in (0,1)$  be polynomial-time computable functions in  $\lambda$  such that  $\epsilon m \gg \lambda$ . Let  $f = \{f_{\lambda} : \{0,1\}^{n(\lambda)} \rightarrow \{0,1\}^{m(\lambda)} \cup \bot\}_{\lambda \in \mathbb{N}}$  be a function ensemble and  $\mathcal{C} = \{\mathcal{C}_{\lambda} : \{0,1\}^{n(\lambda)} \rightarrow \{0,1\}\}$  be a function class ensemble such that for every  $\lambda \in \mathbb{N}$ ,  $f_{\lambda}$  has a  $\epsilon$ -probabilistic representation by  $\mathfrak{C}_{\lambda,1}, \ldots, \mathfrak{C}_{\lambda,m} \subseteq \mathcal{C}_{\lambda}$ . Then for all inputs  $\mathbf{x} = \{\mathbf{x}_{\lambda}\}_{\lambda \in \mathbb{N}}$  such that  $f_{\lambda}(\mathbf{x}_{\lambda}) \neq \bot$  for all  $\lambda \in \mathbb{N}$ , we have

$$\Pr_{C = (C_i)_{i=1}^m \leftarrow \prod_{i=1}^m \mathfrak{C}_{\lambda,i}} \left[ \Delta \left( f_\lambda(\mathbf{x}_\lambda), C(\mathbf{x}_\lambda) \right) > 2\epsilon m \right] < \mathsf{negl}(\lambda).$$

**Definition 3.5** (Probabilistically Searchable Relations). A relation  $\mathcal{R}$  is  $\epsilon$ -probabilistically searchable by a function class  $\mathcal{C}$  if it is searchable by some function class  $\mathcal{F}$ , and for every  $R \in \mathcal{R}$ , letting  $f_R \in \mathcal{F}$  be the corresponding search function, we have that  $f_R$  has an  $\epsilon$ -probabilistic representation by  $\mathcal{C}$ .

#### 3.3 Correlation Intractability

We start by defining the syntax of a hash family  $\mathcal{H} = (\text{Gen}, \text{Hash})$  with input length  $n(\lambda)$  and output length  $m(\lambda)$  (we require that m < n):

- Gen $(1^{\lambda}) \rightarrow$  hk. A PPT algorithm that on input the security parameter  $1^{\lambda}$  returns a key hk.
- Hash(hk, x) → h. A deterministic poly-time algorithm that on input a key hk and an element x ∈ {0,1}<sup>n(λ)</sup>, returns a hash output h ∈ {0,1}<sup>m(λ)</sup>.

We now define various notions of correlation intractability for  $\mathcal{H}$ .

**Definition 3.6** (Correlation Intractability [CGH04]). *A hash family*  $\mathcal{H}$  *is said to be* correlation intractable (CI) *for a relation family*  $\mathcal{R} = \{\mathcal{R}_{\lambda}\}_{\lambda \in \mathbb{N}}$  *if for every PPT adversary*  $\mathcal{A} = \{\mathcal{A}_{\lambda}\}_{\lambda \in \mathbb{N}}$ *, there exists a negligible function* negl( $\lambda$ ) *such that for every*  $R \in \mathcal{R}_{\lambda}$ *,* 

$$\mathbf{Adv}^{\mathsf{CI}}_{\mathcal{H},R}(\mathcal{A}) := \Pr\left[ (\mathbf{x}, \mathcal{H}.\mathsf{Hash}(\mathsf{hk}, \mathbf{x})) \in R \; \middle| \; \begin{array}{c} \mathsf{hk} \leftarrow \mathcal{H}.\mathsf{Gen}(1^{\lambda}) \\ \mathbf{x} \leftarrow \mathcal{A}_{\lambda}(\mathsf{hk}) \end{array} \right] \leq \mathsf{negl}(\lambda).$$

**Definition 3.7** (Programmability [CCH<sup>+</sup>19]). A hash family  $\mathcal{H}$  with input length  $n(\lambda)$  and output length  $m(\lambda)$  is said to be programmable if there exists a PPT algorithm Sim such that the following conditions hold for every  $\lambda \in \mathbb{N}$ ,  $\mathbf{x} \in \{0, 1\}^n$ , and  $\mathbf{h} \in \{0, 1\}^m$ :

• 1-Universality. We have

$$\Pr\left[\mathcal{H}.\mathsf{Hash}(\mathsf{hk},\mathbf{x})=\mathsf{h}\;\middle|\;\mathsf{hk}\leftarrow\mathcal{H}.\mathsf{Gen}(1^{\lambda})\right]=2^{-m}$$

• **Programmability.**  $Sim(1^{\lambda}, \mathbf{x}, \mathbf{h})$  samples from the conditional distribution

$$\left\{\mathsf{hk} \leftarrow \mathcal{H}.\mathsf{Gen}(1^{\lambda}) \mid \mathcal{H}.\mathsf{Hash}(\mathsf{hk}, \mathbf{x}) = \mathsf{h}\right\}.$$

**Definition 3.8** (CI for Approximable Relations [BKM20]). Let  $C = \{C_{\lambda} : \{0,1\}^{n(\lambda)} \rightarrow \{0,1\}^{m(\lambda)}\}_{\lambda \in \mathbb{N}}$ be a function class and let  $\epsilon \in (0,1)$ . For every  $C \in C$ , we define the relation  $\epsilon$ -approximable by C to be

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

A hash family  $\mathcal{H}$  that is CI for all relations  $\{R_C^{\epsilon} \mid C \in C\}$  is said to satisfy  $\epsilon$ -approximate CI (CI-Apx<sub> $\epsilon$ </sub>) for C.

It is known that approximate CI for a function class C implies (exact) CI for any function class approximable by C.

**Lemma 3.6** (Theorem 4.2 in [BKM20]). Let  $\mathcal{F}$  be a function class that has a  $\epsilon$ -probabilistic representation by C. If  $\mathcal{H}$  is a hash function satisfying CI-Apx<sub>2 $\epsilon$ </sub> for C, then  $\mathcal{H}$  satisfies CI for relations searchable by  $\mathcal{F}$ .

### 3.4 Non-Interactive Zero-Knowledge

**Definition 3.9** (NIZK). A non-interactive zero knowledge (NIZK) argument  $\Pi$  for a NP relation  $\mathcal{R} = \{\mathcal{R}_{\lambda}\}_{\lambda \in \mathbb{N}}$ , with instance length  $n = n(\lambda)$  and associated NP language  $\mathcal{L} = \{\mathcal{L}_{\lambda}\}_{\lambda \in \mathbb{N}}$ , is a tuple of three PPT algorithms NIZK = (Setup,  $\mathcal{P}, \mathcal{V}$ ) satisfying the following properties:

- Syntax:
  - Setup $(1^{\lambda}) \rightarrow$  crs. On input the security parameter  $1^{\lambda}$ , output a common reference string crs.
  - $\mathcal{P}(\operatorname{crs}, x, w) \to \pi$ . On input the common reference string crs and an instance-witness pair  $(x, w) \in \mathcal{R}_{\lambda}$  with  $x \in \{0, 1\}^{n(\lambda)}$ , output a proof  $\pi$ .
  - $\mathcal{V}(\operatorname{crs}, x, \pi) \to b$ . On input the common reference string crs, an instance  $x \in \{0, 1\}^{n(\lambda)}$  and a proof  $\pi$ , outputs a bit b indicating acceptance or rejection.
- **Completeness:** For every  $\lambda \in \mathbb{N}$  and  $(x, w) \in \mathcal{R}_{\lambda}$ , there exists a negligible function  $\operatorname{negl}(\lambda)$  such that

$$\Pr\left[\mathcal{V}(\mathsf{crs}, x, \pi) = 1 \middle| \begin{array}{c} \mathsf{crs} \leftarrow \mathsf{Setup}(1^{\lambda}) \\ \pi \leftarrow \mathcal{P}(\mathsf{crs}, x, w) \end{array} \right] \ge 1 - \mathsf{negl}(\lambda).$$

Adaptive Computational Soundness: For every polynomial-size adversary P\*, there exists a negligible function negl(λ) such that

$$\Pr\begin{bmatrix} x \in \{0,1\}^n \land x \notin \mathcal{L} \\ \land \mathcal{V}(\mathsf{crs}, x, \pi) = 1 \end{bmatrix} \stackrel{\mathsf{crs}}{\underset{(x,\pi)}{\mathsf{crs}}} \stackrel{\mathsf{Setup}(1^\lambda)}{\underset{(x,\pi)}{\mathsf{crs}}} \leq \mathsf{negl}(\lambda).$$

• Adaptive Computational Zero-Knowledge: There exists a stateful PPT simulator S such that for any polynomial-size adversary A, there exists a negligible function  $negl(\lambda)$  such that

$$\left|\Pr[\mathsf{Real}_{\mathcal{A}}(1^{\lambda}) = 1] - \Pr[\mathsf{Ideal}_{\mathcal{S},\mathcal{A}}(1^{\lambda}) = 1]\right| \leq \mathsf{negl}(\lambda).$$

Here the games are defined as follows:

Real<sub> $\mathcal{A}$ </sub>(1<sup> $\lambda$ </sup>): 1. crs  $\leftarrow$  Setup(1<sup> $\lambda$ </sup>) 2. (x, w)  $\leftarrow \mathcal{A}(crs)$ If (x, w)  $\notin \mathcal{R}_{\lambda}$ , output 0. 3.  $\pi \leftarrow \mathcal{P}(crs, x, w)$ 4. Output  $\mathcal{A}(\pi)$  Ideal<sub>S,A</sub>(1<sup> $\lambda$ </sup>): 1. crs  $\leftarrow S(1^{\lambda})$ 2. (x,w)  $\leftarrow A(crs)$ If (x,w)  $\notin \mathcal{R}_{\lambda}$ , output 0. 3.  $\pi \leftarrow S(crs, x)$ 4. Output  $A(\pi)$ 

If the above holds for an unbounded adversary A, we say that NIZK satisfies adaptive statistical zero-knowledge.

### 3.5 Public-Key Encryption

**Definition 3.10** (PKE). A public-key (bit) encryption *scheme is a tuple of PPT algorithms* PKE = (Gen, Enc, Dec) *with the following properties:* 

- Syntax:
  - $Gen(1^{\lambda}) \rightarrow (pk, sk)$ . On input the security parameter  $1^{\lambda}$ , output a public key pk and a secret key sk.
  - $Enc(pk, \mu) \rightarrow ct$ . On input the public key pk and a message  $\mu \in \{0, 1\}$ , output a ciphertext ct.
  - $Dec(sk, ct) \rightarrow \mu \in \{0, 1\} \cup \{\bot\}$ . On input the secret key sk and the commitment ct, output a message  $\mu \in \{0, 1\}$  or failure  $\bot$ .
- *Correctness.* For every λ ∈ N and (pk,td) ← Gen(1<sup>λ</sup>), there exists a negligible function negl(λ) such that for every μ ∈ {0,1}:

$$\Pr\left[\mathsf{Dec}(\mathsf{sk},\mathsf{Enc}(\mathsf{pk},\mu))=\mu\Big|(\mathsf{pk},\mathsf{sk})\leftarrow\mathsf{Gen}(1^{\lambda})\right]\geq 1-\mathsf{negl}(\lambda).$$

• Semantic Security. The following two distributions are computationally indistinguishable:

 $\left\{\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, 0) \mid (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^{\lambda})\right\} \approx_c \left\{\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, 1) \mid (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^{\lambda})\right\}.$ 

**Definition 3.11** (Lossy PKE). A public-key encryption scheme PKE is called lossy if there exists a PPT algorithm  $\text{LossyGen}(1^{\lambda}) \rightarrow \widetilde{\text{pk}}$  satisfying the following:

• Statistical Hiding. The following two distributions are statistically indistinguishable:

$$\left\{\mathsf{ct} \leftarrow \mathsf{Enc}(\widetilde{\mathsf{pk}}, 0) \ \Big| \ \widetilde{\mathsf{pk}} \leftarrow \mathsf{LossyGen}(1^{\lambda}) \right\} \approx_s \left\{\mathsf{ct} \leftarrow \mathsf{Enc}(\widetilde{\mathsf{pk}}, 1) \ \Big| \ \widetilde{\mathsf{pk}} \leftarrow \mathsf{LossyGen}(1^{\lambda}) \right\}.$$

• *Mode Indistinguishability.* The following two distributions are computationally indistinguishable:

$$\left\{\mathsf{pk} \mid (\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{Gen}(1^{\lambda})\right\} \approx_c \left\{\widetilde{\mathsf{pk}} \mid \widetilde{\mathsf{pk}} \leftarrow \mathsf{LossyGen}(1^{\lambda})\right\}$$

### 3.6 Commit-then-Open Protocols

We define commit-then-open  $\Sigma$ -protocols as in [BKM20], including properties that are sufficient for instantiating NIZK with adaptive security. Examples of such protocols for an NP-complete language (such as Graph Hamiltonicity) can be found in [Blu86, FLS90, CCH<sup>+</sup>19].

**Definition 3.12** (Commit-then-Open). A commit-then-open  $\Sigma$ -protocol for an NP language L is a public-coin, 3-message, honest-verifier zero-knowledge argument  $\Pi = (\text{Setup}, \mathcal{P}, \mathcal{V})$  for L, relying on a public-key encryption scheme PKE = (Gen, Enc, Dec), where the verifier's challenge is a single bit  $c \in \{0, 1\}$ , and with the following requirements:

- *Auxiliary Algorithms.* There exists four PPT algorithms with the following syntax:
  - Setup' $(1^{\lambda}, pk) \rightarrow crs'$ . On input the security parameter  $1^{\lambda}$  and encryption key pk, outputs auxiliary common reference string crs'.
  - $\mathsf{P}_1(\mathsf{crs}, x, w) \to (a', \pi, \mathsf{st}')$ . On input  $\mathsf{crs}$ , and instance x, and witness w, outputs auxiliary information a', underlying proof string to be encrypted  $\pi \in \{0, 1\}^{\ell}$ , and state  $\mathsf{st}'$ .

- $\mathsf{P}'(\mathsf{crs}, x, w, \mathsf{st}') \to I$ . Takes input  $\mathsf{crs}, x, w, \mathsf{st}$  as above, and outputs a subset  $I \subseteq [\ell]$ .
- $V'(crs, x, (I, \pi_I)) \rightarrow b$ . Takes as input  $crs, x, I \subseteq [\ell]$  as above, together with the proof substring  $\pi_I$ , and returns a bit  $b \in \{0, 1\}$ .
- Format. The protocol is of the following form:
  - Setup $(1^{\lambda}) \rightarrow$  crs. Sample a public key  $(pk, sk) \leftarrow PKE.Gen(1^{\lambda})$  and possibly additional output crs'  $\leftarrow$  Setup' $(1^{\lambda}, pk)$ , and return crs = (crs', pk).
  - $\mathcal{P}(\operatorname{crs}, x, w) \to (a, \operatorname{st})$ . The prover computes  $(a', \pi, \operatorname{st}') \leftarrow \mathsf{P}_1(\operatorname{crs}, x, w)$ , then encrypts  $\operatorname{ct} \leftarrow \mathsf{PKE}.\mathsf{Enc}(\mathsf{pk}, \pi; r)$ . It sets the first message to be  $a = (a', \operatorname{ct})$ , and sets its state as  $\operatorname{st} = (\pi, \operatorname{st}', r)$ .
  - $\mathcal{P}(\operatorname{crs}, x, w, \operatorname{st}, c) \to z$ . Upon receiving a random challenge  $c \in \{0, 1\}$ , if c = 1, the prover computes the opening slots  $I \leftarrow \mathsf{P}'(\operatorname{crs}, x, w, \operatorname{st}')$ , then retrieves the proof substring  $\pi_I$  along with the associated encryption randomness  $r_I$ . It sets the third message to be  $z = (I, \pi_I, r_I)$ . There is no restriction for c = 0.
  - $\mathcal{V}(\operatorname{crs}, x, a, c, z) \to b \in \{0, 1\}$ . Upon receiving all messages, if c = 1, the verifier checks that the encryptions are correct  $\operatorname{ct}_I = \operatorname{Enc}(\operatorname{pk}, \pi_I; r_I)$ , runs the auxiliary verification procedure  $b \leftarrow \mathsf{V}'(\operatorname{crs}, x, (I, \pi_I))$ , then outputs b. There is no restriction for c = 0.

We define the following properties for  $\Pi$ :

• Instance-Universal. The bad challenge relation

 $\mathcal{R}_{\mathsf{crs}} := \{(a, c) \mid \exists x \notin L \text{ and } z \text{ such that } \mathcal{V}(\mathsf{crs}, x, a, c, z) = 1\}$ 

*is unique-output (Definition 3.3), and the computation of* P' *and* V' *are independent of* x, w, and the *prover's state* st'.

• Unique Bad Challenge. Given that Π is instance-universal, we require that the bad challenge relation

 $\mathcal{R}_{\mathsf{crs}} := \{(a, c) \mid \exists \ x \notin L \text{ and } z \text{ such that } \mathcal{V}(\mathsf{crs}, x, a, c, z) = 1\}$ 

is searchable by the function  $BadChal_{sk}(a)$  (where sk is generated during Setup) that is of the following form:

- 1. Parse a = (a', ct), then compute  $\mathsf{PKE}.\mathsf{Dec}(\mathsf{sk}, ct) \to \pi \in \{0, 1, \bot\}^*$ .
- 2. If  $\pi \in \{0,1\}^*$ , then compute  $\mathsf{P}'(\mathsf{crs}) \to I$  and  $\mathsf{V}'(\mathsf{crs},(I,\pi_I)) \to b$ , and return 1 if b = 1.
- 3. Else return 0.
- **Delayed-Input.** The computation of P<sub>1</sub> is independent of the instance x and witness w.
- 3CNF Verification. The verification procedure V'(crs, x, (I, π<sub>I</sub>)) is the evaluation of a 3CNF formula on (I, π<sub>I</sub>).

We now recall the transformation in [BKM20] that takes in a commit-then-open  $\Sigma$ -protocol and turn it into a related protocol  $\widetilde{\Pi}$  with 3CNF verification.<sup>8</sup>

<sup>&</sup>lt;sup>8</sup>At a high level, the transformation uses a Cook-Levin reduction to turn  $C_{crs,x}(I, \pi_I) := V'(crs, x, (I, \pi_I))$  into a 3CNF formula  $\Phi_{crs,x}$ , such that  $C_{crs,x}(I, \pi_I) = 1$  if and only if there exists a witness w so that  $\Phi_{crs,x}(I, \pi_I, w) = 1$ . The prover of  $\widetilde{\Pi}$  would compute this witness w and send its commitment in the first round, opening the commitment if the challenge is c = 1.

**Lemma 3.7** (Implicit in Theorem 3.16 of [BKM20]). Let  $\Pi$  be a commit-then-open  $\Sigma$ -protocol for a language L. Then there exists an efficient transformation of  $\Pi$  into a new commit-then-open  $\Sigma$ -protocol  $\widetilde{\Pi}$  for the same language L, but with 3CNF verification. The transformation preserves the instance-universal, unique bad challenge, and delayed-input properties.

# 4 Assumptions

### 4.1 Multivariate Cryptography

**Notation.** Let  $m, n, d \in \mathbb{N}$  and  $\mathbb{F}$  be a finite field. For a tuple of degrees  $\vec{\alpha} = (\alpha_1, \dots, \alpha_n) \in \mathbb{N}^n$ and a vector  $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{F}^n$ , we write  $\mathbf{x}^{\vec{\alpha}} := \prod_{i=1}^n x_i^{\alpha_i} \in \mathbb{F}$ . We also define  $|\vec{\alpha}| := \sum_{i=1}^n \alpha_i$ to be the (*total*) *degree* of  $\vec{\alpha}$ . A polynomial system  $\mathscr{P}$  of degree (at most) d over  $\mathbb{F}$ , with n variables and m equations, is a tuple

$$\left(\left(p_{j,\vec{\boldsymbol{\alpha}}}\right)_{|\vec{\boldsymbol{\alpha}}|=d},\ldots,\left(p_{j,\vec{\boldsymbol{\alpha}}}\right)_{|\vec{\boldsymbol{\alpha}}|=1},p_{j,\mathbf{0}}\right)_{j\in[m]}\in\left(\mathbb{F}^{\binom{n+d-1}{d}}\times\cdots\times\mathbb{F}^{n}\times\mathbb{F}\right)^{m}.$$

More compactly, we may write

$$\mathscr{P} = \left(p_{j,\vec{\alpha}}\right)_{|\vec{\alpha}| \le d, j \in [m]} \in \mathsf{Poly}(n, m, d, \mathbb{F}) := \left(\mathbb{F}^{\binom{n+d}{d}}\right)^m$$

We may evaluate the system on any input  $\mathbf{x} \in \mathbb{F}^n$ , giving output  $\mathscr{P}(\mathbf{x}) \in \mathbb{F}^m$ . We say that  $\mathbf{x}$  is a solution to  $\mathscr{P}$  if  $\mathscr{P}(\mathbf{x}) = \mathbf{0}$ , or more explicitly:

$$\begin{cases} \sum_{|\vec{\boldsymbol{\alpha}}| \leq d} p_{1,\vec{\boldsymbol{\alpha}}} \cdot \mathbf{x}^{\vec{\boldsymbol{\alpha}}} = 0, \\ \vdots \\ \sum_{|\vec{\boldsymbol{\alpha}}| \leq d} p_{m,\vec{\boldsymbol{\alpha}}} \cdot \mathbf{x}^{\vec{\boldsymbol{\alpha}}} = 0. \end{cases}$$

The central assumption in multivariate cryptography is that it is difficult to solve a random system of polynomial equations. In this work, we also consider the problem of *approximately* solving a system of polynomial equations, where a solution only needs to satisfy some  $(1 - \epsilon)$ -fraction of all equations. We will state the most general assumption of this form and say how it specializes to other assumptions that are more well-studied in the literature.

**Definition 4.1** ((Approximate) Multivariate Polynomial Solving). Let  $d \ge 2$  be a constant,  $\mathbb{F}$  be a finite field,  $\epsilon = \epsilon(\lambda) \in [0, 1)$  be the error rate, and  $n = n(\lambda)$ ,  $m = m(\lambda)$  be polynomials in  $\lambda$ . We say that the approximate multivariate polynomial solving assumption Apx-MPS<sub> $n,m,d,\epsilon,\mathbb{F}$ </sub> is  $(T(\lambda), \delta(\lambda))$ -hard if for every adversary  $\mathcal{A}$  running in time at most T, the following holds for every  $\lambda \in \mathbb{N}$ :

$$\mathbf{Adv}_{n,m,d,\epsilon,\mathbb{F}}^{\mathsf{Apx}\mathsf{-}\mathsf{MPS}}(\mathcal{A}) := \Pr\left[\Delta\left(\mathscr{P}(\mathbf{x}), \mathbf{0}\right) \le \epsilon m \; \middle| \; \begin{array}{c} \mathscr{P} \leftarrow \mathsf{Poly}(n,m,d,\mathbb{F}) \\ \mathbf{x} \leftarrow \mathcal{A}(\mathscr{P}) \end{array} \right] \le \delta(\lambda).$$

When  $\epsilon = 0$ , we get the multivariate polynomial solving assumption MPS<sub>*n*,*m*,*d*, $\mathbb{F}$ . When d = 2 and  $\mathbb{F} = \mathbb{F}_2$ , we get the approximate multivariate quadratics assumption Apx-MQ<sub>*n*,*m*, $\epsilon$ , and furthermore when  $\epsilon = 0$ , we get the multivariate quadratics assumption MQ<sub>*n*,*m*</sub>.</sub></sub>

In our applications, we will work in the *underdetermined* regime of MPS or MQ, where  $n \gg m$ . In this regime, as long as n is sufficiently larger than m, the image of a random system of polynomial equations would be relatively "well-spread", and hence a solution exists with overwhelming probability. **Lemma 4.1.** Let  $n, m, d \in \mathbb{N}$  and  $\mathbb{F} = \mathbb{F}_q$  be a finite field. When n > 5m, over the random choice of  $\mathscr{P} \leftarrow \mathsf{Poly}(n, m, d, \mathbb{F})$ , the system of equations  $\mathscr{P}(\mathbf{x}) = \mathbf{t}$  has a solution  $\mathbf{x} \in \mathbb{F}^n$  for every  $\mathbf{t} \in \mathbb{F}^m$  with probability at least  $1 - q^{-m}$ .

*Proof.* We define the *d*-th symmetric tensor product of a vector  $\mathbf{x} \in \mathbb{F}^n$  as  $\text{Sym}_d(\mathbf{x}) = (\mathbf{x}^{\vec{\alpha}})_{|\vec{\alpha}|=d} \in \mathbb{F}^{\binom{n+d-1}{d}}$ . Define  $\text{Sym}_{\leq d}(\mathbf{x}) = (\text{Sym}_d(\mathbf{x}), \dots, \text{Sym}_2(\mathbf{x}), \mathbf{x}, 1)$ . For any  $\mathbf{t} \in \mathbb{F}^m$  (expressed as a row vector), we can rewrite the system of equations  $\mathscr{P}(\mathbf{x}) = \mathbf{t}$  as a vector-matrix multiplication:

$$\mathsf{Sym}_{\leq d}(\mathbf{x}) \cdot \mathsf{Mat}(\mathscr{P}) = \mathbf{t}, \text{ where } \mathsf{Mat}(\mathscr{P}) := \begin{pmatrix} (p_{1,\vec{\alpha}})_{|\vec{\alpha}|\leq d}^\top & \|\dots\| & (p_{m,\vec{\alpha}})_{|\vec{\alpha}|\leq d}^\top \end{pmatrix}$$

Note that the mapping  $\mathbf{x} \mapsto \text{Sym}_{\leq d}(\mathbf{x})$  is an injection, hence we have that  $H_{\infty}(\text{Sym}_{\leq d}(\mathbf{x})) = H_{\infty}(\mathbf{x}) = n \log q > 5m \log q$ . Since the coefficients of  $\mathscr{P}$  are chosen randomly, we can apply the Leftover Hash Lemma to get that (for uniformly random  $\mathbf{u} \leftarrow \mathbb{F}^m$ ):

$$\mathop{\mathbb{E}}_{\mathscr{P}}\left[\mathbf{SD}(\mathscr{P}(\mathbf{x}),\mathbf{u})\right] = \mathbf{SD}\left(\left(\mathscr{P},\mathsf{Sym}_{\leq d}(\mathbf{x})\cdot\mathsf{Mat}(\mathscr{P})\right),(\mathscr{P},\mathbf{u})\right) \leq q^{-2m}.$$

If the equation  $Sym_{\leq d}(\mathbf{x}) \cdot Mat(\mathscr{P}) = \mathbf{t}$  have no solution for some  $\mathbf{t} \in \mathbb{F}^m$ , then

$$\mathbf{SD}(\mathscr{P}(\mathbf{x}),\mathbf{u}) = \sum_{\mathbf{u}\in\mathbb{F}^m} \left| \Pr_{\mathbf{x}\leftarrow\mathbb{F}^m} \left[ \mathsf{Sym}_{\leq d}(\mathbf{x})\cdot\mathsf{Mat}(\mathscr{P}) = \mathbf{u} \right] - q^{-m} \right| \geq q^{-m},$$

since one of the probabilities (for  $\mathbf{u} = \mathbf{t}$ ) will be 0. This cannot happen except for at most  $q^{-m}$  fraction of  $\mathscr{P}$ , which gives the desired conclusion.

We now provide a reduction from the *exact* problem  $MPS_{n,m,d,\mathbb{F}_2}$  to the *approximate* problem  $Apx-MPS_{n,m,d,\epsilon,\mathbb{F}_2}$ , with a loss in success probability that is *exponential* in *m* (assuming  $\epsilon > 0$  is a constant). We give this reduction in the case of the binary field, though it also generalizes to arbitrary finite field  $\mathbb{F}_q$  (where the reduction loss additionally depends on *q*).

**Lemma 4.2.** For every adversary A running in time T against Apx-MPS<sub> $n,m,d,\epsilon,\mathbb{F}_2$ </sub>, there exists an adversary B running in time  $T' \approx T$  against MPS<sub> $n,m,d,\mathbb{F}_2$ </sub> such that

$$\mathbf{Adv}_{n,m,d,\epsilon,\mathbb{F}_2}^{\mathsf{Apx-MPS}}(\mathcal{A}) \leq 2^{H(\epsilon)m} \cdot \mathbf{Adv}_{n,m,d,\mathbb{F}_2}^{\mathsf{MPS}}(\mathcal{B}).$$

In particular, for a constant  $\epsilon > 0$  and every  $T = \text{poly}(\lambda)$ , if  $\text{MPS}_{n,m,d,\mathbb{F}_2}$  is  $(T, 2^{-H(\epsilon)m} \cdot \text{negl}(m))$ -hard, then  $\text{Apx-MPS}_{n,m,d,\epsilon,\mathbb{F}_2}$  is (T', negl(m))-hard, where  $T' = T + p(\lambda)$  for some fixed polynomial p.

*Proof.* From an adversary A, we build the adversary B as follows:

- 1.  $\mathcal{B}$  receives the MPS instance  $\mathscr{P} \leftarrow \mathsf{Poly}(n, m, d, \mathbb{F}_2)$  and samples a noise vector  $\mathbf{e} \leftarrow \mathsf{Ball}(m, \epsilon m)$ .
- B runs A on the MPS instance P' := P + e (where addition occurs at the constant terms, i.e., P'(x) = P(x) + e for all x), and receives A's output x'.
- 3.  $\mathcal{B}$  returns x'.

Note that  $\mathcal{B}$  runs in almost the same time as  $\mathcal{A}$ , plus the sampling of e. From the description of  $\mathcal{B}$ , it is clear that  $\mathcal{B}$  wins  $MPS_{n,m,d,\mathbb{F}_2}$ , meaning that  $\mathscr{P}(\mathbf{x}') = \mathbf{0}$ , if both of the following events happen:

<sup>&</sup>lt;sup>9</sup>Recall that uniform sampling from a Hamming ball is possible in polynomial time (See Section 3).

- $\mathcal{A}$  wins Apx-MPS<sub>*n*,*m*,*d*, $\epsilon,\mathbb{F}_2$ , meaning there exists some  $\mathbf{e}' \in \mathbb{F}_2^m$  such that  $\mathscr{P}'(\mathbf{x}') = \mathbf{e}'$  and  $\|\mathbf{e}'\|_0 \leq \epsilon m$ ,</sub>
- $\mathcal{B}$  correctly guesses  $\mathbf{e} = \mathbf{e}'$ .

We note that these events are independent. This is because e is independently sampled from  $\mathscr{P}$ , and conditioned on any fixed e we still have that  $\mathscr{P}'$  is uniformly random over the randomness of  $\mathscr{P}$ . The first event has probability exactly  $\operatorname{Adv}_{n,m,d,\epsilon,\mathbb{F}_2}^{\operatorname{Apx-MPS}}(\mathcal{A})$ . We can calculate the probability of the second event using the following simple result.

**Claim 4.1.** Let X be a finite set,  $\mathcal{D}$  be an arbitrary distribution on X, and  $\mathcal{U}_X$  be the uniform distribution from X. Then  $\Pr[x = y \mid x \leftarrow \mathcal{D}, y \leftarrow \mathcal{U}_X] = \frac{1}{|X|}$ .

Proof. We have

$$\Pr[x = y \mid x \leftarrow \mathcal{D}, y \leftarrow \mathcal{U}_X] = \frac{1}{|X|} \sum_{y \in X} \Pr[x = y \mid x \leftarrow \mathcal{D}] = \frac{1}{|X|}.$$

Using the claim with  $\mathcal{D}$  being the distribution of e' generated by  $\mathcal{A}$  (conditioned on the event that  $\mathcal{A}$  wins Apx-MPS), it follows that the second event happens with probability  $\frac{1}{|\mathsf{Ball}(m,\epsilon m)|} \geq \frac{1}{2^{H(\epsilon)m}}$ . The conclusion follows since

$$\mathbf{Adv}_{n,m,d,\mathbb{F}_{2}}^{\mathsf{MPS}}(\mathcal{B}) \geq \mathbf{Adv}_{n,m,d,\epsilon,\mathbb{F}_{2}}^{\mathsf{Apx}-\mathsf{MPS}}(\mathcal{A}) \cdot \frac{1}{2^{H(\epsilon)m}}.$$

Security of MQ and MPS. We now summarize the best known algorithms for solving underdetermined constant-degree systems of equations, and explain why  $MQ_{n,m}$  and  $MPS_{n,m,d}$  are plausibly  $(poly(m), 2^{-\Omega(m)})$ -secure for our parameter regime:

- 1. For quadratic equations, the best known attack over an arbitrary field  $\mathbb{F}_q$  is by Cheng, Hashimoto, Miura, and Takagi [CHMT14], improving over a sequence of prior works [MHT13, TW12, CGMT02, KPG99]. These attacks have time complexity  $2^{\Omega(m-n/m)}$ ; in other words, they run in polynomial time when  $m = \Theta(\sqrt{n})$ , and exponential time once  $m = n^{1/2+\delta}$  for *any* constant  $0 < \delta < 1/2$ . As  $MQ_{n,m}$  is a "natural" problem, we expect that  $(2^{\Omega(m)}, \operatorname{negl}(m))$ -security should translate to  $(\operatorname{poly}(m), 2^{-\Omega(m)})$ -security, following our discussion after Theorem 1.1.
- 2. The threshold  $m = \omega(\sqrt{n})$  also seems to hold for higher-degree equations, for the (intuitive) reason that they should be *harder* to solve than quadratic ones.<sup>10</sup> However, we do not know of any reference for the complexity of solving  $MPS_{n,m,d}$  for general values of  $m \ll n$ , and leave the task of determining the threshold  $\tau \in (0,1)$  at which  $MPS_{n,O(n^{\tau}),d}$  is polynomial-time solvable to future work.

<sup>&</sup>lt;sup>10</sup>Assuming we have a *worst-case* algorithm for solving (say) cubic equations, then it will also solve quadratic equations as a subclass. However, it is more tricky when considering *average-case* algorithms, since the distribution of random cubic equations is different from the distribution of random quadratic ones.

#### 4.2 Learning Parity with Noise

**Definition 4.2** (Decisional LPN). Let  $\mathbb{F}$  be a finite field, and  $n = n(\lambda)$ ,  $m = m(\lambda)$ ,  $\epsilon = \epsilon(\lambda) \in (0,1)$  be polynomial-time computable functions in  $\lambda$ . Given an efficiently sampleable distribution  $\mathcal{M} = \mathcal{M}(n,m,\mathbb{F})$  over matrices in  $\mathbb{F}^{n\times m}$ , we say that the LPN<sub> $n,m,\mathcal{M},\epsilon,\mathbb{F}$ </sub> assumption is  $(T(\lambda),\delta(\lambda))$ -hard if for every adversary  $\mathcal{A}$  running in time at most T, the following holds for every  $\lambda \in \mathbb{N}$ :

$$\mathbf{Adv}_{n,m,\mathcal{M},\epsilon,\mathbb{F}}^{\mathsf{LPN}}(\mathcal{A}) := \left| \Pr\left[ b = 1 \left| \begin{array}{c} \mathbf{A} \leftarrow \mathcal{M} \\ \mathbf{s} \leftarrow \mathbb{F}^{1 \times n} \\ \mathbf{e} \leftarrow \mathsf{Ber}(\mathbb{F},\epsilon)^{1 \times m} \\ b \leftarrow \mathcal{A}(\mathbf{A},\mathbf{sA}+\mathbf{e}) \end{array} \right| - \Pr\left[ b = 1 \left| \begin{array}{c} \mathbf{A} \leftarrow \mathcal{M} \\ \mathbf{u} \leftarrow \mathbb{F}^{1 \times m} \\ b \leftarrow \mathcal{A}(\mathbf{A},\mathbf{u}) \end{array} \right| \le \delta. \right] \right|$$

We say that  $LPN_{n,m,\mathcal{M},\epsilon,\mathbb{F}}$  is (polynomially) hard if it is  $(poly(\lambda), negl(\lambda))$ -hard. When  $\mathbb{F} = \mathbb{F}_2$  and  $\mathcal{M}$  is the uniform distribution over matrices in  $\mathbb{F}_2^{n \times m}$ , we say that this is the  $LPN_{n,m,\epsilon}$  assumption [BFKL94].

Security of LPN. We summarize the best known attacks against LPN<sub>*n*,*m*, $\epsilon$ </sub>. While there have been multiple attack strategies against LPN over the years (see [BCGI18, BCG<sup>+</sup>20] for a survey of attacks), these attacks can all be captured by the *linear test framework* (see e.g. [CRR21] for a formal statement, though this observation goes back to at least [MST03]). As a consequence, their time complexity in solving LPN with error  $\epsilon = o(1)$  is at least  $2^{\tilde{O}(\epsilon n)}$ . In particular, this means that LPN<sub>*n*,*m*, $\epsilon$  is plausibly polynomially secure for m = poly(n) and  $\epsilon = 1/n^{0.5+\delta}$  for any  $0 < \delta < 1/2$ . This is the parameter regime required for the PKE construction in [BKM20] that suffices for building NIZK in Theorem 5.3.</sub>

We now define the Dense-Sparse LPN assumption with a different distribution of **A**, as introduced in [DJ24]. Before introducing the assumption, however, we first discuss a technicality regarding sampling sparse matrix **M** that has no constant-weight vectors in its kernel with *overwhelming* probability.

**Definition 4.3** (Good Distributions of Sparse Matrices). For every  $n \in \mathbb{N}$ , k = k(n),  $m = m(n) < n^{k/2}$  and  $d = \omega(1)$ , define

$$\mathsf{SpMat}(n, m, k, d) = \left\{ \mathbf{M} \in \mathsf{SpMat}(n, m, k) \middle| \begin{array}{l} \forall \mathbf{x} \in \mathbb{F}_2^m \text{ such that } \mathbf{Mx} = \mathbf{0}, \\ we \text{ have } \mathbf{x} = \mathbf{0} \text{ or } \|\mathbf{x}\|_0 \ge d \end{array} \right\}$$

to be the subset of SpMat(n, m, k) consisting of matrices with dual distance of at least d. We say that an efficiently sampleable distribution  $\mathcal{M}_{sp}$  over SpMat(n, m, k) is  $(d, \delta)$ -good if  $\mathcal{M}_{sp}$  has min-entropy at least  $n^c$  for some constant c, and furthermore,

$$\Pr[\mathbf{M} \notin \mathsf{SpMat}(n, m, k, d) \mid \mathbf{M} \leftarrow \mathcal{M}_{\mathsf{sp}}] \leq \delta.$$

We say that  $\mathcal{M}_{sp}$  is good if it is  $(d, \delta)$ -good for some  $d = \omega(1)$  and  $\delta = \operatorname{negl}(n)$ .

Random sampling of *k*-sparse matrices  $\mathbf{A} \leftarrow \mathsf{SpMat}(n, m, k)$  do *not* give a good distribution (for any constant *k*), because the probability of getting a good matrix is only  $1 - 1/\mathsf{poly}(n)$ . However, there exists other instantiations of a good distribution of sparse matrices, which can be used for our constructions.

**Theorem 4.1** (Theorem 7.18 [AK19], adapted). For every even  $k \ge 6$ , every 1 < c < k/4 with  $\gamma = k - 4c$ , there exists an efficiently computable,  $(O(n^{\delta}), \operatorname{negl}(n))$ -good distribution over SpMat(n, m, k), where  $m = n^c$  and  $\delta = \frac{k - 4c - \gamma}{k - \gamma - 4}$ . We call this the AK19 distribution.

**Definition 4.4** (Dense-Sparse LPN). Let  $k \in \mathbb{N}$ ,  $k \geq 3$  be a constant, and let  $n = n(\lambda)$ ,  $m = m(\lambda) < n^{k/2}$ , and  $\epsilon = \epsilon(\lambda) \in (0,1)$  be polynomial-time computable functions of  $\lambda$ . Let  $\mathcal{M}_{sp}$  be a good distribution over SpMat(n, m, k). We define the Dense-Sparse LPN assumption DS-LPN<sub> $n,m,k,\mathcal{M}_{sp},\epsilon$ </sub> to be the LPN<sub> $n,m,\mathcal{M},\epsilon,\mathbb{F}_2$ </sub> assumption for the following distribution of matrices  $\mathcal{M}$ :

$$\mathcal{M} = \{ \mathbf{T} \cdot \mathbf{M} \mid \mathbf{T} \leftarrow \mathbb{F}_2^{n/2 \times n}, \mathbf{M} \leftarrow \mathcal{M}_{\mathsf{sp}} \}.$$

In other words, the assumption states that the following two distributions are computationally indistinguishable:

$$\{(\mathbf{A}, \mathbf{sA} + \mathbf{e})\}_{n \in \mathbb{N}} \approx_c \{(\mathbf{A}, \mathbf{u})\}_{n \in \mathbb{N}},\$$

where  $\mathbf{T} \leftarrow \mathbb{F}_2^{n/2 \times n}$ ,  $\mathbf{M} \leftarrow \mathcal{M}_{sp}$ ,  $\mathbf{A} = \mathbf{T} \cdot \mathbf{M}$ ,  $\mathbf{s} \leftarrow \mathbb{F}_2^{1 \times n/2}$ ,  $\mathbf{e} \leftarrow \mathsf{Ber}(\epsilon)^{1 \times m}$ , and  $\mathbf{u} \leftarrow \mathbb{F}_2^m$ .

In particular, we will use Dense-Sparse LPN in its compression regime as defined in [D]24].

**Definition 4.5** (Compression Regime of Dense-Sparse LPN). Let  $k \in \mathbb{N}$ ,  $k \ge 3$  and D > 1 be constants. Define  $\delta_{(4.5)}(k, D) := 1 - \frac{k/2-1}{Dk-1}$ . For any constant  $\delta \in (\delta_{(4.5)}(k, D), 1)$ , we define the  $(D, \delta)$ -compression regime of Dense-Sparse LPN to be the regime where  $m \ge n^{1+(Dk-1)(1-\delta)}$ .

This compression regime satisfies the following property. Let  $t = n^{\delta}$ . Consider the hash function

$$f_{\mathbf{A}}: \mathsf{Ball}_{\mathsf{reg}}(m,t) o \mathbb{F}_2^{n/2} \quad ext{defined by} \quad f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A} \cdot \mathbf{x}$$

where  $\mathbf{A} = \mathbf{TM}$  is drawn from the DS-LPN<sub>*n*,*m*,*k*, $\mathcal{M}_{sp},\epsilon$  distribution. If the number of samples *m* is in the  $(D, \delta)$ -compression regime, then  $f_{\mathbf{A}}$  achieves output size compression with ratio *D*, e.g.,</sub>

$$|\mathsf{Ball}_{\mathsf{reg}}(m,t)| = 2^{t \log(m/t)} > (|\mathsf{Ball}(n,kt)|)^{D} \ge (|f(\mathsf{Ball}_{\mathsf{reg}}(m,t))|)^{D}$$

Security of Dense-Sparse LPN. We summarize the security of Dense-Sparse LPN according to [D]24]. For DS-LPN<sub>*n*,*m*,*k*, $\mathcal{M}_{sp},\epsilon$  with  $n < m < n^{k/2}$ , and  $\mathcal{M}_{sp}$  being any good distribution, the best attacks run in subexponential time that is min  $\left(2^{\widetilde{O}(\epsilon n)}, 2^{\widetilde{O}(\widetilde{n})}\right)$ , where  $\widetilde{n} = n \left(\frac{n}{m}\right)^{\frac{1}{k/2-1}}$ . Roughly speaking, the first asymptotic comes from generic attacks against LPN (see "Security of LPN" above), and the second asymptotic comes from finding cycles of size  $\widetilde{n}$  in  $\mathbf{A} = \mathbf{TM}$ , which exist with high probability and can be used to break Dense-Sparse LPN.</sub>

In particular, Dense-Sparse LPN is plausibly polynomially secure for any  $m = n^{k/2-\rho}$  and  $\epsilon = 1/n^{\tau}$ , for any  $\rho > 0$  and  $\tau \in (0, 1)$ . This in particular is the parameter regime required for our lossy PKE construction in Section 6.

### 5 NIZK and CI Hash Constructions

In this section, we present our constructions of correlation-intractable hashing (Section 5.1) and NIZK arguments (Section 5.2).

**Parameters.** Let  $d \ge 2$  be the degree,  $\epsilon \in [0, 1/2)$  be the approximation error, n be the input length, and  $m = n^{1-\rho}$  be the output length for any  $\rho \in (0, 1/2)$ .

**Construction.** We define  $\mathcal{H} = (\mathsf{Gen}, \mathsf{Hash})$  as follows.

- $\operatorname{Gen}(1^{\lambda}) \to \operatorname{hk.} \operatorname{Return} \operatorname{hk} := \mathscr{P} \leftarrow \operatorname{Poly}(n, m, d, \mathbb{F}_2).$
- Hash(hk,  $\mathbf{x}$ )  $\rightarrow$  h. On input  $\mathbf{x} \in \mathbb{F}_2^n$ , return  $\mathbf{h} := \mathscr{P}(\mathbf{x}) \in \mathbb{F}_2^m$ .

Figure 2: Correlation-intractable hashing from MPS for functions  $\epsilon$ -approximable by constantdegree polynomials

### 5.1 Correlation Intractability from Approximate MPS

First, we construct correlation-intractable hashing for functions approximable by constant-degree polynomials (or by a NIZK-friendly suitable subclass). Our first construction works for approximate degree-*d* polynomials assuming the exponential hardness of solving degree-*d* polynomial equations, or more generally, the polynomial hardness of approximately solving degree-*d* equations.

**Theorem 5.1.** Let  $d \ge 2, 0 \le \epsilon < 1/2, 0 < \rho < 1/2$  be constants such that for every polynomial  $n = n(\lambda)$  and  $m(\lambda) = n(\lambda)^{1-\rho}$ , either

- $MPS_{n,m,d,\mathbb{F}_2}$  is  $(poly(m), 2^{-H(\epsilon)m} \cdot negl(m))$ -hard, or
- Apx-MPS<sub> $n,m,d,\epsilon,\mathbb{F}_2$ </sub> is (poly(m), negl(m))-hard.

Then Figure 2 gives a construction of correlation-intractible hashing for all functions  $\frac{\epsilon}{2}$ -approximable by degree-d polynomials. In particular, when  $\epsilon = 0$ , Figure 2 gives a hash function that is CI for degree-d polynomials assuming MPS<sub>*n*,*n*<sup>1- $\rho$ </sup> is (poly(*m*), negl(*m*))-hard.</sub>

*Proof.* From Lemma 3.6, it suffices to show that the hash function  $\mathcal{H}$  in Figure 2 satisfies  $\text{CI-Apx}_{\epsilon}$  for the function class  $\text{Poly}(n, m, d, \mathbb{F}_2)$  of degree-*d* polynomials. Since the former assumption (exponential hardness of MPS) implies the latter assumption (polynomial hardness of Apx-MPS) by Lemma 4.2, we only need to show  $\text{CI-Apx}_{\epsilon}$  based on the polynomial hardness of Apx-MPS<sub>*n*,*m*,*d*, $\mathbb{F}_2$ .</sub>

Recall that  $\mathcal{H}$  satisfies CI-Apx<sub> $\epsilon$ </sub> for the function class Poly $(n, m, d, \mathbb{F}_2)$  if it satisfies CI for every relation of the form

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

for every  $\mathscr{C} \in \mathsf{Poly}(n, m, d, \mathbb{F}_2)$ . Fix such a polynomial tuple  $\mathscr{C}$ . We will show that for every  $\mathcal{A}$  in the CI game for relation  $R^{\epsilon}_{\mathscr{C}}$ , there exists an adversary  $\mathcal{B}$  (nearly as efficient as  $\mathcal{A}$ ) in the Apx-MPS<sub> $n,m,d,\epsilon,\mathbb{F}_2$ </sub> game such that

$$\mathbf{Adv}^{\mathsf{CI}}_{\mathcal{H},R^{\epsilon}_{\mathscr{C}}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{Apx-MPS}}_{n,m,d,\epsilon,\mathbb{F}_{2}}(\mathcal{B}).$$

This claim would finish the proof. The adversary  $\mathcal{B}$  works as follows:

- 1.  $\mathcal{B}$  receives input  $\mathscr{P}$  from the Apx-MPS game, and runs  $\mathcal{A}$  with hash key hk :=  $\mathscr{P} + \mathscr{C}$ .
- 2.  $\mathcal{B}$  receives output x from  $\mathcal{A}$  and returns x.

Note that the hash key hk is uniformly distributed by the uniform distribution of  $\mathscr{P}$ , which is guaranteed by the Apx-MPS game. It suffices to show that  $\mathcal{B}$  wins the Apx-MPS game whenever  $\mathcal{A}$  wins the CI game. Indeed, if the latter happens, then  $(\mathbf{x}, \mathcal{H}.\mathsf{Hash}(\mathsf{hk}, \mathbf{x})) \in R^{\epsilon}_{\mathscr{C}}$ , which is equivalent to  $\Delta((\mathscr{P} + \mathscr{C})(\mathbf{x}), \mathscr{C}(\mathbf{x})) \leq \epsilon m$ , and this is equivalent to  $\Delta(\mathscr{P}(\mathbf{x}), \mathbf{0}) \leq \epsilon m$ , meaning that  $\mathcal{B}$  wins.

Our second construction is based on the exponential hardness of MQ, which is more wellstudied in the literature than the general constant-degree version MPS, with the tradeoff that to guarantee compression, we can only support a subclass of constant-degree polynomials that we define below. In Section 5.2, we show that this subclass is expressive enough to obtain NIZK, along with a public-key encryption scheme having approximate linear decryption.

**Definition 5.1** (Concatenated Constant-Degree Polynomials). For any constant degree  $d \ge 1$ , individual input length n, individual output length m, and number of concatenation  $\ell$ , we define the class CPoly $(n, m, d, \ell, \mathbb{F}_2)$  of  $\ell$ -concatenated degree-d polynomials to be

$$\mathsf{CPoly}(n, m, d, \ell, \mathbb{F}_2) := \{ (\mathscr{P}_1 \| \dots \| \mathscr{P}_\ell) \mid \mathscr{P}_i \in \mathsf{Poly}(n, m, d, \mathbb{F}_2) \forall i \in [\ell] \},\$$

with the notation that  $(\mathscr{P}_1 \| \dots \| \mathscr{P}_\ell)(\mathbf{x}) = \mathscr{P}_1(\mathbf{x}_1) \| \dots \| \mathscr{P}_\ell(\mathbf{x}_\ell) \in \mathbb{F}_2^{m\ell}$  for every  $\mathbf{x} = \mathbf{x}_1 \| \dots \| \mathbf{x}_\ell \in \mathbb{F}_2^{n\ell}$ with  $\mathbf{x}_i \in \mathbb{F}_2^n$  for all *i*. Here  $\|$  denotes string concatenation.

The key property we use for  $\ell$ -concatenated degree-d polynomials is that they can be *linearized* with an input size blowup that is independent of  $\ell$ , simply by linearizing each concatenated term. In other words, given  $\mathscr{P} = (\mathscr{P}_1 \| \dots \| \mathscr{P}_\ell) \in \mathsf{CPoly}(n, m, d, \ell, \mathbb{F}_2)$ , there exists a *linear* form  $\mathscr{L} : \mathbb{F}_2^{\binom{n+d}{d}\ell} \to \mathbb{F}_2^{m\ell}$  such that for every  $\mathbf{x}_1, \dots, \mathbf{x}_\ell \in \mathbb{F}_2^n$ , we have

$$\mathscr{L}\left(\mathsf{Sym}_{< d}(\mathbf{x}_1) \| \dots \| \mathsf{Sym}_{< d}(\mathbf{x}_\ell)\right) = \mathscr{P}\left(\mathbf{x}_1 \| \dots \| \mathbf{x}_\ell\right).$$

This linear form is obtained by concatenating the linearization of each polynomial. Namely, if  $\mathscr{P} = \mathscr{P}_1 \| \dots \| \mathscr{P}_{\ell}$ , and each  $\mathscr{P}_i = (p_{i,j,\vec{\alpha}})_{j \in [m], |\vec{\alpha}| \leq d'}$  then we simply define  $\mathscr{L} = \mathscr{L}_1 \| \dots \| \mathscr{L}_{\ell}$  where  $\mathscr{L}_i = ((p_{i,j,\vec{\alpha}})_{0 < |\vec{\alpha}| \leq d}, p_{i,j,0})_{i \in [m]}$  is a linear form.

**Theorem 5.2.** Let  $d \ge 2$ ,  $0 \le \epsilon < 1/2$ ,  $0 < \rho < 1/2$  be constants such that either of the following happens for every polynomial  $n(\lambda)$  and every  $n(\lambda)^{1-\rho} < m(\lambda) < n(\lambda)$ :

- $MQ_{n,m}$  is  $(poly(m), 2^{-H(\epsilon)m} \cdot negl(m))$ -hard, or
- Apx-MQ<sub> $n,m,\epsilon$ </sub> is (poly(m), negl(m))-hard.

Then for every  $\tilde{m}$  and  $\tilde{n} > \tilde{m}$  that are polynomials in  $\lambda$ , Figure 3 gives a construction of a correlationintractible hash  $\mathcal{H}$  for all functions  $\frac{\epsilon}{2}$ -approximable by the class  $\mathsf{CPoly}(\tilde{n}, \tilde{m}, d, \ell, \mathbb{F}_2)$  for every large enough  $\ell$  (as determined in Figure 3). Furthermore,  $\mathcal{H}$  satisfies programmability.

*Proof.* First, note that programmability holds since the constant part of  $hk = \mathcal{Q}$  is chosen uniformly at random. Similar to the proof of Theorem 5.1, we may assume the polynomial hardness of Apx-MQ<sub>*n*,*m*, $\epsilon$ </sub> (by Lemma 4.2), and we only need to show that  $\mathcal{H}$  satisfies CI-Apx<sub> $\epsilon$ </sub> for the class CPoly( $\tilde{n}, \tilde{m}, d, \ell, \mathbb{F}_2$ ) (by Lemma 3.6).

Fix some  $\mathscr{C} \in \mathsf{CPoly}(\tilde{n}, \tilde{m}, d, \ell, \mathbb{F}_2)$ . We will show that  $\mathcal{H}$  satisfies CI for the relation  $R^{\epsilon}_{\mathscr{C}}$ . Let  $\mathcal{A}$  be an adversary in the corresponding CI game. We will construct an adversary  $\mathcal{B}$  (nearly as efficient as  $\mathcal{A}$ ) in the Apx-MQ<sub>*n*,*m*, $\epsilon$  game, where  $n = {\tilde{n}+d \choose d}\ell$  and  $m = \tilde{m}\ell$ , such that</sub>

$$\mathbf{Adv}^{\mathsf{Cl}}_{\mathcal{H},R^{\epsilon}_{\mathscr{C}}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{Apx-MQ}}_{n,m,\epsilon}(\mathcal{B}).$$

The adversary  $\mathcal{B}$  works as follows:

### Parameters.

- Let  $d \ge 2$  be the degree,  $\epsilon \in [0, 1/2)$  be the approximation error, and  $\rho \in (0, 1/2)$  be such that Apx-MQ<sub>*n*,*m*, $\epsilon$ </sub> is polynomially-hard for every polynomial  $n(\lambda)$  and every  $n^{1-\rho} < m < n$ .
- Let  $\widetilde{m}$  and  $\widetilde{n} > \widetilde{m}$  be polynomials in  $\lambda$ . Choose  $\ell > {\binom{\widetilde{n}+d}{d}}^{\frac{1-\rho}{\rho}} \widetilde{m}^{-\frac{1}{\rho}}$  large enough so that  ${\binom{\widetilde{n}+d}{d}}\ell^{1-\rho} < \widetilde{m}\ell$ .

**Construction.** We define  $\mathcal{H} = (Gen, Hash)$  as follows.

- $\operatorname{Gen}(1^{\lambda}) \to \operatorname{hk.} \operatorname{Return} \operatorname{hk} := \mathscr{Q} \leftarrow \operatorname{Poly}\left(\binom{\widetilde{n}+d}{d}\ell, \widetilde{m}\,\ell, 2, \mathbb{F}_2\right).$
- Hash(hk, x) → h. On input x ∈ F<sub>2</sub><sup>nℓ</sup>, decompose it as x = x<sub>1</sub> || ... ||x<sub>ℓ</sub> where x<sub>i</sub> ∈ F<sub>2</sub><sup>n</sup> for all i ∈ [ℓ].
  Return h := 𝔅 (Sym<sub><d</sub>(x<sub>1</sub>) || ... ||Sym<sub><d</sub>(x<sub>ℓ</sub>)) ∈ F<sub>2</sub><sup>mℓ</sup>.

Figure 3: Correlation-intractable hashing from MQ for functions  $\epsilon$ -approximable by concatenated constant-degree polynomials

- 1.  $\mathcal{B}$  receives input  $\mathscr{Q} \leftarrow \mathsf{Poly}(n, m, 2, \mathbb{F}_2)$  from the Apx-MQ game.
- 2.  $\mathcal{B}$  constructs the linearized polynomial  $\mathscr{L} : \mathbb{F}_2^{(\tilde{n}_d^+)\ell} \to \mathbb{F}_2^{\tilde{m}\ell}$  corresponding to  $\mathscr{C}$  (following the discussion after Definition 5.1).
- 3.  $\mathcal{B}$  runs  $\mathcal{A}$  on the hash key hk :=  $\mathcal{Q} + \mathcal{L}$ .
- 4.  $\mathcal{B}$  receives output  $\mathbf{x} \in \mathbb{F}_2^{\widetilde{n}\ell}$  from  $\mathcal{A}$ , parses  $\mathbf{x} = \mathbf{x}_1 \| \dots \| \mathbf{x}_\ell$  where  $\mathbf{x}_i \in \mathbb{F}_2^{\widetilde{n}}$  for all  $i \in [\ell]$ , and returns  $\widetilde{\mathbf{x}} = \mathsf{Sym}_{\leq d}(\mathbf{x}_1) \| \dots \| \mathsf{Sym}_{\leq d}(\mathbf{x}_\ell)$ .

Note that the hash key hk is randomly distributed since  $\mathscr{Q}$  is. It remains to show that  $\mathscr{B}$  wins whenever  $\mathscr{A}$  wins. Indeed, if  $\mathscr{A}$  wins then we have that  $\Delta(\mathcal{H}.\mathsf{Hash}(\mathsf{hk},\mathbf{x}),\mathscr{C}(\mathbf{x})) \leq \epsilon m$ , which is equivalent to

$$\Delta\left(\left(\mathscr{Q}+\mathscr{L}\right)\left(\mathsf{Sym}_{< d}(\mathbf{x}_{1})\right\|\ldots\|\mathsf{Sym}_{< d}(\mathbf{x}_{\ell})\right), \mathscr{C}(\mathbf{x})\right) \leq \epsilon m.$$

Since  $\mathscr{L}$  is the linearized polynomial corresponding to  $\mathscr{C}$ , we have that  $\mathscr{L}(\mathsf{Sym}_{\leq d}(\mathbf{x}_1) \| \dots \| \mathsf{Sym}_{\leq d}(\mathbf{x}_\ell)) = \mathscr{C}(\mathbf{x})$ . Hence the above is equivalent to

$$\Delta\left(\mathscr{Q}\left(\mathsf{Sym}_{< d}(\mathbf{x}_{1}) \| \dots \| \mathsf{Sym}_{< d}(\mathbf{x}_{\ell})\right), \mathbf{0}\right) \leq \epsilon m,$$

which implies that  $\mathcal{B}$  wins in the Apx-MQ game.

### 5.2 NIZK Constructions

We now show that our CI hash functions in Section 5.1 suffice for achieving NIZK, together with a public-key encryption scheme satisfying approximate linear decryption. In particular, we explain how the framework of Brakerski-Koppula-Mour [BKM20] for building NIZK can accommodate the CI hash constructed in Theorem 5.2.

**Theorem 5.3.** *There exists NIZK for* NP *with adaptive computational soundness and adaptive computational zero-knowledge, assuming the following:* 

- (a) For every polynomial  $p(\lambda)$  and some large enough constant  $d \in \mathbb{N}$ , there exists a public-key encryption scheme PKE where the decryption function  $\text{Dec}(td, \cdot)$  has a  $\frac{1}{p(\lambda)}$ -probabilistic representation by *d*-degree polynomials;
- (b) For some large enough constant d', small enough constant  $\epsilon > 0$ , and polynomial  $n = n(\lambda)$ , there exists some large enough polynomial  $\ell(\lambda)$  and a programmable CI hash function  $\mathcal{H}$  for relations  $\epsilon$ -probabilistically searchable by the class CPoly $(n, 1, d', \ell, \mathbb{F}_2)$  of  $\ell$ -concatenated degree-d' polynomials.

*Furthermore, we achieve NIZK with non-adaptive computational soundness and adaptive statistical zeroknowledge for NP under the conditions above and* 

(c) The public-key encryption scheme PKE in part (a) is lossy.

Since our statement is different from the corresponding result from [BKM20, Corollary 3.16], we will sketch how their framework gives rise to Theorem 5.3. To do so, we recall the NIZK construction at a high level in Figure 4. From Figure 4, it is clear that the bad challenge function BadChal has the following properties:

- 1. It is the *concatenation* of the bad challenge function  $\operatorname{BadChal}_{td}^{(i)}$  for each execution  $i \in [\ell]$  of the parallel-repeated protocol. That is, we have  $\operatorname{BadChal}_{td}(\mathbf{a}) = \operatorname{BadChal}_{td}^{(1)}(a_1) \| \dots \| \operatorname{BadChal}_{td}^{(\ell)}(a_\ell).$
- 2. Therefore, if each BadChal<sub>td</sub><sup>(*i*)</sup> has a  $\epsilon$ -probabilistic representation by constant-degree polynomials, then BadChal<sub>td</sub> has a  $\epsilon$ -probabilistic representation by  $\ell$ -concatenated constant-degree polynomials. Note that this is the case if the encryption scheme PKE satisfies condition (a) of Theorem 5.3.

Therefore, as long as we set the number of repetitions  $\ell$  to be large enough so that the CI hash from Theorem 5.2 achieves compression, then the hash function satisfies condition (b) of Theorem 5.3. Finally, the fact that condition (c) implies statistical zero-knowledge follows from [CCH<sup>+</sup>19, Theorem 5.5].

Putting everything together, we derive our NIZK constructions. First, we get NIZK from LPN and MQ using [BKM20, Theorem 6.1] and our Theorem 5.2.

**Corollary 5.1.** There exists NIZK for NP in the common random string model, with adaptive computational soundness and adaptive computational zero-knowledge, under the following assumptions:

- 1. For some  $\delta \in (0, 1/2)$  and polynomial p, LPN<sub> $n,m,\epsilon$ </sub> with m = p(n) and  $\epsilon = n^{-0.5-\delta}$  is  $(poly(\lambda), negl(\lambda))$ -hard, and
- 2. For some  $\rho \in (0, 1/2)$  and  $\tau \in (0, 1)$ ,  $MQ_{n,m}$  with  $m = n^{1-\rho}$  is  $(poly(\lambda), 2^{-\tau m})$ -hard.

*Proof.* It suffices to show how the conditions of the corollary can enable constructions of NIZK according to prior results. We choose parameters as follows:

- Choose  $\xi$  such that  $H(2\xi) = \tau/2$ . This will guarantee that the MQ-based hash will be CI for functions  $\xi$ -approximable by concatenated constant-degree polynomials (see 5.2).
- Choose *d* such that  $2^{-d} < \xi/2$ . This will be the degree of the constant-degree approximation of the 3CNF verification predicate.

### Ingredients.

- Let PKE be a (lossy) public-key encryption scheme.
- Let  $\Pi$  be a commit-then-open  $\Sigma$ -protocol for an NP-complete language L, using PKE for encrypting the first round's message, satisfying instance-universality, delayed-input, 3CNF verification, and unique bad challenge, with BadChal<sub>td</sub> be the bad challenge function (See Section 3.6). Denote the transcript for one execution of  $\Pi$  by (a, c, z), where  $c \in \{0, 1\}$ .
- Let  $\mathcal{H}$  be a CI hash for the class of bad challenge functions  $\mathsf{BadChal}_{\mathsf{td}}$ .

### Construction.

- Setup $(1^{\lambda}) \rightarrow (crs, td)$ . Sample hk  $\leftarrow \mathcal{H}.Gen(1^{\lambda})$ . For statistical zero-knowledge, sample pk  $\leftarrow \mathsf{PKE}.\mathsf{LossyGen}(1^{\lambda})$ . Else sample (pk, sk)  $\leftarrow \mathsf{PKE}.Gen(1^{\lambda})$ . Return crs = (pk, hk) and td = sk.
- *P*(crs, *x*, *w*) → *π* = (**a**, **z**). Parse (*x*, *w*) as an instance-witness pair for Π. Do a parallel repetition of Π for *l* = poly(λ) times, giving prover's messages (**a**, **z**), and where the challenges are derived via **c** ← *H*.Hash(hk, **a**).
- $\mathcal{V}(crs, x, \pi) \rightarrow b$ . Parse  $\pi = (\mathbf{a}, \mathbf{z})$ , and check  $\pi$  according to  $\Pi$ 's verifier, with challenges derived via  $\mathbf{c} \leftarrow \mathcal{H}$ .Hash(hk, **a**).
- BadChal<sub>td</sub>(a)  $\rightarrow$  c. For each  $i \in [\ell]$ , compute  $c_i \leftarrow \text{BadChal}_{td}(a_i)$ . Concatenate the bad challenges to form  $\mathbf{c} \in \mathbb{F}_2^{\ell}$ .

Figure 4: Template for NIZK from correlation intractability [CCH<sup>+</sup>19, BKM20]. The purple parts denote modifications for achieving statistical zero-knowledge.

- Let *n* be the length of the proof string π to be encrypted by *P*. Choose n<sub>1</sub> large enough so that the linear-approximate PKE scheme in [BKM20, Construction 6.1] has decryption error 1/n<sub>1</sub><sup>δ</sup> < ξ/(2n). This guarantees that when decrypting all encrypted bits of π, the correctness error is less than ξ/2.</li>
- Combining the above two settings of *d* and *n*<sub>1</sub>, together with [BKM20, Theorem 3.15], gives a *ξ*-approximate degree-*d'* representation of the BadChal<sub>td</sub> function, for some constant *d'*.
- Finally, choose the repetition parameter ℓ large enough so that the MQ-based hash will hash inputs of length ℓ <sup>(n1+d')</sup><sub>d'</sub> = ℓ<sup>1−ρ</sup> to ℓ bits.

From this choice of parameters, we can see that our PKE scheme and CI hash satisfies the conditions of 5.3, which finishes the corollary.  $\Box$ 

Second, we get NIZK with statistical zero-knowledge from Dense-Sparse LPN and MQ, using Theorem 6.1, Theorem 5.2, and a similar parameter instantiation as in 5.1.

**Corollary 5.2.** There exists NIZK for NP in the common reference string model, with non-adaptive computational soundness and adaptive statistical zero-knowledge, under the following assumptions:

- 1. For parameters  $n, m, k, \mathcal{M}_{sp}, \epsilon$  defined in Figure 5, DS-LPN<sub> $n,m,k,\mathcal{M}_{sp},\epsilon$ </sub> is  $(\text{poly}(\lambda), \text{negl}(\lambda))$ -hard, and
- 2. For some  $\rho \in (0, 1/2)$  and  $\tau \in (0, 1)$ ,  $\mathsf{MQ}_{n,m}$  with  $m = n^{1-\rho}$  is  $(\mathsf{poly}(\lambda), 2^{-\tau m})$ -hard.

**Remark 5.1.** We briefly explain why our NIZK in Corollary 5.1 is in the common *random* string. The hash key for our CI hash from MQ (Theorem 5.2) is uniformly random. For the PKE scheme from LPN [BKM20], the public key is of the form (**A**, **B**), where **B** = **SA** + **E** is *not* close to being randomly distributed. However, we can circumvent this issue and have a random public key by just sampling **B** uniformly at random. In the security proof, we switch to **B** = **SA** + **E** in the first hybrid, which is indistinguishable due to the LPN<sub>*n*,*m*, $\epsilon$  assumption.</sub>

# 6 Lossy PKE from Dense-Sparse LPN

In this section, we describe a lossy public-key encryption scheme from the Dense-Sparse LPN assumption (Definition 4.4), whose decryption function is  $1/\operatorname{poly}(\lambda)$ -approximable by a linear function. Our parameters for this lossy PKE scheme is in a similar regime to that enabling lossy trapdoor functions in [D]24].

**Theorem 6.1.** Let  $p(\lambda)$  be an arbitrarily large polynomial. Assume that the DS-LPN<sub>*n,m,k,M*<sub>sp</sub>,  $\epsilon$  assumption is (polynomially) secure with parameters as defined in Figure 5). Then Figure 5 gives a construction of a lossy PKE scheme such that the decryption function has a  $1/p(\lambda)$ -probabilistic representation by linear forms.</sub>

*Proof.* We establish the desired properties of the construction one-by-one.

**Mode Indistinguishability.** Recall that this means  $pk = (\mathbf{A}, \mathbf{SA} + \mathbf{E})$  is indistinguishable from  $\widetilde{pk} = (\mathbf{A}, \mathbf{B})$ , for  $\mathbf{A} = \mathbf{TM}$  from the Dense-Sparse distribution,  $\mathbf{S} \leftarrow \mathbb{F}_2^{\ell \times n/2}$ ,  $\mathbf{E} \leftarrow \mathsf{Ber}(\epsilon)^{\ell \times m}$ , and  $\mathbf{B} \leftarrow \mathbb{F}_2^{\ell \times m}$ . This follows directly from DS-LPN<sub>*n*,*m*,*k*, $\mathcal{M}_{sp},\epsilon$  and a hybrid argument over the rows of **B**.</sub>

**Statistical Hiding.** Since the message  $\mu$  is only present in the second part of the ciphertext  $\mathbf{B}\widetilde{\mathbf{x}} + (\underbrace{\mu, \dots, \mu}_{\ell})$ , it suffices to show that  $\mathbf{B}\widetilde{\mathbf{x}}$  is statistically close to random given the public key and the

first part of the ciphertext. In other words, we need to show that:

$$(\mathbf{A}, \mathbf{B}, \mathbf{A}\widetilde{\mathbf{x}}, \mathbf{B}\widetilde{\mathbf{x}}) \approx_s (\mathbf{A}, \mathbf{B}, \mathbf{A}\widetilde{\mathbf{x}}, \mathbf{u}),$$

where  $\tilde{\mathbf{x}} = \mathsf{spfy}(\mathbf{x})$  with  $\mathbf{x} \leftarrow \mathbb{F}_2^{t \log(m/t)}$ , and  $\mathbf{u} \leftarrow \mathbb{F}_2^{\ell}$  is also uniformly sampled. In fact, we show that the above holds for every fixed  $\mathbf{A}$ , namely

$$(\mathbf{B}, \mathbf{B}\widetilde{\mathbf{x}}, \mathbf{A}\widetilde{\mathbf{x}}) \approx_s (\mathbf{B}, \mathbf{u}, \mathbf{A}\widetilde{\mathbf{x}}).$$

This is now the setting for us to apply the Leftover Hash Lemma (Lemma 3.4), with  $X = \tilde{\mathbf{x}} = spfy(\mathbf{x})$  as the random variable,  $Z = \mathbf{A}\tilde{\mathbf{x}}$  as the leakage,  $h = \mathbf{B}$  as the description of a universal hash function, and  $h(X) = \mathbf{B}\tilde{\mathbf{x}} \in \mathbb{F}_2^{\ell}$  as the hash value. Since our parameters for Dense-Sparse LPN are in the  $(D, \delta)$ -compression regime, it means that

$$2^{t\log(m/t)} > |\mathcal{Z}|^D,$$

### Parameters.

- Let  $p(\lambda)$  be a polynomial in  $\lambda$  such that we want the decryption function to have a  $1/p(\lambda)$ -probabilistic representation by linear polynomials.
- Let  $k \ge 3$  be an integer, D > 1, and  $\delta \in (\delta_{(4.5)}(k, D), 1)$ . Let  $\rho \in (0, 1 \delta)$  be an arbitrarily small constant.
- Let  $n > (2p(\lambda))^{1/\rho}$ ,  $m = n^{1+(Dk-1)(1-\delta)}$ , and consider a good distribution  $\mathcal{M}_{sp} \subseteq SpMat(n, m, k)$ .
- Let  $\epsilon = \frac{1}{n^{\delta+\rho}}$ ,  $\tau = \frac{2}{n^{\rho}}$  and  $t = \ell = n^{\delta}$ .

Construction. We define PKE as follows.

- Gen $(1^{\lambda}) \rightarrow (\mathsf{pk},\mathsf{sk})$ . Sample  $\mathbf{A} = \mathbf{TM} \in \mathbb{F}_2^{n/2 \times m}$ , where  $\mathbf{T} \leftarrow \mathbb{F}_2^{n/2 \times n}$  and  $\mathbf{M} \leftarrow \mathcal{M}_{\mathsf{sp}}$ , according to the Dense-Sparse LPN distribution. Sample  $\mathbf{S} \leftarrow \mathbb{F}_2^{\ell \times n/2}$ ,  $\mathbf{E} \leftarrow \mathsf{Ber}(\epsilon)^{\ell \times m}$ , and compute  $\mathbf{B} = \mathbf{SA} + \mathbf{E}$ . Return  $\mathsf{pk} := (\mathbf{A}, \mathbf{B})$  and  $\mathsf{sk} := \mathbf{S}$ .
- LossyGen $(1^{\lambda}) \to \widetilde{\mathsf{pk}}$ . Sample  $\mathbf{A} = \mathbf{TM} \in \mathbb{F}_2^{n/2 \times m}$  as above. Sample  $\mathbf{B} \leftarrow \mathbb{F}_2^{\ell \times m}$  and return  $\widetilde{\mathsf{pk}} := (\mathbf{A}, \mathbf{B})$ .
- $\operatorname{Enc}(\operatorname{pk}, \mu \in \mathbb{F}_2) \to \operatorname{ct.} \operatorname{Sample} \mathbf{x} \leftarrow \mathbb{F}_2^{t \log(m/t)}$ , compute  $\widetilde{\mathbf{x}} = \operatorname{spfy}(\mathbf{x}) \in \operatorname{Ball}_{\operatorname{reg}}(m, t)$ , and return  $\operatorname{ct} := \left(\mathbf{A}\widetilde{\mathbf{x}}, \mathbf{B}\widetilde{\mathbf{x}} + (\underbrace{\mu, \dots, \mu}_{\ell})\right)$ .
- Dec(sk, ct) → μ' ∈ {0, 1, ⊥}. Parse ct = (c ∈ 𝔽<sup>n/2</sup><sub>2</sub>, d ∈ 𝔽<sup>ℓ</sup><sub>2</sub>), then return Majority<sub>τ</sub>(Sc + d). Here the gap majority function is defined as follows:

$$\begin{split} \mathsf{M}\mathsf{a}\mathsf{j}\mathsf{o}\mathsf{r}\mathsf{i}\mathsf{t}\mathsf{y}_{\tau}(\mathbf{v}\in\mathbb{F}_{2}^{k}) := \begin{cases} 0 & \text{ if } & \|\mathbf{v}\|_{0} \leq \tau\ell, \\ 1 & \text{ if } & \|\mathbf{v}\|_{0} \geq (1-\tau)\ell, \\ \bot & \text{ otherwise.} \end{cases} \end{split}$$

### Figure 5: Lossy PKE from Dense-Sparse LPN

where  $\mathcal{Z} = \{\mathbf{A} \cdot \mathsf{spfy}(\mathbf{x}) \mid \mathbf{x} \in \mathbb{F}_2^{t \log(m/t)}\}$ . Hence, using Lemma 3.3 we have that

$$2^{\mathbf{H}_{\infty}(X|Z)} \geq \frac{2^{\mathbf{H}_{\infty}(X)}}{|\mathcal{Z}|} > 2^{t \log(m/t)(1-1/D)}.$$

In other words,  $A\tilde{x}$  loses at least (1/D)-th of the information on  $\tilde{x}$ . Now we conclude by Lemma 3.4 that

$$\mathbf{SD}\left((\mathbf{B},\mathbf{B}\widetilde{\mathbf{x}},\mathbf{A}\widetilde{\mathbf{x}}),(\mathbf{B},\mathbf{u},\mathbf{A}\widetilde{\mathbf{x}})\right) \leq \frac{1}{2}\sqrt{\frac{2^{\ell}}{2^{t\log(m/t)(1-1/D)}}} = 2^{-\Omega(t\log(m/t))} = \mathsf{negl}(\lambda),$$

by the choice of  $\ell = t$ .

**Correctness.** Consider any public key  $(\mathbf{A}, \mathbf{B} = \mathbf{S}\mathbf{A} + \mathbf{E})$  as in the construction. Note that if  $(\mathbf{c}, \mathbf{d}) = (\mathbf{A}\widetilde{\mathbf{x}}, \mathbf{B}\widetilde{\mathbf{x}} + (\mu, \dots, \mu))$  is a ciphertext of  $\mu$  in our scheme, then we have

$$\mathbf{Sc} + \mathbf{d} = \mathbf{SA}\widetilde{\mathbf{x}} + (\mathbf{SA} + \mathbf{E})\widetilde{\mathbf{x}} + (\underbrace{\mu, \dots, \mu}_{\ell}) = \mathbf{E}\widetilde{\mathbf{x}} + (\underbrace{\mu, \dots, \mu}_{\ell}).$$

Therefore, we have that decryption is correct if and only if  $\|\mathbf{E}\widetilde{\mathbf{x}}\|_0 \leq \tau \ell$ . Since  $\widetilde{\mathbf{x}} \in \mathsf{Ball}_{\mathsf{reg}}(m, t)$  is always *t*-sparse, it follows that  $\mathbf{e} := \mathbf{E}\widetilde{\mathbf{x}}$  is Bernoulli distributed with probability  $\epsilon' = \frac{1-(1-2\epsilon)^{\ell}}{2} < \epsilon \ell = \frac{1}{n^{\rho}} = \frac{\tau}{2}$ , by Lemma 3.1 and the choice of  $\epsilon$ ,  $\ell$ , and  $\tau$ . Now by Chernoff bound (Lemma 3.2), it follows that

$$\Pr[\|\mathbf{e}\|_0 > \tau\ell] \le \exp(-\Omega(\tau\ell)) = \exp(-\Omega(n^{\delta-\rho})) = \mathsf{negl}(\lambda),$$

which is what we want. Note that  $\delta > \rho$  since  $\delta > \delta_{(4.5)}(k, D) > 1/2$  and  $\rho \le 1 - \delta$  by our choice.

Linear Approximate Decryption. Consider the following randomized family of linear functions

$$\mathscr{L}_{\mathsf{sk}=\mathbf{S}}\left(\mathsf{ct}=(\mathbf{c},\mathbf{d})\in\mathbb{F}_{2}^{n/2+\ell}
ight)$$
 :

Sample  $i \leftarrow [\ell]$ , and output  $\mu' = (\mathbf{S})_i \mathbf{c} + \mathbf{d}_i$ , where  $(\mathbf{S})_i$  is the *i*-th row of  $\mathbf{S}$ .

It follows that if  $\text{Dec}(\text{sk}, \text{ct}) = \mu \neq \perp$ , then  $\mathbf{cS} + \mathbf{d}$  agrees with  $(\mu, \dots, \mu)$  for at least  $(1 - \tau)$ -fraction of the entries. Thus a randomly chosen *i*-th entry  $(\mathbf{Sc} + \mathbf{d})_i = (\mathbf{S})_i \mathbf{c} + \mathbf{d}_i$  will agree with  $\mu$  with probability at least  $1 - \tau$ . This finishes the proof.

# 7 References

- [AFF<sup>+</sup>14] Martin R. Albrecht, Jean-Charles Faugère, Robert Fitzpatrick, Ludovic Perret, Yosuke Todo, and Keita Xagawa. Practical cryptanalysis of a public-key encryption scheme based on new multivariate quadratic assumptions. In Hugo Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 446–464. Springer, Berlin, Heidelberg, March 2014. 5
- [AHI<sup>+</sup>17] Benny Applebaum, Naama Haramaty, Yuval Ishai, Eyal Kushilevitz, and Vinod Vaikuntanathan. Low-complexity cryptographic hash functions. In Christos H. Papadimitriou, editor, *ITCS 2017*, volume 4266, pages 7:1–7:31, 67, January 2017. LIPIcs. 5
- [AK19] Benny Applebaum and Eliran Kachlon. Sampling graphs without forbidden subgraphs and unbalanced expanders with negligible error. In David Zuckerman, editor, *60th FOCS*, pages 171–179. IEEE Computer Society Press, November 2019. 19
- [Ale03] Michael Alekhnovich. More on average case vs approximation complexity. In 44th FOCS, pages 298–307. IEEE Computer Society Press, October 2003. 2, 6
- [AY20] Shweta Agrawal and Shota Yamada. Optimal broadcast encryption from pairings and LWE. In Anne Canteaut and Yuval Ishai, editors, EUROCRYPT 2020, Part I, volume 12105 of LNCS, pages 13–43. Springer, Cham, May 2020. 6
- [BBK+23] Zvika Brakerski, Maya Farber Brodsky, Yael Tauman Kalai, Alex Lombardi, and Omer Paneth. SNARGs for monotone policy batch NP. In Helena Handschuh and Anna Lysyanskaya, editors, CRYPTO 2023, Part II, volume 14082 of LNCS, pages 252–283. Springer, Cham, August 2023. 1
- [BCG<sup>+</sup>14] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In 2014 IEEE Symposium on Security and Privacy, pages 459–474. IEEE Computer Society Press, May 2014. 1
- [BCG<sup>+</sup>19] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Efficient pseudorandom correlation generators: Silent OT extension and more. In Alexandra Boldyreva and Daniele Micciancio, editors, CRYPTO 2019, Part III, volume 11694 of LNCS, pages 489–518. Springer, Cham, August 2019. 5
- [BCG<sup>+</sup>20] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Correlated pseudorandom functions from variable-density LPN. In 61st FOCS, pages 1069–1080. IEEE Computer Society Press, November 2020. 2, 19
- [BCGI18] Elette Boyle, Geoffroy Couteau, Niv Gilboa, and Yuval Ishai. Compressing vector OLE. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, ACM CCS 2018, pages 896–912. ACM Press, October 2018. 2, 19
- [BCM<sup>+</sup>18] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh V. Vazirani, and Thomas Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. In Mikkel Thorup, editor, 59th FOCS, pages 320–331. IEEE Computer Society Press, October 2018. 1, 2
- [Beu21] Ward Beullens. Improved cryptanalysis of UOV and Rainbow. In Anne Canteaut and Franccois-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 348–373. Springer, Cham, October 2021. 5
- [Beu22a] Ward Beullens. Breaking rainbow takes a weekend on a laptop. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 464–479. Springer, Cham, August 2022. 5
- [Beu22b] Ward Beullens. MAYO: Practical post-quantum signatures from oil-and-vinegar maps. In Riham AlTawy and Andreas Hülsing, editors, *SAC 2021*, volume 13203 of *LNCS*, pages 355–376. Springer, Cham, September / October 2022. 5

- [BFKL94] Avrim Blum, Merrick L. Furst, Michael J. Kearns, and Richard J. Lipton. Cryptographic primitives based on hard learning problems. In Douglas R. Stinson, editor, CRYPTO'93, volume 773 of LNCS, pages 278–291. Springer, Berlin, Heidelberg, August 1994. 2, 19
- [BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *20th ACM STOC*, pages 103–112. ACM Press, May 1988. 1, 4
- [BFP11] Luk Bettale, Jean-Charles Faugère, and Ludovic Perret. Cryptanalysis of multivariate and oddcharacteristic HFE variants. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, PKC 2011, volume 6571 of LNCS, pages 441–458. Springer, Berlin, Heidelberg, March 2011. 5
- [BGI<sup>+</sup>01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 1–18. Springer, Berlin, Heidelberg, August 2001. 2
- [BGP06] Côme Berbain, Henri Gilbert, and Jacques Patarin. QUAD: A practical stream cipher with provable security. In Serge Vaudenay, editor, EUROCRYPT 2006, volume 4004 of LNCS, pages 109–128. Springer, Berlin, Heidelberg, May / June 2006. 5
- [BHY09] Mihir Bellare, Dennis Hofheinz, and Scott Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In Antoine Joux, editor, *EURO-CRYPT 2009*, volume 5479 of *LNCS*, pages 1–35. Springer, Berlin, Heidelberg, April 2009. 4
- [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*, volume 3876 of *LNCS*, pages 60–79. Springer, Berlin, Heidelberg, March 2006. 1
- [BKM20] Zvika Brakerski, Venkata Koppula, and Tamer Mour. NIZK from LPN and trapdoor hash via correlation intractability for approximable relations. In Daniele Micciancio and Thomas Ristenpart, editors, CRYPTO 2020, Part III, volume 12172 of LNCS, pages 738–767. Springer, Cham, August 2020. 1, 4, 6, 8, 9, 11, 12, 14, 15, 16, 19, 23, 24, 25, 26
- [BLSV18] Zvika Brakerski, Alex Lombardi, Gil Segev, and Vinod Vaikuntanathan. Anonymous IBE, leakage resilience and circular security from new assumptions. In Jesper Buus Nielsen and Vincent Rijmen, editors, EUROCRYPT 2018, Part I, volume 10820 of LNCS, pages 535–564. Springer, Cham, April / May 2018. 9
- [Blu86] Manuel Blum. How to prove a theorem so no one else can claim it. In *Proceedings of the International Congress of Mathematicians*, volume 1, page 2. Citeseer, 1986. 8, 14
- [BLVW19] Zvika Brakerski, Vadim Lyubashevsky, Vinod Vaikuntanathan, and Daniel Wichs. Worst-case hardness for LPN and cryptographic hashing via code smoothing. In Yuval Ishai and Vincent Rijmen, editors, EUROCRYPT 2019, Part III, volume 11478 of LNCS, pages 619–635. Springer, Cham, May 2019. 4, 9
- [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, EUROCRYPT 2003, volume 2656 of LNCS, pages 614–629. Springer, Berlin, Heidelberg, May 2003. 1
- [BP17] Ward Beullens and Bart Preneel. Field lifting for smaller UOV public keys. In Arpita Patra and Nigel P. Smart, editors, *INDOCRYPT 2017*, volume 10698 of *LNCS*, pages 227–246. Springer, Cham, December 2017. 5
- [Bra18] Zvika Brakerski. Quantum FHE (almost) as secure as classical. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 67–95. Springer, Cham, August 2018. 1, 2

- [BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In Rafail Ostrovsky, editor, *52nd FOCS*, pages 97–106. IEEE Computer Society Press, October 2011. 1, 2
- [BY93] Mihir Bellare and Moti Yung. Certifying cryptographic tools: The case of trapdoor permutations. In Ernest F. Brickell, editor, CRYPTO'92, volume 740 of LNCS, pages 442–460. Springer, Berlin, Heidelberg, August 1993. 1
- [BY96] Mihir Bellare and Moti Yung. Certifying permutations: Noninteractive zero-knowledge based on any trapdoor permutation. *Journal of Cryptology*, 9(3):149–166, June 1996. 4
- [CCH<sup>+</sup>18] Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, and Ron D. Rothblum. Fiat-Shamir from simpler assumptions. Cryptology ePrint Archive, Report 2018/1004, 2018. 3
- [CCH<sup>+</sup>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 Moses Charikar and Edith Cohen, editors, 51st ACM STOC, pages 1082–1090. ACM Press, June 2019. 1, 4, 6, 8, 9, 12, 14, 24, 25
- [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, EUROCRYPT 2018, Part I, volume 10820 of LNCS, pages 91–122. Springer, Cham, April / May 2018. 1, 3
- [CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *Journal of the ACM (JACM)*, 51(4):557–594, 2004. 12
- [CGJ<sup>+</sup>23] Arka Rai Choudhuri, Sanjam Garg, Abhishek Jain, Zhengzhong Jin, and Jiaheng Zhang. Correlation intractability and SNARGs from sub-exponential DDH. In Helena Handschuh and Anna Lysyanskaya, editors, CRYPTO 2023, Part IV, volume 14084 of LNCS, pages 635–668. Springer, Cham, August 2023. 1, 2
- [CGMT02] Nicolas Courtois, Louis Goubin, Willi Meier, and Jean-Daniel Tacier. Solving underdefined systems of multivariate quadratic equations. In David Naccache and Pascal Paillier, editors, *PKC 2002*, volume 2274 of *LNCS*, pages 211–227. Springer, Berlin, Heidelberg, February 2002. 3, 18
- [CHK03] Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In Eli Biham, editor, EUROCRYPT 2003, volume 2656 of LNCS, pages 255–271. Springer, Berlin, Heidelberg, May 2003. 1
- [CHK07] Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. *Journal of Cryptology*, 20(3):265–294, July 2007. 4
- [CHMT14] Chen-Mou Cheng, Yasufumi Hashimoto, Hiroyuki Miura, and Tsuyoshi Takagi. A polynomialtime algorithm for solving a class of underdetermined multivariate quadratic equations over fields of odd characteristics. In Michele Mosca, editor, *Post-Quantum Cryptography - 6th International Workshop*, PQCrypto 2014, pages 40–58. Springer, Cham, October 2014. 3, 18
- [CJJ21] Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. Non-interactive batch arguments for NP from standard assumptions. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021*, *Part IV*, volume 12828 of *LNCS*, pages 394–423, Virtual Event, August 2021. Springer, Cham. 1, 2
- [CJJ22] Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. SNARGs for  $\mathcal{P}$  from LWE. In *62nd FOCS*, pages 68–79. IEEE Computer Society Press, February 2022. 1, 2
- [CJJQ23] Geoffroy Couteau, Abhishek Jain, Zhengzhong Jin, and Willy Quach. A note on noninteractive zero-knowledge from CDH. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part IV*, volume 14084 of *LNCS*, pages 731–764. Springer, Cham, August 2023. 1

- [CKPS00] Nicolas Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In Bart Preneel, editor, EUROCRYPT 2000, volume 1807 of LNCS, pages 392–407. Springer, Berlin, Heidelberg, May 2000. 3
- [CKU20] Geoffroy Couteau, Shuichi Katsumata, and Bogdan Ursu. Non-interactive zero-knowledge in pairing-free groups from weaker assumptions. In Anne Canteaut and Yuval Ishai, editors, EUROCRYPT 2020, Part III, volume 12107 of LNCS, pages 442–471. Springer, Cham, May 2020.
- [CL18] Ran Canetti and Amit Lichtenberg. Certifying trapdoor permutations, revisited. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part I*, volume 11239 of *LNCS*, pages 476– 506. Springer, Cham, November 2018. 1
- [CRR21] Geoffroy Couteau, Peter Rindal, and Srinivasan Raghuraman. Silver: Silent VOLE and oblivious transfer from hardness of decoding structured LDPC codes. In Tal Malkin and Chris Peikert, editors, CRYPTO 2021, Part III, volume 12827 of LNCS, pages 502–534, Virtual Event, August 2021. Springer, Cham. 19
- [CW23] Jeffrey Champion and David J. Wu. Non-interactive zero-knowledge from non-interactive batch arguments. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part II*, volume 14082 of *LNCS*, pages 38–71. Springer, Cham, August 2023. 1
- [DDN91] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography (extended abstract). In 23rd ACM STOC, pages 542–552. ACM Press, May 1991. 1
- [DDS<sup>+</sup>20] Jintai Ding, Joshua Deaton, Kurt Schmidt, Vishakha, and Zheng Zhang. Cryptanalysis of the lifted unbalanced oil vinegar signature scheme. In Daniele Micciancio and Thomas Ristenpart, editors, CRYPTO 2020, Part III, volume 12172 of LNCS, pages 279–298. Springer, Cham, August 2020. 5
- [DDY<sup>+</sup>08] Jintai Ding, Vivien Dubois, Bo-Yin Yang, Chia-Hsin Owen Chen, and Chen-Mou Cheng. Could SFLASH be repaired? In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *ICALP 2008, Part II*, volume 5126 of *LNCS*, pages 691–701. Springer, Berlin, Heidelberg, July 2008. 5
- [DFSS07] Vivien Dubois, Pierre-Alain Fouque, Adi Shamir, and Jacques Stern. Practical cryptanalysis of SFLASH. In Alfred Menezes, editor, CRYPTO 2007, volume 4622 of LNCS, pages 1–12. Springer, Berlin, Heidelberg, August 2007. 5
- [DGH<sup>+</sup>20] Nico Döttling, Sanjam Garg, Mohammad Hajiabadi, Daniel Masny, and Daniel Wichs. Two-round oblivious transfer from CDH or LPN. In Anne Canteaut and Yuval Ishai, editors, *EU-ROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 768–797. Springer, Cham, May 2020.
   6
- [DGKV22] Lalita Devadas, Rishab Goyal, Yael Kalai, and Vinod Vaikuntanathan. Rate-1 non-interactive arguments for batch-NP and applications. In *63rd FOCS*, pages 1057–1068. IEEE Computer Society Press, October / November 2022. 1
- [DJ24] Quang Dao and Aayush Jain. Lossy cryptography from code-based assumptions. *Cryptology ePrint Archive*, 2024. 3, 4, 9, 19, 20, 26
- [DMN12] Nico Döttling, Jörn Müller-Quade, and Anderson C. A. Nascimento. IND-CCA secure cryptography based on a variant of the LPN problem. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 485–503. Springer, Berlin, Heidelberg, December 2012. 6
- [DMP88] Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Non-interactive zero-knowledge proof systems. In Carl Pomerance, editor, *CRYPTO'87*, volume 293 of *LNCS*, pages 52–72. Springer, Berlin, Heidelberg, August 1988. 1

- [DRS04] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Christian Cachin and Jan Camenisch, editors, EUROCRYPT 2004, volume 3027 of LNCS, pages 523–540. Springer, Berlin, Heidelberg, May 2004. 11
- [DS05] Jintai Ding and Dieter Schmidt. Rainbow, a new multivariable polynomial signature scheme. In *International conference on applied cryptography and network security*, pages 164–175. Springer, 2005. 5
- [DY08] Jintai Ding and Bo-Yin Yang. Multivariates polynomials for hashing. In *Information Security and Cryptology: Third SKLOIS Conference, Inscrypt 2007, Xining, China, August 31-September 5, 2007, Revised Selected Papers 3,* pages 358–371. Springer, 2008. 5
- [FJ03] Jean-Charles Faugère and Antoine Joux. Algebraic cryptanalysis of hidden field equation (HFE) cryptosystems using gröbner bases. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 44–60. Springer, Berlin, Heidelberg, August 2003. 5
- [FLS90] Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple non-interactive zero knowledge proofs based on a single random string (extended abstract). In *31st FOCS*, pages 308–317. IEEE Computer Society Press, October 1990. 1, 4, 8, 14
- [FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Berlin, Heidelberg, August 1987. 1, 9
- [GKW17] Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. In Chris Umans, editor, *58th FOCS*, pages 612–621. IEEE Computer Society Press, October 2017. 1, 2
- [GKW18] Rishab Goyal, Venkata Koppula, and Brent Waters. Collusion resistant traitor tracing from learning with errors. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *50th ACM STOC*, pages 660–670. ACM Press, June 2018. 1, 2
- [GLS22] Riddhi Ghosal, Paul Lou, and Amit Sahai. Efficient NIZKs from LWE via polynomial reconstruction and "MPC in the head". In Shweta Agrawal and Dongdai Lin, editors, ASI-ACRYPT 2022, Part II, volume 13792 of LNCS, pages 496–521. Springer, Cham, December 2022.
- [GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *17th ACM STOC*, pages 291–304. ACM Press, May 1985. 1
- [GOS06a] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive zaps and new techniques for NIZK. In Cynthia Dwork, editor, CRYPTO 2006, volume 4117 of LNCS, pages 97–111. Springer, Berlin, Heidelberg, August 2006. 1, 4
- [GOS06b] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for NP. In Serge Vaudenay, editor, EUROCRYPT 2006, volume 4004 of LNCS, pages 339–358. Springer, Berlin, Heidelberg, May / June 2006. 1, 2, 4
- [GR13] Oded Goldreich and Ron D. Rothblum. Enhancements of trapdoor permutations. *Journal of Cryptology*, 26(3):484–512, July 2013. 1
- [GS08] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, Berlin, Heidelberg, April 2008. 2
- [GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, CRYPTO 2013, Part I, volume 8042 of LNCS, pages 75–92. Springer, Berlin, Heidelberg, August 2013. 1, 2

- [GSWW22] Rachit Garg, Kristin Sheridan, Brent Waters, and David J. Wu. Fully succinct batch arguments for NP from indistinguishability obfuscation. In Eike Kiltz and Vinod Vaikuntanathan, editors, TCC 2022, Part I, volume 13747 of LNCS, pages 526–555. Springer, Cham, November 2022. 1
- [GVW13] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, 45th ACM STOC, pages 545–554. ACM Press, June 2013. 1, 2
- [GVW15] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from LWE. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 503–523. Springer, Berlin, Heidelberg, August 2015. 1, 2
- [HJKS22] James Hulett, Ruta Jawale, Dakshita Khurana, and Akshayaram Srinivasan. SNARGs for P from sub-exponential DDH and QR. In Orr Dunkelman and Stefan Dziembowski, editors, EUROCRYPT 2022, Part II, volume 13276 of LNCS, pages 520–549. Springer, Cham, May / June 2022. 1, 2
- [HL18] Justin Holmgren and Alex Lombardi. Cryptographic hashing from strong one-way functions (or: One-way product functions and their applications). In Mikkel Thorup, editor, 59th FOCS, pages 850–858. IEEE Computer Society Press, October 2018. 3
- [HLOV11] Brett Hemenway, Benoit Libert, Rafail Ostrovsky, and Damien Vergnaud. Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. In Dong Hoon Lee and Xiaoyun Wang, editors, ASIACRYPT 2011, volume 7073 of LNCS, pages 70–88, December 2011. 4
- [HLY12] Yun-Ju Huang, Feng-Hao Liu, and Bo-Yin Yang. Public-key cryptography from new multivariate quadratic assumptions. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 190–205. Springer, Berlin, Heidelberg, May 2012. 5
- [JJ21] Abhishek Jain and Zhengzhong Jin. Non-interactive zero knowledge from sub-exponential DDH. In Anne Canteaut and Franccois-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 3–32. Springer, Cham, October 2021. 1, 4
- [JKKZ21] Ruta Jawale, Yael Tauman Kalai, Dakshita Khurana, and Rachel Yun Zhang. SNARGs for bounded depth computations and PPAD hardness from sub-exponential LWE. In Samir Khuller and Virginia Vassilevska Williams, editors, 53rd ACM STOC, pages 708–721. ACM Press, June 2021. 1
- [JLS21] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In Samir Khuller and Virginia Vassilevska Williams, editors, 53rd ACM STOC, pages 60–73. ACM Press, June 2021. 2, 6
- [JLS22] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from LPN over  $\mathbb{F}_p$ , DLIN, and PRGs in  $NC^0$ . In Orr Dunkelman and Stefan Dziembowski, editors, *EURO-CRYPT 2022, Part I*, volume 13275 of *LNCS*, pages 670–699. Springer, Cham, May / June 2022. 2, 6
- [KLV23] Yael Tauman Kalai, Alex Lombardi, and Vinod Vaikuntanathan. SNARGs and PPAD hardness from the decisional Diffie-Hellman assumption. In Carmit Hazay and Martijn Stam, editors, EUROCRYPT 2023, Part II, volume 14005 of LNCS, pages 470–498. Springer, Cham, April 2023. 1
- [KLVW23] Yael Kalai, Alex Lombardi, Vinod Vaikuntanathan, and Daniel Wichs. Boosting batch arguments and RAM delegation. In Barna Saha and Rocco A. Servedio, editors, 55th ACM STOC, pages 1545–1552. ACM Press, June 2023. 1
- [KMP14] Eike Kiltz, Daniel Masny, and Krzysztof Pietrzak. Simple chosen-ciphertext security from lownoise LPN. In Hugo Krawczyk, editor, PKC 2014, volume 8383 of LNCS, pages 1–18. Springer, Berlin, Heidelberg, March 2014. 6

- [KPG99] Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced Oil and Vinegar signature schemes. In Jacques Stern, editor, EUROCRYPT'99, volume 1592 of LNCS, pages 206–222. Springer, Berlin, Heidelberg, May 1999. 3, 5, 18
- [KS98] Aviad Kipnis and Adi Shamir. Cryptanalysis of the Oil & Vinegar signature scheme. In Hugo Krawczyk, editor, CRYPTO'98, volume 1462 of LNCS, pages 257–266. Springer, Berlin, Heidelberg, August 1998. 3, 5
- [KS99] Aviad Kipnis and Adi Shamir. Cryptanalysis of the HFE public key cryptosystem by relinearization. In Michael J. Wiener, editor, CRYPTO'99, volume 1666 of LNCS, pages 19–30. Springer, Berlin, Heidelberg, August 1999. 3
- [LLL22] Hanjun Li, Huijia Lin, and Ji Luo. ABE for circuits with constant-size secret keys and adaptive security. In Eike Kiltz and Vinod Vaikuntanathan, editors, TCC 2022, Part I, volume 13747 of LNCS, pages 680–710. Springer, Cham, November 2022. 6
- [LLY08] Feng-Hao Liu, Chi-Jen Lu, and Bo-Yin Yang. Secure PRNGs from specialized polynomial maps over any. In Johannes Buchmann and Jintai Ding, editors, *Post-quantum cryptography, second international workshop*, *PQCRYPTO 2008*, pages 181–202. Springer, Berlin, Heidelberg, October 2008. 5
- [LMZ23] Jiahui Liu, Hart Montgomery, and Mark Zhandry. Another round of breaking and making quantum money: How to not build it from lattices, and more. In Carmit Hazay and Martijn Stam, editors, EUROCRYPT 2023, Part I, volume 14004 of LNCS, pages 611–638. Springer, Cham, April 2023. 2
- [Mah18a] Urmila Mahadev. Classical homomorphic encryption for quantum circuits. In Mikkel Thorup, editor, *59th FOCS*, pages 332–338. IEEE Computer Society Press, October 2018. 1, 2
- [Mah18b] Urmila Mahadev. Classical verification of quantum computations. In Mikkel Thorup, editor, *59th FOCS*, pages 259–267. IEEE Computer Society Press, October 2018. 1, 2
- [MHT13] Hiroyuki Miura, Yasufumi Hashimoto, and Tsuyoshi Takagi. Extended algorithm for solving underdefined multivariate quadratic equations. In Philippe Gaborit, editor, Post-Quantum Cryptography - 5th International Workshop, PQCrypto 2013, pages 118–135. Springer, Berlin, Heidelberg, June 2013. 3, 18
- [MI88] Tsutomu Matsumoto and Hideki Imai. Public quadratic polynominal-tuples for efficient signature-verification and message-encryption. In C. G. Günther, editor, *EUROCRYPT'88*, volume 330 of *LNCS*, pages 419–453. Springer, Berlin, Heidelberg, May 1988. 2, 5
- [MST03] Elchanan Mossel, Amir Shpilka, and Luca Trevisan. On e-biased generators in NC0. In 44th FOCS, pages 136–145. IEEE Computer Society Press, October 2003. 19
- [NWW23] Shafik Nassar, Brent Waters, and David J Wu. Monotone policy bargs from bargs and additively homomorphic encryption. *Cryptology ePrint Archive*, 2023. 1
- [NY90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In 22nd ACM STOC, pages 427–437. ACM Press, May 1990. 1
- [OSS84] H. Ong, Claus-Peter Schnorr, and Adi Shamir. Efficient signature schemes based on polynomial equations. In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 37–46. Springer, Berlin, Heidelberg, August 1984. 2
- [Pas13] Rafael Pass. Unprovable security of perfect NIZK and non-interactive non-malleable commitments. In Amit Sahai, editor, TCC 2013, volume 7785 of LNCS, pages 334–354. Springer, Berlin, Heidelberg, March 2013. 3
- [Pat95] Jacques Patarin. Cryptanalysis of the Matsumoto and Imai public key scheme of eurocrypt'88. In Don Coppersmith, editor, CRYPTO'95, volume 963 of LNCS, pages 248–261. Springer, Berlin, Heidelberg, August 1995. 5

- [Pat96] Jacques Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. In Ueli M. Maurer, editor, EUROCRYPT'96, volume 1070 of LNCS, pages 33–48. Springer, Berlin, Heidelberg, May 1996. 5
- [Pat97] Jacques Patarin. The oil and vinegar algorithm for signatures. In *Dagstuhl Workshop on Cryp*tography, 1997, 1997. 3, 5
- [PCG01] Jacques Patarin, Nicolas Courtois, and Louis Goubin. FLASH, a fast multivariate signature algorithm. In David Naccache, editor, *CT-RSA 2001*, volume 2020 of *LNCS*, pages 298–307. Springer, Berlin, Heidelberg, April 2001. 5
- [PGC98] Jacques Patarin, Louis Goubin, and Nicolas Courtois. C<sup>\*</sup><sub>-+</sub> and HM: Variations around two schemes of T. Matsumoto and H. Imai. In Kazuo Ohta and Dingyi Pei, editors, ASIACRYPT'98, volume 1514 of LNCS, pages 35–49. Springer, Berlin, Heidelberg, October 1998. 5
- [PP22] Omer Paneth and Rafael Pass. Incrementally verifiable computation via rate-1 batch arguments. In *63rd FOCS*, pages 1045–1056. IEEE Computer Society Press, October / November 2022. 1
- [PS19] Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for NP from (plain) learning with errors. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I,* volume 11692 of *LNCS*, pages 89–114. Springer, Cham, August 2019. 1, 4
- [PVW08] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In David Wagner, editor, CRYPTO 2008, volume 5157 of LNCS, pages 554–571. Springer, Berlin, Heidelberg, August 2008. 4, 9
- [PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 187–196. ACM Press, May 2008. 4
- [Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, 37th ACM STOC, pages 84–93. ACM Press, May 2005. 1
- [SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In David B. Shmoys, editor, *46th ACM STOC*, pages 475–484. ACM Press, May / June 2014. 1
- [TDTD13] Chengdong Tao, Adama Diene, Shaohua Tang, and Jintai Ding. Simple matrix scheme for encryption. In Post-Quantum Cryptography: 5th International Workshop, PQCrypto 2013, Limoges, France, June 4-7, 2013. Proceedings 5, pages 231–242. Springer, 2013. 5
- [TW12] Enrico Thomae and Christopher Wolf. Solving underdetermined systems of multivariate quadratic equations revisited. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 156–171. Springer, Berlin, Heidelberg, May 2012. 3, 18
- [WBP05] Christopher Wolf, An Braeken, and Bart Preneel. Efficient cryptanalysis of RSE(2)PKC and RSSE(2)PKC. In Carlo Blundo and Stelvio Cimato, editors, *SCN 04*, volume 3352 of *LNCS*, pages 294–309. Springer, Berlin, Heidelberg, September 2005. 5
- [WW22] Brent Waters and David J. Wu. Batch arguments for NP and more from standard bilinear group assumptions. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022*, *Part II*, volume 13508 of *LNCS*, pages 433–463. Springer, Cham, August 2022. 1
- [WZ17] Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under LWE. In Chris Umans, editor, *58th FOCS*, pages 600–611. IEEE Computer Society Press, October 2017. 1, 2
- [YZ16] Yu Yu and Jiang Zhang. Cryptography with auxiliary input and trapdoor from constant-noise LPN. In Matthew Robshaw and Jonathan Katz, editors, CRYPTO 2016, Part I, volume 9814 of LNCS, pages 214–243. Springer, Berlin, Heidelberg, August 2016. 6