

Zero-Knowledge PCPs of Quasilinear Size via Locally Simulatable Sheaf Codes

Hadas Zeilberger
Yale University

Jack O'Connor
Riverlane

Tom Gur*
University of Cambridge

Nicholas Spooner
Cornell University

Abstract

We show that for every polynomial $T, b: \mathbb{N} \rightarrow \mathbb{N}$, there exist an $O(1)$ -query probabilistically checkable proof (PCP) for $\text{NTIME}(T)$ of length $\tilde{O}(T(n) + b(n)^2)$, which is $b(n)$ -query perfect zero knowledge. This strictly improves on the polynomial-length zero-knowledge PCPs of Gur, O'Connor, and Spooner (STOC 2024; STOC 2025). Our construction builds on the PCPs of Ben-Sasson and Sudan (SICOMP 2008) and Dinur (JACM 2007). We prove the zero-knowledge property of our PCPs via the new machinery of locally simulatable sheaf codes.

*Supported by ERC Starting Grant 101163189 and UKRI Future Leaders Fellowship MR/X023583/1.

Contents

1	Introduction	4
	Concurrent work	4
	Organization of the Paper	4
	Acknowledgements	4
2	Preliminaries	5
2.1	Set Theory and Functions	5
2.2	Linear Algebra	5
2.3	Coding Theory	7
2.3.1	Maximum Distance Separable Codes	7
2.3.2	Polynomial Codes	8
2.3.3	Locally Testable Codes	8
2.4	Probabilistically-Checkable Proofs (PCP)	9
2.5	Zero-Knowledge PCPs	9
3	Sheaf Codes	10
3.1	Basic Definitions	10
3.2	The Dimensions Of Local Codes	11
3.3	Isomorphism of Sheaf Codes	13
3.4	Merging Sheaf Codes	14
4	Testability of Sheaf Codes	15
4.1	Definitions	15
4.2	Recursive Testing of Sheaf Codes	16
4.3	Claims Related To Testability Of Sheaf Codes	17
5	Locally Simulatable Sheaf Codes	18
5.1	Basic Definitions	18
5.2	Extendability Implies Local Simulatability	19
5.3	Proving κ -Extendability in Sheaf Codes	21
5.3.1	Extendability in Extension Sheaf Codes	21
5.3.2	Basic Facts About Extendability	21
5.3.3	Extendability Reductions	22
5.3.4	Extendability For Maximum Distance Separable Codes	23
6	Perfect Zero-Knowledge PCPPs From Sheaf Codes	24
6.1	Perfect Zero-Knowledge PCPPs From d -Testable Sheaf Codes with κ -extendable covers	24
6.2	Constructing d -testable and κ -extendable covers through merging	25
6.2.1	Correctness of Construction 6.5	26
6.2.2	Testability of Construction 6.5	27
6.2.3	Extendability of Construction 6.5	27
6.3	d -Testable Sheaf Codes with κ -Extendable Covers From Recursively Testable Codes	29
7	Short, Perfect Zero-Knowledge PCPP For Reed-Solomon Codes	30
7.1	A Useful Generalization Of The BS08 Bivariate Reed-Muller Code	30
7.2	κ -Extendable Covers In Bivariate Reed-Muller Sheaf Codes	30
7.3	Testing Reed-Solomon Codes With Bivariate Reed-Muller Sheaf Codes	34
7.4	Our Construction	34

8	Perfect Zero-Knowledge PCP For NTIME	35
8.1	Linear Algebraic Constraint Satisfaction Problems	36
8.2	From zkPCPP for Reed-Solomon Codes to zkPCP For NTIME	37
	References	38
A	Zero-Knowledge Gap Amplification	39
B	Deferred Proofs Of Facts From Section 6.2	40
C	Deferred Proofs Of Facts From Proof of Theorem 7.3.	40
D	Proof Of Lemma 7.16	42

1 Introduction

A *probabilistically-checkable proof* (PCP) is a proof that can be efficiently verified by querying only a few locations chosen at random. PCPs are a central object in complexity theory, with deep connections to cryptography, coding theory and hardness of approximation. In this work we continue the study of *zero-knowledge PCPs* [KPT97] (ZKPCPs), which combine efficient verification with a strong privacy property, namely: *any* view of the PCP of bounded size should not reveal any information beyond the truth of the claim.

In recent work, Gur, O’Connor and Spooner [GOS24; GOS25] showed the existence of zero knowledge PCPs with parameters matching the original PCP theorem [AS98; ALMSS98]. That is, they showed that for any query bound $b: \mathbb{N} \rightarrow \mathbb{N}$, there exist an $O(1)$ -query PCPs NP with proof length $\text{poly}(n, b)$, such that the distribution of any local view of size at most $b(n)$ can be perfectly and efficiently simulated.

Over the decades that followed the original proof of the PCP theorem, much work has been done to reduce the length of the PCP proof, while maintaining constant query complexity. The current state-of-the-art of (non-ZK) PCPs [BS08; Din07] achieves quasi-linear proof length, i.e., $n \cdot \text{polylog}(n)$.

In this work we show that there exist ZKPCPs with comparable proof length to the best known non-ZK PCPs. Our main result is the following.

Theorem 1.1 (informally stated; see Theorem 8.2). *For every polynomial $T, b: \mathbb{N} \rightarrow \mathbb{N}$, there exist $O(1)$ -query ZKPCPs for $\text{NTIME}(T)$ of length $\tilde{O}(T(n) + b(n)^2)$, which are $b(n)$ -query perfect zero knowledge.*

Our construction is based on the short PCP of [BS08]. Underlying that construction is a PCP of proximity (PCPP) for Reed–Solomon codes. Our key technical contribution is to modify this PCPP to achieve zero knowledge properties and analyse it via a new object we introduce called *locally simulatable sheaf codes*. In addition to matching the proof length of the shortest non-ZK PCP constructions, this is the first construction of a ZKPCP in which the proof length does not grow *multiplicatively* in the query bound; i.e., we obtain zero knowledge up to $b(n) \approx \sqrt{T(n)}$ essentially “for free”, in an asymptotic sense.

Concurrent work

Independently and concurrently to this work, Bafna and Vyas [BV26] obtained constructions of similar but incomparable objects, which they refer to as “private PCPs”. Unlike ZKPCPs, private PCPs do not require efficient simulation; on the other hand, the privacy guarantee for private PCPs should also hold in some sense in the “no” case, whereas ZKPCPs provide a privacy guarantee only in the “yes” case.

Organization of the Paper

The rest of the paper is organized as follows. In Section 3, we lay the ground work by defining concepts related to sheaf codes. Next, in Section 4, we describe the properties needed for a sheaf code to be testable, and construct a simple codeword test. After that, in Section 5, we show that sheaf codes satisfying certain extendability properties are locally simulatable and we present a toolbox for proving extendability. In Section 6, we combine the results from Sections 4 and 5 to build perfect zero-knowledge PCPPs from sheaf codes satisfying both testability and extendability. We also show how to build testable and extendable sheaf codes from “recursively testable code families”, which we define. Next, we show that Reed–Solomon codes are a recursively testable code family and use them to instantiate the generic PZK-PCPP from sheaf codes. We conclude the paper with Section 8, where we build a PCP using techniques from [BCFGRS17].

Acknowledgements

We are grateful to Jack O’Connor for invaluable technical and conceptual contributions in early stages of this work.

Statement on use of AI. The authors used generative AI solely to convert diagrams to TikZ, to convert PDF to TeX for some of the preliminary definitions, to search for literature to include in the introduction, and for very limited proofreading.

2 Preliminaries

Notation. $\mathbb{F}$ is a finite field. For $d \in \mathbb{N}$, denote by $\mathbb{F}^{\leq d}[X]$ the set of univariate polynomials with coefficients in $\mathbb{F}$ and degree at most d . For $d_X, d_Y \in \mathbb{N}$, denote by $\mathbb{F}^{\leq [d_X, d_Y]}[X, Y]$ the set of bivariate polynomials with coefficients in $\mathbb{F}$, X -degree at most d_X and Y -degree at most d_Y . For $p : \mathbb{F} \rightarrow \mathbb{F}$ and $D \subset \mathbb{F}$, denote by $p(D) = \{p(x) : x \in D\}$ and for $Q : \mathbb{F} \times \mathbb{F} \rightarrow \mathbb{F}$ and $D \subseteq \mathbb{F} \times \mathbb{F}$, denote by $Q(D) = \{Q(x, y) : (x, y) \in D\}$. For all $X, Y \subset \mathbb{F}$, $D \subset X \times Y$, denote by D_X the set $\{x \in \mathbb{F} : \exists y \in \mathbb{F} \mid (x, y) \in D\}$ and by D_Y the set $\{y \in \mathbb{F} : \exists x \in \mathbb{F} \mid (x, y) \in D\}$. For $n \in \mathbb{N}$, denote by $[n]$ the interval between 0 and $n - 1$ inclusive.

2.1 Set Theory and Functions

Claim 2.1. Let $n, m \in \mathbb{N}$ and let $V_1, \dots, V_n, W_1, \dots, W_m$ be sets such that for all $i \in [n]$, $j \in [m]$, $V_i \cap W_j = \emptyset$. Let $V = \bigcup_{i \in [n]} V_i$ and $W = \bigcup_{j \in [m]} W_j$. Then,

$$V \cap W = \emptyset.$$

Proof.

$$V \cap W = \left(\bigcup_{i \in [n]} V_i \right) \cap \left(\bigcup_{j \in [m]} W_j \right) = \bigcup_{i \in [n]} V_i \cap \left(\bigcup_{j \in [m]} W_j \right) = \bigcup_{i \in [n]} \bigcup_{j \in [m]} V_i \cap W_j = \bigcup_{i \in [n]} \bigcup_{j \in [m]} \emptyset.$$

□

Claim 2.2. Let D_1, D_2 be two domains, and let $g : D_1 \rightarrow D_2$ be a bijection. Let $X, Y \subseteq D_1$ such that $X \subseteq Y$. Then $g(X) \subseteq g(Y)$.

Proof. Let $v \in g(X)$. Then there exists $w \in X$ such that $v = g(w)$. Since $X \subseteq Y$, $w \in Y$. Therefore $v = g(w) \in g(Y)$. □

2.2 Linear Algebra

For set D , we view a vector as a function $v : D \rightarrow \mathbb{F}$. For sets D_1, D_2 , we view a matrix, M , as a function $M : D_1 \times D_2 \rightarrow \mathbb{F}$. For $n \in \mathbb{N}$, we denote by Id_n the $n \times n$ identity matrix.

Definition 2.3 (Function Restriction). Let D be a set, let U be a subset of D , and let $w : D \rightarrow \mathbb{F}$ be a function from D to $\mathbb{F}$. We denote by $w|_U$ the restriction of w to U , that is the function $w|_U : U \rightarrow \mathbb{F}$ such that $w|_U(x) = w(x)$ for all $x \in U$.

Definition 2.4 (Vector Space Restriction). Let D be a set and let $V \subseteq \mathbb{F}^D$ be a vector space. Let $D' \subseteq D$. Then the restriction of V to D' , denoted by $V|_{D'}$ is the following vector space.

$$V|_{D'} = \{v|_{D'} : v \in V\}.$$

Definition 2.5 (Support Of A Vector). Let D be a finite set and let $V \subseteq \mathbb{F}^D$ be a vector space. Then the support of a vector $v \in V$, denote by $\text{supp}(v)$, is the set $\{x \in D : v[x] \neq 0\}$.

Claim 2.6. Let V_1, V_2 be two vector spaces and let $V = V_1 + V_2$. Then

$$\dim(V) = \dim(V_1) + \dim(V_2) - \dim(V_1 \cap V_2).$$

Furthermore if $V_1 \cap V_2 = \{\vec{0}\}$, then $\dim(V) = \dim(V_1) + \dim(V_2)$.

Permutation of a vector space.

Definition 2.7 (Permutation Mapping). *Let D, E be two domains, and let $g : D \rightarrow E$ be a bijection between them. Then g induces a natural mapping of vector spaces: for any subspace V of $\mathbb{F}^D$,*

$$g(V) = \{w \in \mathbb{F}^E \mid \exists v \in V : w(g(x)) = v(x) \quad \forall x \in D\}$$

is a subspace of $\mathbb{F}^E$. For each $v \in V$, we denote by $g(v)$ the vector in $g(V)$ satisfying $g(v)(g(x)) = v(x)$ for all $x \in D$. Similarly for each $w \in g(V)$, we denote by $g^{-1}(w)$ the vector in V such that $g^{-1}(w)(x) = w(g(x))$ for all $x \in D$.

Proposition 2.8. *Let D, E be domains. Let $V \subseteq \mathbb{F}^D$ be a vector space and let $g : D \rightarrow E$ be a bijection. Then for all $v \in V$ and $S \subseteq D$,*

$$g(v)|_{g(S)} = g|_S(v|_S) \tag{1}$$

$$g(V)|_{g(S)} = g|_S(V|_S) \tag{2}$$

Going forward, we denote $g|_S(V|_S)$ by $g(V|_S)$ and $g|_S(v|_S)$ by $g(v|_S)$.

Proof. To prove (1), for all $x \in S$,

$$g(v)|_{g(S)}(g(x)) = g(v)(g(x)) = v(x) \text{ and } g|_S(v|_S)(g(x)) = v|_S(x) = v(x)$$

To prove (2), we have the following.

$$g(V)|_{g(S)} = \{g(v)|_{g(S)} : v \in V\} = \{g|_S(v|_S) : v \in V\} = \{g|_S(w) : w \in V|_S\} = g|_S(V|_S) \quad \square$$

Padded vector spaces. Let D be a finite set, let $D' \subseteq D$ and let $V \subseteq \mathbb{F}^{D'}$.

Definition 2.9 (Zero-Padding of a Vector Space). *A zero-padding of $v \in \mathbb{F}^{D'}$ with respect to D , denoted by $\bar{v}_D$, is the vector in $\mathbb{F}^D$ such that*

$$\bar{v}_D|_{D'} = v \text{ and } \bar{v}_D|_{D \setminus D'} = \vec{0}.$$

A zero padded vector space, denoted by $\bar{V}_D \subseteq \mathbb{F}^D$, is the following subspace.

$$\bar{V}_D = \{\bar{v}_D : v \in V\} = V \times \{\vec{0}_{D \setminus D'}\}.$$

Claim 2.10. *The function $f : V \rightarrow \bar{V}_D$ defined by $f(v) = \bar{v}_D$ is a bijection and its inverse is the restriction map $g : \bar{V}_D \rightarrow V$ defined by $g(w) = w|_{D'}$.*

Proof. By definition of f and g , for all $v \in V$, $g(f(v)) = \bar{v}_D|_{D'} = v$ and for all $w \in \bar{V}_D$, $f(g(w)) = f(w|_{D'}) = (\overline{w|_{D'}})_D = w$. Therefore f and g are inverses of each other and so f is a bijection. $\square$

Corollary 2.11.

$$\dim(\bar{V}_D) = \dim(V).$$

Proof. By Claim 2.10, $|\bar{V}_D| = |V|$, and therefore $\dim(\bar{V}_D) = \log_{|\mathbb{F}|}(|\bar{V}_D|) = \log_{|\mathbb{F}|}(|V|) = \dim(V)$. $\square$

2.3 Coding Theory

Let Σ an alphabet and let $\ell \in \mathbb{N}$. A code C is a subset $C \subseteq \Sigma^\ell$. Given two strings $x, y \in \Sigma^n$, we denote the Hamming distance between x and y by $\Delta(x, y) := |\{i \in [n] : x_i \neq y_i\}|/n$. The minimum distance of a code, which we denote by Δ_C , $\min_{x, y \in C, x \neq y} (\Delta(x, y))$. When clear from context, we will sometimes abuse notation and refer instead to the absolute distance and absolute minimum distance as Δ and Δ_C respectively. If $\Sigma = \mathbb{F}$ and C is a linear subspace of $\mathbb{F}^\ell$ then C is a linear code.

Definition 2.12 (Code family). *Let $\{n_i\}_{i \geq 1}$ be an increasing sequence of block lengths, and suppose there exist sequences $\{k_i\}_{i \geq 1}$ and $\{d_i\}_{i \geq 1}$ such that, for every $i \geq 1$, there exists a code $C_i \subseteq \Sigma^{n_i}$ with distance d_i such that $|C_i| = k_i$. Then the sequence*

$$\mathcal{C} = \{C_i\}_{i \geq 1}$$

is a family of codes.

Definition 2.13 (Dual Code). *Let D be a finite set and let $C \subseteq \mathbb{F}^D$ be a linear code. Then the dual code, denoted by $C^\perp$ is equal to the following.*

$$C^\perp = \{v \in \mathbb{F}^D : \langle v, c \rangle = 0 \quad \forall c \in C\}.$$

Claim 2.14. *Let C_1, C_2 be two linear codes.*

$$(C_1 + C_2)^\perp = C_1^\perp \cap C_2^\perp.$$

Claim 2.15. *Let D be a finite set and let C be a linear code in $\mathbb{F}^D$. Then,*

$$\dim(C) = |D| - \dim(C^\perp).$$

Claim 2.16. *Let D be a finite set, let $D' \subseteq D$ and let $C \subseteq \mathbb{F}^{D'}$ be a linear code. Denote by $\widehat{C}_D$ the space $(\overline{C^\perp_D})^\perp$, i.e the dual code of the padded vector of the dual code of C . Then,*

$$\widehat{C}_D = \{c \in \mathbb{F}^D \mid c|_{D'} \in C\} = C \times \mathbb{F}^{D \setminus D'}$$

Proof. First we show that $\widehat{C}_D \subseteq \{c \in \mathbb{F}^D \mid c|_{D'} \in C\}$. If $v \in \widehat{C}_D$, then by Definition 2.13, $\langle v, c \rangle = 0$ for all $c \in \overline{C^\perp_D}$. Since $\text{supp}(\overline{C^\perp_D}) \subseteq D'$, this implies that $\langle v|_{D'}, c|_{D'} \rangle = 0$ for all $c \in \overline{C^\perp_D}$, which implies that $\langle v|_{D'}, c' \rangle = 0$ for all $c' \in \overline{C^\perp_D}|_{D'} = C^\perp$, where the equality follows by Claim 2.10. Therefore $v|_{D'} \in (C^\perp)^\perp$. Next, we show that $\{c \in \mathbb{F}^D \mid c|_{D'} \in C\} \subseteq \widehat{C}_D$. Suppose that $v \in \{c \in \mathbb{F}^D \mid c|_{D'} \in C\}$. Then for all $w \in \overline{C^\perp_D}$, $\langle v, w \rangle = \langle v|_{D'}, w|_{D'} \rangle = 0$, where the first equality follows from the fact that $\text{supp}(\overline{C^\perp_D}) \subseteq D'$, and the second equality follows because, by assumption, $v|_{D'} \in C$ and by definition of padded vector space (Definition 2.9), $w|_{D'} \in C^\perp$. Therefore, by definition of dual code (Definition 2.13), $v \in \widehat{C}_D$. $\square$

2.3.1 Maximum Distance Separable Codes

Definition 2.17 (Maximum Distance Separable Code). *Let D be a finite set and let $C \subseteq \mathbb{F}^D$ be a linear code. Then C is maximum distance separable (MDS) if*

$$\Delta_C = |D| - \dim(C) + 1.$$

Lemma 2.18 (Small Restrictions of MDS Codes Have Full Rank). *Let D be a finite set and let $C \subseteq \mathbb{F}^D$ be a linear, MDS, code (Definition 2.17). Then, for all $D' \subseteq D$ such that $|D'| \leq \dim(C)$, then*

$$\dim(C|_{D'}) = |D'|$$

Proof. First, suppose that $|D'| = \dim(C)$. Consider the map $f : C \rightarrow C|_{D'}$. By definition of vector space restriction (Definition 2.4), f is surjective. Next we prove that it's injective. Let $c_1, c_2 \in C$ such that $c_1 \neq c_2$. Suppose by contradiction that $f(c_1) = f(c_2)$. Then $f(c_1) - f(c_2) = 0$. By definition of vector space restriction (Definition 2.4), $f(c_1) - f(c_2) = c_1|_{D'} - c_2|_{D'} = (c_1 - c_2)|_{D'} = 0$. But since C is MDS, any non-zero codeword in C has at most $\dim(C) - 1$ zeroes. By assumption $c_1 - c_2 \neq 0$, but $|D'| = \dim(C)$, which implies that $c_1 - c_2$ has $\dim(C)$ zeroes, which is a contradiction. Therefore f is injective, which implies it is bijective and so $\dim(C|_{D'}) = \dim(C) = |D'|$. Next, let $E \subseteq D$ such that $|E| < \dim(C)$. Then there exists $D' \subseteq D$ such that $E \subseteq D'$ and $|D'| = \dim(C)$. Since $C|_{D'} \subseteq \mathbb{F}^{D'}$ and their dimensions are equal, it follows that $C|_{D'} = \mathbb{F}^{D'}$. By definition of restriction map, $C|_E = C|_{D'}|_E = \mathbb{F}^{D'}|_E = \mathbb{F}^E$, which completes the proof. $\square$

Lemma 2.19 (Support of Dual Vectors of MDS Code Is Larger Than Dimension). *Let D be a finite set and let $C \subseteq \mathbb{F}^D$ be an MDS code. Then for all non-zero $v \in C^\perp$, $|\text{supp}(v)| > \dim(C)$.*

Proof. Suppose that $|\text{supp}(v)| \leq \dim(C)$. Then there exists $D' \subseteq D$, $|D'| \leq \dim(C)$ such that $\text{supp}(v) = D'$. But this implies that for all $c \in C$, $\langle v, c \rangle = \langle v|_{D'}, c|_{D'} \rangle = 0$. In particular, this implies that $C|_{D'}^\perp$ contains $v|_{D'} \neq \vec{0}$ and is therefore nonzero. But this is a contradiction because by Lemma 2.18, since $|D'| \leq \dim(C)$, it follows that $\dim(C|_{D'}) = |D'|$ which implies that $\dim((C|_{D'})^\perp) = 0$. $\square$

2.3.2 Polynomial Codes

Definition 2.20 (Reed-Muller Code). *Given a field $\mathbb{F}$, a positive integer m , a degree vector $\vec{d} = (d_1, \dots, d_m) \in \mathbb{N}^m$, and $D \subset \mathbb{F}^m$, we denote by $RM[\mathbb{F}, D, \vec{d}]$ the following vector space.*

$$RM[\mathbb{F}, D, \vec{d}] = \{Q|_D : Q \in \mathbb{F}^{\leq \vec{d}}[X_1, \dots, X_m]\}.$$

In this section, we focus on bivariate Reed-Muller codes, i.e. the code $RM[\mathbb{F}, D, (d_X, d_Y)]$ for $d_X, d_Y \in \mathbb{N}$. The local codes of bivariate Reed-Muller codes are Reed-Solomon codes, which we define next.

Definition 2.21 (Reed-Solomon Code). *Given a field $\mathbb{F}$, a domain $D \subseteq \mathbb{F}$ a positive integer d , we denote by $RS[\mathbb{F}, D, d]$ the following vector space.*

$$RS[\mathbb{F}, D, d] = \{p|_D : p \in \mathbb{F}^{\leq d}[X]\}$$

i.e. the code of evaluations of univariate polynomials over D with degree less than or equal to d .

It is well-known that Reed-Solomon codes are maximum-distance separable.

Claim 2.22. *Let $D \subseteq \mathbb{F}$, let d be a positive integer smaller than $|D|$. Then $C = RS[\mathbb{F}, D, d]$ is maximum distance separable.*

Proof. By the Schwartz-Zippel Lemma, a univariate polynomial of degree at most d has at most d zeroes. Therefore, it's minimum Hamming weight is $|D| - d$, and since C is a linear code, this implies $\Delta_C = |D| - d$. Furthermore, $\dim(C) = d + 1$. Therefore $\Delta_C \geq |D| - d = |D| - (\dim(C) - 1) = |D| - \dim(C) + 1$, which completes the proof. $\square$

2.3.3 Locally Testable Codes

Definition 2.23 (Codeword tests). *Let $C : \{0, 1\}^k \rightarrow \{0, 1\}^n$ be a code (of distance d), and let $q \in \mathbb{N}$ and $\delta \in (0, 1)$. A q -local δ -tester for C is a probabilistic oracle machine M that makes at most q (non-adaptive) queries and satisfies the following two conditions:*

- *Accepts codewords (completeness): For any $x \in \{0, 1\}^k$, given oracle access to $C(x)$, machine T accepts with probability 1. That is, $\Pr[T^{C(x)}(1^k) = 1] = 1$, for any $x \in \{0, 1\}^k$.*

- *Rejects words far from C : For any $w \in \{0, 1\}^n$ that is δ -far from C , given oracle access to w , machine T rejects with probability at least $1/2$. That is, $\Pr[M^w(1^k) = 1] \leq 1/2$, for any $w \in \{0, 1\}^n$ that is δ -far from C .*

We call q the query complexity of M , and δ the proximity parameter.

2.4 Probabilistically-Checkable Proofs (PCP)

Definition 2.24 (PCP). A probabilistically checkable proof (PCP) for a relation R consists of a prover P and a PPT verifier V such that the following holds.

1. *Completeness. For every $(x, w) \in R$,*

$$\Pr_{\pi \leftarrow P(x, w)} [V^\pi(x) = 1] = 1.$$

2. *Soundness. For every $x \notin L(R)$ and every oracle π^* ,*

$$\Pr[V^{\pi^*}(x) = 1] \leq \frac{1}{2}.$$

We say that a PCP is efficient if $P(x, w)$ can be computed efficiently for any $(x, w) \in R$ and we say that it has query complexity q if the number of queries the verifier makes is bounded by q .

Definition 2.25 (PCPP). For $\delta : \mathbb{N} \rightarrow [0, 1]$, a probabilistically checkable proof of proximity (PCPP) for a pair language L with proximity parameter δ consists of a prover P and a verifier V such that the following holds for every pair of strings (x, y) .

1. *Completeness. If $(x, y) \in L$,*

$$\Pr_{\pi \leftarrow P(x, y)} [V^{(y, \pi)}(x) = 1] = 1.$$

2. *Soundness. If y is δ -far from the set $L[x] := \{y : (x, y) \in L\}$, then for every oracle π^* ,*

$$\Pr[V^{(y, \pi^*)}(x) = 1] \leq \frac{1}{2}.$$

We say that a PCPP has query complexity q if the number of queries the verifier makes is bounded by q .

2.5 Zero-Knowledge PCPs

Definition 2.26 (View of PCP of proximity). For a PCP of proximity (P, V) and a (possibly malicious) verifier V^* , $\text{View}_{V^*, P}(x, y)$ denotes the view of V^* with explicit input x and oracle access to both implicit input y and $\pi \leftarrow P(x)$. That is, $\text{View}_{V^*, P}(x, y)$ comprises V^* 's random coins and all answers to V^* 's queries to (y, π) .

Definition 2.27 (Black-box l -local perfect zero-knowledge PCP of proximity). Let $l : \mathbb{N} \rightarrow \mathbb{N}$. We say that a PCP of proximity system (P, V) for a pair language L is black-box l -local, perfect zero knowledge with query bound q^* if there exists an algorithm Sim , such that for every (possibly malicious) adaptive q^* -query-bounded verifier V^* , and for every $(x, y) \in L$, $\text{Sim}^{(V^*, y)}(x)$ is distributed identically to $\text{View}_{V^*, P}(x, y)$, and moreover, Sim runs in time $\text{poly}(|x|, q^*)$ and makes at most $q^* \cdot l(n)$ queries to y .

Definition 2.28 (Black-box perfect zero-knowledge PCP). We say that a PCP system (P, V) for a relation R is black-box perfect zero knowledge with query bound q^* if there exists a PPT algorithm Sim , called the simulator, such that for every possibly malicious, adaptive, q^* -query-bounded verifier V^* , and for every $(x, w) \in R$,

$$\text{Sim}^{V^*}(x) \equiv \text{View}_{V^*, P}(x, w),$$

where $\equiv$ denotes that the two random variables are distributed identically.

3 Sheaf Codes

In this section, we present results and definitions related to sheaf codes [PK24], which sets the foundation for the rest of the paper.

3.1 Basic Definitions

We start with some basic definitions.

Definition 3.1 (Poset). *A poset is a set P equipped with a partial order $\leq$.*

Definition 3.2 (Upper Sets of a Poset). *Let P be a poset. For $\sigma \in P$, denote by $P_{\geq \sigma}$ the set $\{x \in P \mid x \geq \sigma\}$. $U \subseteq P$ is an upper set if for all $\sigma \in U$, $P_{\geq \sigma} \subseteq U$; i.e., for all $\sigma \in U$ and $x \in P$ such that $x \geq \sigma$, $x \in U$.*

Definition 3.3 (Maximal Elements). *The set of maximal elements, denoted by P_* , is the set of elements in P that are either greater than or unordered with every other element in P . For $\sigma \in P$, denote by P_σ the set $\{x \in P_* \mid x \geq \sigma\}$, i.e., the maximal elements in $P_{\geq \sigma}$. For set $U \subseteq P$, $P_U = \bigcup_{\sigma \in U} P_\sigma$. Equivalently, $P_U = P_* \cap U$.*

Claim 3.4. *Let P be a poset and U_1, U_2 be upper sets in P . Then, $P_{U_1 \cup U_2} = P_{U_1} \cup P_{U_2}$ and $P_{U_1 \cap U_2} = P_{U_1} \cap P_{U_2}$.*

Proof. First we prove that $P_{U_1 \cup U_2} = P_U$, where $U = U_1 \cup U_2$. By definition, $P_U = U \cap P_*$. Therefore,

$$P_U = U \cap P_* = (U_1 \cup U_2) \cap P_* = (U_1 \cap P_*) \cup (U_2 \cap P_*) = P_{U_1} \cup P_{U_2}$$

Next, we prove that $P_{U_1 \cap U_2} = P_{U_1} \cap P_{U_2}$. We have the following.

$$P_{U_1 \cap U_2} = P_* \cap (U_1 \cap U_2) = (P_* \cap U_1) \cap (P_* \cap U_2) = P_{U_1} \cap P_{U_2}. \quad \square$$

Definition 3.5 (Coded Space). *Given a domain D , a coded space P for D is a poset that contains all elements of D and a collection of subsets of D , which are viewed as domains of local codes. The partial ordering on P satisfies $\sigma \leq \rho$ if and only if $\rho \subseteq \sigma$.*

Remark 3.6. While each $\tau \in P_*$ is technically a singleton set containing a single domain element, following the convention of [PK24], we will abuse notation and denote by τ the domain element itself, i.e. the element inside the singleton set. Using this convention, $P_* = D$, and for any $\sigma \in P$, $P_\sigma = \sigma$.

We start by showing how to apply functions on the maximal elements upper sets.

Definition 3.7 (Permutations on Local Domains). *Let P, P' be coded spaces. Let $g : P_* \rightarrow P'_*$ be a bijection. For each set $U \subseteq P$, we define $g(U)$ as follows.*

$$g(U) := \{g(\sigma) : \sigma \in U\}.$$

Proposition 3.8. *Let P, P' be coded spaces. Let $g : P_* \rightarrow P'_*$ be a bijection and let $U \subseteq P$ be an upper set. Then $g(U)$ is an upper set.*

Proof. Let $\sigma \in g(U)$. Let $\tau \in P'$ such that $\tau \geq \sigma$. Since $\tau \subseteq \sigma$, it follows by Claim 2.2 that $g^{-1}(\tau) \subseteq g^{-1}(\sigma)$, which implies that $g^{-1}(\tau) \in U$, because U is an upper set. Therefore, $\tau = g(g^{-1}(\tau))$, and so $\tau \in g(U)$, which completes the proof. $\square$

Definition 3.9 (Sheaf Of Linear Codes). *Let P be a coded space. A sheaf of linear codes $\mathcal{F}$ on P is a sequence of linear codes $(\mathcal{F}_\sigma)_{\sigma \in P}$ where $\mathcal{F}_\sigma \subseteq \mathbb{F}^\sigma$, such that for every $\sigma, \tau \in P$ with $\sigma \leq \tau$, if $c \in \mathcal{F}_\sigma$ then $c|_\tau \in \mathcal{F}_\tau$. For each upper set $U \subseteq P$, we define the local section*

$$\mathcal{F}(U) = \{c \in \mathbb{F}^{P_U} \mid \forall \sigma \in U : c|_{P_\sigma} \in \mathcal{F}_\sigma\}.$$

When clear from context, we will refer to $\mathcal{F}(P)$ as $\mathcal{F}$. The domain of a sheaf code, denoted $\text{Dom}(\mathcal{F})$, is the domain P_* of its coded space.

Definition 3.10 (Extendable Upper Set). *Let $\mathcal{F}$ be a sheaf code on coded space P . Let U, V be upper sets in P with $U \subseteq V$. Then U is extendable to V with respect to $\mathcal{F}$ if there exists a map $e: \mathcal{F}(U) \rightarrow \mathcal{F}(V)$ such that for all $f \in \mathcal{F}(U)$, $e(f)|_U = f$. We refer to e as an extension map and $e(f)$ as an extension of f in $\mathcal{F}(V)$. If e is unique, we say that U is uniquely extendable to V . We say U is globally extendable if it is extendable to P . Equivalently, U is globally extendable if $\mathcal{F}(P)|_U = \mathcal{F}(U)$.*

When $\mathcal{F}, P$ are clear from context, we drop the “with respect to $\mathcal{F}$ ”.

Proposition 3.11. *If U is uniquely extendable to V then the unique extension map e is a bijection.*

Proof. Clearly e is an injection; it remains to show surjectivity. Suppose otherwise; then there exists $c \in \mathcal{F}(V)$ that is not in the image of e . But then $e(c|_U) = c' \neq c$, and both c and c' are extensions of $c|_U$, which contradicts the uniqueness of e . $\square$

Definition 3.12 (Code Cover). *Let P be a coded space. $S \subseteq P$ is a cover of P if for all $x \in P_*$, there exists $\rho \in S$ such that $x \geq \rho$.*

Definition 3.13 (Upper Set Closure). *Let P be a poset. Let $S \subseteq P$ be a set of elements in P . The upper set closure of S , denoted by U_S , is equal to $\bigcup_{\sigma \in S} P_{\geq \sigma}$.*

3.2 The Dimensions Of Local Codes

In this subsection, we show that the padded dual of $\mathcal{F}(U)$ decomposes into the sum of the padded duals of the local codes in U . We then use this to derive the dimension of $\mathcal{F}(U)$ from the dimensions of its local codes. This will be useful later when proving extendability.

Lemma 3.14 (Decomposition of Local (Padded) Dual Codes). *Let $\mathcal{F}$ be a sheaf code on coded space P . Let U be an upper set in P . Then,*

$$\overline{\mathcal{F}(U)}^\perp = \sum_{\sigma \in U} \overline{\mathcal{F}_\sigma}^\perp.$$

Proof. To show $\overline{\mathcal{F}(U)}^\perp \subseteq \sum_{\sigma \in U} \overline{\mathcal{F}_\sigma}^\perp$, it suffices to show that $(\sum_{\sigma \in U} \overline{\mathcal{F}_\sigma}^\perp)^\perp \subseteq \widehat{\mathcal{F}(U)}$. By Claim 2.14, $(\sum_{\sigma \in U} \overline{\mathcal{F}_\sigma}^\perp)^\perp = \bigcap_{\sigma \in U} \overline{\mathcal{F}_\sigma}^{\perp\perp} = \bigcap_{\sigma \in U} \widehat{\mathcal{F}_\sigma}$. Next, we prove the following claim.

Claim 3.15.

$$\bigcap_{\sigma \in U} \widehat{\mathcal{F}_\sigma} \subseteq \widehat{\mathcal{F}(U)}_D.$$

Proof. If $v \in \bigcap_{\sigma \in U} \widehat{\mathcal{F}_\sigma}$ then $v \in \widehat{\mathcal{F}_\sigma}$ for all $\sigma \in U$, which by Claim 2.16, implies that $v|_{P_\sigma} \in \mathcal{F}_\sigma$ for all $\sigma \in U$. Therefore, by definition of sheaf of linear codes (Definition 3.9), $v|_{P_U} \in \mathcal{F}(U)$, and so by Claim 2.16, $v \in \widehat{\mathcal{F}(U)}$. $\square$

Next, we show that $\sum_{\sigma \in U} \overline{\mathcal{F}_\sigma}^\perp \subseteq \overline{\mathcal{F}(U)}^\perp$. Let $v \in \sum_{\sigma \in U} \overline{\mathcal{F}_\sigma}^\perp$. Then, there exists a set $\{v_\sigma \in \overline{\mathcal{F}_\sigma}^\perp : \sigma \in U\}$ such that $v = \sum_{\sigma \in U} v_\sigma$. We will use the following fact, which shows that v is in the dual of $\widehat{\mathcal{F}(U)}$.

Fact 3.16. For all $c \in \widehat{\mathcal{F}(U)}$,

$$\langle c, v \rangle = 0.$$

Proof.

$$\langle c, v \rangle = \langle c, \sum_{\sigma \in U} v_\sigma \rangle = \sum_{\sigma \in U} \langle c, v_\sigma \rangle = \sum_{\sigma \in U} \langle c|_{P_\sigma}, v_\sigma|_{P_\sigma} \rangle = 0.$$

The second to last equality follows because for each $\sigma \in U$, $\text{supp}(\overline{\mathcal{F}_\sigma^\perp}) \subseteq P_\sigma$. The last equality follows because by Claim 2.16 and definition of sheaf of linear codes (Definition 3.9), for all $\sigma \in U$, $c|_{P_\sigma} \in \mathcal{F}_\sigma$ and by definition of padded vector space (Definition 2.9), $v_\sigma|_{P_\sigma} \in \mathcal{F}_\sigma^\perp$. $\square$

Therefore, by definition of dual (Definition 2.13), $v \in (\widehat{\mathcal{F}(U)}_D)^\perp = \overline{\mathcal{F}(U)}^\perp$, which completes the proof. $\square$

Corollary 3.17. *Let $\mathcal{F}$ be a sheaf code on coded space P . Let U_1, U_2 be upper sets in P . Then,*

$$\overline{\mathcal{F}(U_1 \cup U_2)}^\perp = \overline{\mathcal{F}(U_1)}^\perp + \overline{\mathcal{F}(U_2)}^\perp,$$

Proof.

$$\overline{\mathcal{F}(U_1 \cup U_2)}^\perp = \sum_{\sigma \in U_1 \cup U_2} \overline{\mathcal{F}_\sigma^\perp} = \sum_{\sigma \in U_1} \overline{\mathcal{F}_\sigma^\perp} + \sum_{\sigma \in U_2} \overline{\mathcal{F}_\sigma^\perp} = \overline{\mathcal{F}(U_1)}^\perp + \overline{\mathcal{F}(U_2)}^\perp$$

where the first equality follows by Lemma 3.14, the second equality follows by definition of summation, and the third equality once again follows by Lemma 3.14. $\square$

Lemma 3.18 (Dimension of $\mathcal{F}(U)$). *Let $\mathcal{F}$ be a sheaf code on coded space P . Let U_1, U_2 be upper sets in P such that $U = U_1 \cup U_2$. Then,*

$$\dim(\mathcal{F}(U)) = \dim(\mathcal{F}(U_1)) + \dim(\mathcal{F}(U_2)) - (|P_{U_1} \cap P_{U_2}| - \dim(\overline{\mathcal{F}(U_1)}^\perp \cap \overline{\mathcal{F}(U_2)}^\perp)).$$

Proof.

$$\begin{aligned} \dim(\mathcal{F}(U)) &= |P_U| - \dim(\mathcal{F}(U)^\perp) && \text{(By Claim 2.15)} \\ &= |P_U| - \dim(\overline{\mathcal{F}(U)}^\perp) && \text{(By Corollary 2.11)} \\ &= |P_U| - \dim(\overline{\mathcal{F}(U_1)}^\perp + \overline{\mathcal{F}(U_2)}^\perp) && \text{(By Corollary 3.17)} \\ &= |P_{U_1}| + |P_{U_2}| - |P_{U_1 \cap U_2}| && \text{(By Claim 3.4)} \\ &\quad - \left(\dim(\overline{\mathcal{F}(U_1)}^\perp) + \dim(\overline{\mathcal{F}(U_2)}^\perp) - \dim(\overline{\mathcal{F}(U_1)}^\perp \cap \overline{\mathcal{F}(U_2)}^\perp) \right) && \text{(By Claim 2.6)} \\ &= (|P_{U_1}| - \dim(\overline{\mathcal{F}(U_1)}^\perp)) + (|P_{U_2}| - \dim(\overline{\mathcal{F}(U_2)}^\perp)) - (|P_{U_1 \cap U_2}| - \dim(\overline{\mathcal{F}(U_1)}^\perp \cap \overline{\mathcal{F}(U_2)}^\perp)) \\ &= (|P_{U_1}| - \dim(\mathcal{F}(U_1)^\perp)) + (|P_{U_2}| - \dim(\mathcal{F}(U_2)^\perp)) - (|P_{U_1 \cap U_2}| - \dim(\overline{\mathcal{F}(U_1)}^\perp \cap \overline{\mathcal{F}(U_2)}^\perp)) && \text{(By Claim 2.11)} \\ &= \dim(\mathcal{F}(U_1)) + \dim(\mathcal{F}(U_2)) - (|P_{U_1} \cap P_{U_2}| - \dim(\overline{\mathcal{F}(U_1)}^\perp \cap \overline{\mathcal{F}(U_2)}^\perp)) && \text{(By Claim 2.15)} \end{aligned}$$

$\square$

Corollary 3.19. *Let $\mathcal{F}$ be a sheaf code on coded space P . Let U_1, U_2 be upper sets in P such that $U = U_1 \cup U_2$ and $\overline{\mathcal{F}(U_1)}^\perp \cap \overline{\mathcal{F}(U_2)}^\perp = \{\vec{0}\}$. Then,*

$$\dim(\mathcal{F}(U)) = \dim(\mathcal{F}(U_1)) + \dim(\mathcal{F}(U_2)) - |P_{U_1} \cap P_{U_2}|.$$

Proof. Follows directly from Lemma 3.18 since $\dim(\{\vec{0}\}) = 0$. $\square$

3.3 Isomorphism of Sheaf Codes

In this section, we define a sheaf code isomorphism, which preserves local restrictions. In the next subsection, we will use this to define merged sheaf codes.

Definition 3.20 (Isomorphism of sheaf codes). *Let $\mathcal{F}, \mathcal{F}'$ be sheaf codes on P, P' respectively, and let $U \subseteq P, V \subseteq P'$ be two upper sets. Let $g : P_U \rightarrow P'_V$ be a bijection. Then g is an isomorphism at U if*

$$g(\mathcal{F}_\sigma) = \mathcal{F}'_{g(\sigma)} \quad \forall \sigma \in U.$$

For the remainder of the section, $\mathcal{F}, \mathcal{F}'$ are two sheaf codes on coded spaces $P, P', U \subseteq P, V \subseteq P'$ are two upper sets, and $g : P_U \rightarrow P'_V$ is a sheaf code isomorphism at U .

Proposition 3.21. $g^{-1} : P'_V \rightarrow P_U$ is a sheaf code at V .

Proof. For all $\sigma \in V$,

$$g^{-1}(\mathcal{F}'_\sigma) = g^{-1}(g(\mathcal{F}_{g^{-1}(\sigma)})) = \mathcal{F}_{g^{-1}(\sigma)}.$$

The first equality follows because g is a sheaf code isomorphism at U and $g^{-1}(\sigma) \in U$. The second equality follows because g has an inverse. This completes the proof. $\square$

Proposition 3.22. Let $W \subseteq U$ be an upper set on P . Then $g|_{P_W}$ is a sheaf code isomorphism at W .

Proof. For all $\sigma \in W$, $g|_{P_W}(\mathcal{F}_\sigma) = \mathcal{F}'_{g(\sigma)}$ because g is a sheaf code isomorphism on U and $\sigma \in W \subseteq U$. $\square$

Proposition 3.23 (Composition). *Let $\hat{\mathcal{F}}$ be a sheaf code with coded space $\hat{P}$ and let $W \subseteq \hat{P}$ be an upper set. Let $h : P_W \rightarrow P_U$ be a bijection such that $h(W) \subseteq U$. If h is a sheaf code isomorphism at V , then $g \circ h : \hat{\mathcal{F}} \rightarrow \mathcal{F}'$ is a sheaf code isomorphism at V .*

Proof. Let $\sigma \in V$.

$$g(h(\hat{\mathcal{F}}_\sigma)) = g(\mathcal{F}_{h(\sigma)}) = \mathcal{F}'_{g(h(\sigma))}.$$

The first equality follows because h is a sheaf code isomorphism at V , the second equality follows because g is a sheaf code isomorphism at U and by assumption, $h(\sigma) \in U$. This completes the proof. $\square$

Proposition 3.24 (Isomorphisms preserve local sections).

$$g(\mathcal{F}(U)) = \mathcal{F}'(g(U)).$$

Proof. Let $v \in g(\mathcal{F}(U))$. For all $\sigma \in g(U)$, we have the following.

$$v|_\sigma \in g(\mathcal{F}(U))|_\sigma = g(\mathcal{F}(U)|_{g^{-1}(\sigma)}) \subseteq g(\mathcal{F}_{g^{-1}(\sigma)}) = \mathcal{F}'_\sigma.$$

The first inclusion follows by definition of vector space restriction, the first equality follows by Proposition 2.8, the second inclusion follows since U is an upper set and $\hat{\mathcal{F}}$ is a sheaf code and the final equality follows by definition of sheaf isomorphism. Next let $w \in \mathcal{F}'(g(U))$. We show that $g^{-1}(w) \in \mathcal{F}(U)$, which is sufficient to prove the claim. For all $\sigma \in U$,

$$g^{-1}(w)|_\sigma = g^{-1}(w|_{g(\sigma)}) \in g^{-1}(\mathcal{F}'_{g(\sigma)}) = g^{-1}(g(\mathcal{F}_\sigma)) = \mathcal{F}_\sigma.$$

The first equality follows from Proposition 2.8, the inclusion follows because $\hat{g}(U)$ is an upper set, the second equality follows by definition of sheaf isomorphism, and the final equality follows from definition of permutation map. $\square$

Lemma 3.25 (Isomorphisms preserve extendability). *Let $X, W \subseteq U$ be two upper sets. If X is extendable to W with respect to $\mathcal{F}$, then $g(X)$ is extendable to $g(W)$ with respect to $\mathcal{F}'$.*

Proof. Let $f \in \mathcal{F}'(g(V))$. By Proposition 3.24, $\mathcal{F}'(g(V)) = g(\mathcal{F}(V))$. Therefore, $g^{-1}(f) \in \mathcal{F}(V)$. Let $h \in \mathcal{F}(W)$ be the extension of $g^{-1}(f)$ in W , which must exist since V is extendable to W . We show next that $g(h)$ is an extension of f in $g(W)$.

$$g(h)|_{g(V)} = g(h|_V) = g(g^{-1}(f)) = f.$$

The first equality follows by Proposition 2.8, the second follows by definition of h and the third follows because g is a bijection. $\square$

3.4 Merging Sheaf Codes

In this subsection, we show how to merge multiple sheaf codes into one. We will use this technique in Section 6 to construct our zkPCPP. The construction takes as input a set of pairs $(\hat{\mathcal{F}}, \hat{g})$ where $\hat{\mathcal{F}}$ is a sheaf code and $\hat{g}: \text{Dom}(\hat{\mathcal{F}}) \rightarrow \mathcal{U}$ is an injection that maps the domain of $\hat{\mathcal{F}}$ into a common universe $\mathcal{U}$. The role of the $\hat{g}$ is to “glue” the different sheaf codes together to create one conglomerate sheaf code. All of the local domains in all of the individual coded spaces will have a corresponding local domain in the final merged coded space. We additionally include the domain $\text{Dom}(\hat{\mathcal{F}})$ in the final coded space for each pair, $(\hat{\mathcal{F}}, \hat{g})$, being merged.

For a set $S \subseteq \text{Dom}(\hat{\mathcal{F}})$, we denote by $\hat{g}(S) = \{\hat{g}(x) : x \in S\}$. Recall that each $\sigma \in \hat{P}$ denotes a set as well as a local code. Furthermore, recall Definition 3.7 which provides the notation $g(U)$ for mapping upper sets with bijections.

Definition 3.26 (Merging Sheaf Codes). *Fix a universe $\mathcal{U}$. Let A be a set of pairs $(\hat{\mathcal{F}}, \hat{g})$ where $\hat{\mathcal{F}}$ is a sheaf code and $\hat{g}: \text{Dom}(\hat{\mathcal{F}}) \rightarrow \mathcal{U}$ is an injection, and suppose A satisfies the following property: for all $(\hat{\mathcal{F}}, \hat{g}), (\hat{\mathcal{F}}', \hat{g}') \in A$, and all $\rho \in \hat{P}'$ such that there exists $\sigma \in \hat{P}$ with $\hat{g}(\sigma) \supseteq \hat{g}'(\rho)$, the composition $\hat{g}^{-1} \circ \hat{g}'|_\rho$ is a sheaf code isomorphism at upper set $\hat{P}'_{\geq \rho}$.*

Then we define the merged sheaf code, $\mathcal{F} = M(A)$ as follows.

$$P := \bigcup \{\hat{g}(\hat{P}) : (\hat{\mathcal{F}}, \hat{g}) \in A\} \cup \{\hat{g}(\hat{P}_*) : (\hat{\mathcal{F}}, \hat{g}) \in A\} \quad (3)$$

$$\mathcal{F}_{\hat{g}(\alpha)} := \hat{g}(\hat{\mathcal{F}}_\alpha) \quad \forall (\hat{\mathcal{F}}, \hat{g}) \in A, \alpha \in \hat{P} \quad \mathcal{F}_{\hat{g}(\hat{P}_*)} := \hat{g}(\hat{\mathcal{F}}) \quad \forall (\hat{\mathcal{F}}, \hat{g}) \in A \quad (4)$$

For the following claims and lemma, let A be as specified in Definition 3.26. We first show that $M(A)$ is indeed a sheaf code.

Proposition 3.27. $\mathcal{F} = M(A)$ is a sheaf code.

Proof. Let P be the coded space for $\mathcal{F}$. It suffices to prove that for every $\sigma, \tau \in P$ with $\sigma \leq \tau$, $\mathcal{F}_\sigma|_\tau \subseteq \mathcal{F}_\tau$. Let $\sigma, \tau \in P$ with $\sigma \leq \tau$. By construction of $M(A)$ there exist $(\hat{\mathcal{F}}, \hat{g}), (\hat{\mathcal{F}}', \hat{g}') \in A$ such that $\sigma = \hat{g}(\alpha), \tau = \hat{g}'(\beta)$. Let $h := \hat{g}^{-1} \circ \hat{g}'$. We prove the following.

Fact 3.28.

$$\hat{g}'(\hat{\mathcal{F}}'_\beta) = \hat{g}(\hat{\mathcal{F}}_{h(\beta)}).$$

Proof. Since $\hat{g}(\alpha) \leq \hat{g}'(\beta)$, it follows by ordering on a coded space that $\hat{g}'(\beta) \subseteq \hat{g}(\alpha)$ and therefore by condition on A , $h = \hat{g}^{-1} \circ \hat{g}'$ is a sheaf code isomorphism at upper set $\hat{P}'_{\geq \beta}$. Therefore, by definition of sheaf code isomorphism (Definition 3.20), since $\beta \in \hat{P}'_{\geq \beta}$, it follows that

$$h(\hat{\mathcal{F}}'_\beta) = \hat{\mathcal{F}}_{h(\beta)}.$$

Therefore, applying the definition of h completes the proof. $\square$

$$\begin{aligned}
\mathcal{F}_\sigma|_\tau &= \mathcal{F}_{\hat{g}(\alpha)}|_{\hat{g}'(\beta)} \\
&= \hat{g}(\hat{\mathcal{F}}_\alpha)|_{\hat{g}'(\beta)} && \text{(by definition of merged sheaf)} \\
&= \hat{g}(\hat{\mathcal{F}}_\alpha)|_{\hat{g}(h(\beta))} && \text{(by definition of } h) \\
&= \hat{g}(\hat{\mathcal{F}}_\alpha|_{h(\beta)}) && \text{(by Proposition 2.8)} \\
&\subseteq \hat{g}(\hat{\mathcal{F}}_{h(\beta)}) && \text{(since } \hat{\mathcal{F}} \text{ is a sheaf code)} \\
&= \hat{g}'(\hat{\mathcal{F}}'_\beta) && \text{(by Fact 3.28)} \\
&= \mathcal{F}_{\hat{g}'(\beta)} && \text{(By condition (4))} \\
&= \mathcal{F}_\tau. && \square
\end{aligned}$$

Proposition 3.29. For each $(\hat{\mathcal{F}}, \hat{g}) \in A$, define the bijection $\hat{k} : \text{Dom}(\hat{\mathcal{F}}) \rightarrow \hat{g}(\text{Dom}(\hat{\mathcal{F}}))$ as $\hat{k}(x) = \hat{g}(x)$ for all $x \in \text{Dom}(\hat{\mathcal{F}})$. Then $\hat{k}$ is a sheaf code isomorphism (Definition 3.20) between $\hat{\mathcal{F}}$ and $\mathcal{F} = M(A)$ at upper set $\hat{P}$.

Proof. Let $(\hat{\mathcal{F}}, \hat{g}) \in A$. By definition of merged sheaf code, for each $\sigma \in \hat{P}$, $\hat{k}(\hat{\mathcal{F}}_\sigma) = \hat{g}(\hat{\mathcal{F}}_\sigma) = \mathcal{F}_{\hat{g}(\sigma)} = \mathcal{F}_{\hat{k}(\sigma)}$, which is exactly the definition of sheaf code isomorphism. $\square$

4 Testability of Sheaf Codes

In this section, we show how to use sheaf codes to test proximity of a function to a codeword.

4.1 Definitions

Definition 4.1 (Testable cover). Let $\mathcal{F}$ be a sheaf code with coded space P , and let $\rho \in P$. Let S be a cover for P_ρ . For $c \in \mathbb{N}$, S is a c -testable cover for $\mathcal{F}_\rho$ with distribution $\mathcal{D}(S)$ over S if for all functions $f : P_* \rightarrow \mathbb{F}$ and all $\delta \in (0, 1]$, if

$$\mathbb{E}_{\sigma \leftarrow \mathcal{D}(S)} [\Delta(f|_\sigma, \mathcal{F}_\sigma)] \leq \delta,$$

then,

$$\Delta(f|_\rho, \mathcal{F}_\rho) \leq c \cdot \delta.$$

Definition 4.2 (d -testable cover). Let $\mathcal{F}$ be a sheaf of linear codes, P its coded space, $\sigma \in P$ and S a c -testable cover for $\mathcal{F}_\sigma$. Then S is a $(c, 1)$ -testable cover. Moreover, for $d > 1$, S is a (c, d) -testable cover if for all $\tau \in S$, $\mathcal{F}_\tau$ has a $(c, d-1)$ -testable cover in $\mathcal{F}$. For $i \leq d$, denote by $\mathcal{S}_i \subseteq P$ the set of codes that are in an i -testable cover and are at most $(c', i-1)$ testable for some constant c' .

Definition 4.3 (d -testable sheaf code). Let $\mathcal{F}$ be a sheaf of linear codes, P its coded space, and let $\sigma \in P$. $\mathcal{F}$ is a (c, d) -testable sheaf code for $\mathcal{F}_\sigma$ if it contains a (c, d) -testable cover for $\mathcal{F}_\sigma$.

Lemma 4.4. Let $\mathcal{F}, \mathcal{F}'$ be two sheaf codes, let $U \subseteq P$ be an upper set, let $h : P \rightarrow P'$ be a sheaf code isomorphism on upper set U , let $S \subseteq U$ be a c -testable cover for $\sigma \in U$. Then $h(S)$ is a c -testable cover for $h(\sigma) \in h(U)$.

Proof. Define $\mathcal{D}(h(S))$ to be the distribution where the probability of sampling $\rho \leftarrow \mathcal{D}(h(S))$ is equal

to the probability of sampling $h^{-1}(\rho) \leftarrow \mathcal{D}(S)$. Then, we have the following.

$$\begin{aligned}
& \mathbb{E}_{\rho \leftarrow \mathcal{D}(h(S))} [\Delta(f|_{\rho}, h(\mathcal{F}_{\rho}))] \\
&= \mathbb{E}_{\rho \leftarrow \mathcal{D}(h(S))} [\Delta(f|_{\rho}, h(\mathcal{F}_{\rho}))] \\
&= \mathbb{E}_{\rho \leftarrow \mathcal{D}(h(S))} [\Delta(h^{-1}(f|_{\rho}), h^{-1}(h(\mathcal{F}_{\rho})))] \\
&= \mathbb{E}_{\rho \leftarrow \mathcal{D}(h(S))} [\Delta(h^{-1}(f)|_{h^{-1}(\rho)}, \mathcal{F}_{\rho})] \\
&= \mathbb{E}_{h^{-1}(\rho) \leftarrow \mathcal{D}(S)} [\Delta(h^{-1}(f)|_{h^{-1}(\rho)}, \mathcal{F}_{\rho})] \\
&= \mathbb{E}_{\alpha \leftarrow \mathcal{D}(S)} [\Delta(h^{-1}(f)|_{\alpha}, \mathcal{F}_{\rho})] \\
&\leq \delta
\end{aligned}$$

Therefore, since S is a c -testable cover, $\Delta(h^{-1}(f), \mathcal{F}_{\sigma}) \leq c \cdot \delta$. Therefore $\Delta(f, h(\mathcal{F}_{\sigma})) \leq c \cdot \delta$, which completes the proof. $\square$

Corollary 4.5. *Let $\mathcal{F}, \mathcal{F}'$ be two sheaf codes, let $U \subseteq P$ be an upper set, let h be a sheaf code isomorphism at upper set U , let $S \subseteq U$ be a (c, d) -testable cover for $\sigma \in U$. Then $h(S)$ is a (c, d) -testable cover for $h(\sigma)$ in $h(U)$.*

Proof. We proceed with proof by induction on d . When $d = 1$, we apply Lemma 4.4. Next, let $d \in \mathbb{N}$. We show that for each $\rho \in h(S)$, $h(\mathcal{F})$ contains a $d - 1$ -testable cover for ρ . Since S is a d -testable cover, it follows that $h^{-1}(\rho)$ has a $d - 1$ testable cover in $\mathcal{F}$. By inductive hypothesis, this implies that $h(\mathcal{F})$ contains a $d - 1$ testable cover for ρ , which completes the proof. $\square$

4.2 Recursive Testing of Sheaf Codes

In this subsection, we construct a codeword test for (c, d) -testable sheaf codes. This will be used as a building block for the PCPP presented in the next section.

Construction 4.6. Let $c, d \in \mathbb{N}$, let $\mathcal{F}$ be a sheaf of linear codes, let P be a coded space, let $\sigma \in P$, and let S be a (c, d) -testable cover for $\mathcal{F}_{\sigma}$ according to distribution $\mathcal{D}(S)$. This construction takes a function $f : \text{Dom}(\mathcal{F}) \rightarrow \mathbb{F}$, σ , and S as input.

- If $d = 0$, then check that $f|_{\sigma} \in \mathcal{F}_{\sigma}$. If it is, output **yes**, otherwise output **no**.
- Sample $\rho \leftarrow \mathcal{D}(S)$ and let S_{ρ} be a $(d - 1)$ -testable cover of $\mathcal{F}_{\rho}$. Run this test recursively on inputs f , ρ , and S_{ρ} .

Theorem 4.7. *For any $\epsilon > 0$, Construction 4.6 is a $(\max_{\alpha \in \mathcal{S}_1} (|\alpha|))$ -local codeword $\epsilon \cdot c^d$ -tester for $\mathcal{F}_{\sigma}$.*

Proof. Completeness. If $d = 0$, then the construction outputs **yes** with probability 1 because by definition of sheaf code, $f \in \mathcal{F}(P)$ implies that $f|_{P_{\sigma}} \in \mathcal{F}_{\sigma}$. For $d > 1$, by definition of sheaf code, $f|_{P_{\rho}} \in \mathcal{F}_{\rho}$ and by assumption $\mathcal{F}_{\rho}$ has a $(d - 1)$ -testable cover, S_{ρ} . Therefore, by the inductive hypothesis, invoking $T(f, \rho, S_{\rho})$ outputs **yes** with probability 1.

Soundness. We show that if $\Pr[T(f, \sigma, S) = \text{no}] \leq \epsilon$ then $\Delta(f|_{P_{\sigma}}, \mathcal{F}_{\sigma}) \leq \epsilon \cdot c^d$. We proceed via induction. For the base case, let $d = 0$. Since the verifier accepts with probability strictly greater than zero, it follows that $f|_{P_{\sigma}} \in \mathcal{F}_{\sigma}$, since this condition is checked directly. Next, for the inductive step, for each $\tau \in S$, let ϵ_{τ} be the failure probability of $T(f, \tau, S_{\tau})$. Then,

$$\mathbb{E}_{\tau \leftarrow \mathcal{D}(S)} [\Delta(f|_{P_{\tau}}, \mathcal{F}_{\tau})] \leq \mathbb{E}_{\tau \leftarrow \mathcal{D}(S)} [\epsilon_{\tau} \cdot c^{d-1}] = \mathbb{E}_{\tau \leftarrow \mathcal{D}(S)} [\epsilon_{\tau}] \cdot c^{d-1}, \quad (5)$$

where the inequality follows by the inductive hypothesis, and the equality follows by linearity of expectation. Next, observe that

$$\mathbb{E}_{\tau \leftarrow \mathcal{D}(S)}[\epsilon_\tau] = \Pr[T(f, \sigma, S) = \text{no}] \leq \epsilon$$

where the inequality is by assumption. Hence

$$\mathbb{E}_{\tau \leftarrow \mathcal{D}(S)}[\Delta(f|_{P_\tau}, \mathcal{F}_\tau)] \leq \epsilon \cdot c^{d-1},$$

where the last inequality follows by assumption of the theorem statement. Since S is a c -testable cover for $\mathcal{F}_\sigma$ according to $\mathcal{D}(S)$, it follows by Definition 4.1 that $\Delta(f|_{P_\sigma}, \mathcal{F}_\sigma) \leq c \cdot \epsilon \cdot c^{d-1} = \epsilon \cdot c^d$. $\square$

4.3 Claims Related To Testability Of Sheaf Codes

The following lemma will be useful in Section 7.3, when we prove that a testable cover for Reed-Muller codes is a testable cover for its local Reed-Solomon codes with sufficiently large domains.

Lemma 4.8 (Distance Of Local Functions From Local Codes). *Let $\mathcal{F}$ be a sheaf code with coded space P , and let $\sigma \in P$. Let $f : \text{Dom}(\mathcal{F}) \rightarrow \mathbb{F}$ be a function. Then*

$$\Delta(f|_\sigma, \mathcal{F}_\sigma) \leq \Delta(f, \mathcal{F}) \cdot \frac{|\text{Dom}(\mathcal{F})|}{|\sigma|}. \quad (6)$$

Proof. Let c be a codeword in $\mathcal{F}$ that is closest to f in Hamming distance, breaking ties arbitrarily. Let $E \subseteq \text{Dom}(\mathcal{F})$ be a maximal subset such that $f|_E \neq c|_E$. By definition, $|E| = \Delta(f, \mathcal{F}) \cdot |\text{Dom}(\mathcal{F})|$. Let $E' = E \cap \sigma$. Then $|E'| \leq |E|$ and so we have the following.

$$\Delta(f|_\sigma, c|_\sigma) = \frac{|E'|}{|\sigma|} \leq \frac{|E|}{|\sigma|} = \frac{\Delta(f, \mathcal{F}) \cdot |\text{Dom}(\mathcal{F})|}{|\sigma|}.$$

By definition of sheaf code, $c|_\sigma \in \mathcal{F}_\sigma$, which completes the proof. $\square$

Corollary 4.9 (Testable Covers Of Large Subcodes). *Let $c \in \mathbb{N}$, let $\mathcal{F}$ be a sheaf code with coded space P , and let $\rho \in P$. Let S be a c -testable cover according to distribution $\mathcal{D}(S)$ for $\mathcal{F}_\rho$ and let $\sigma \in P$ such that $\sigma \supseteq \rho$. If there exists a constant $k \in \mathbb{N}$ such that*

$$\frac{|\rho|}{|\sigma|} = k,$$

then S is a $(c \cdot k)$ -testable cover for $\mathcal{F}_\sigma$ according to distribution $\mathcal{D}(S)$.

Proof. First, if $x \in \sigma$ then it is also in ρ . Since S is a cover of ρ , there must exist $\tau \in S$ such that $x \in \tau$. Therefore, S is a cover for $\mathcal{F}_\sigma$. Next, let $f : \rho \rightarrow \mathbb{F}$ be a function. If $\mathbb{E}_{\tau \leftarrow \mathcal{D}(S)}[\Delta(f|_\tau, \mathcal{F}_\tau)] \leq \delta$ then by definition of c -testable cover, there exists $c \in \mathbb{N}$ such that $\Delta(f, \rho) \leq c \cdot \delta$. Therefore, by Lemma 4.8,

$$\Delta(f|_\sigma, \mathcal{F}_\sigma) \leq \Delta(f, \mathcal{F}_\rho) \cdot \frac{|\rho|}{|\sigma|} \leq c \cdot k \cdot \delta,$$

which completes the proof. $\square$

The following claim considers a code cover where the domains of the local codes are pairwise disjoint and the local codes all belong to the same space. This will be useful in Lemma 7.16, when proving that our final Reed-Muller sheaf code contains a testable cover.

Claim 4.10 (Probability a Local Restriction Is Far From Its Code). *Let $\mathcal{F}$ be a sheaf code over coded space P . Let S be a cover for P such that for each $\tau \neq \rho \in S$, $\tau \cap \rho = \emptyset$ and there exists a code C such that for each $\tau \in S$, $\mathcal{F}_\tau = C$. Let $c \in \mathcal{F}(P)$, $f : \text{Dom}(\mathcal{F}) \rightarrow \mathbb{F}$ and let $\epsilon := \Delta(c, f)$. Then for all $\delta \in (0, 1]$,*

$$\Pr_{\tau \in S} [\Delta(c|_\tau, f|_\tau) > \delta] \leq \frac{\epsilon}{\delta}.$$

Proof. Let $S_\delta := \{\tau \in S : \Delta(f|_\tau, c|_\tau) > \delta\}$. Let D_S be the domain of C , which by definition is equal to τ for each $\tau \in S$.

$$\begin{aligned} \Delta(f, c) \cdot |\text{Dom}(\mathcal{F})| &= \sum_{\tau \in S} \Delta(f|_\tau, c|_\tau) \cdot |\tau| \quad (\text{Since domains of codes in } S \text{ are pairwise disjoint}) \\ &\geq \sum_{\rho \in S_\delta} \delta \cdot |\rho| + \sum_{\tau \in S \setminus S_\delta} \Delta(f|_\tau, c|_\tau) \cdot |\tau| \quad (\text{By definition of } S_\delta) \\ &\geq \sum_{\rho \in S_\delta} \delta \cdot |\rho| + 0 \quad (\text{Since all terms are positive}) \\ &= |S_\delta| \cdot \delta \cdot |D_S| \quad (\text{By definition of } D_S) \end{aligned}$$

Therefore, $|S_\delta| \leq \frac{\Delta(f, c) \cdot |\text{Dom}(\mathcal{F})|}{\delta \cdot |D_S|} \leq \frac{\epsilon \cdot |\text{Dom}(\mathcal{F})|}{\delta \cdot |D_S|}$. By definition of S , $|\text{Dom}(\mathcal{F})| = |S| \cdot |D_S|$. Therefore $|S_\delta| \leq \frac{\epsilon |S|}{\delta}$. Finally, we have the following.

$$\Pr_{\beta \in S} [\Delta(f|_\tau, c|_\tau) > \delta] = \frac{|S_\delta|}{|S|} \leq \frac{\epsilon |S|}{\delta |S|} = \frac{\epsilon}{\delta}.$$

□

5 Locally Simulatable Sheaf Codes

5.1 Basic Definitions

In this subsection, we provide everything we need to prove local simulatability of bivariate Reed-Muller codes. We start by defining locally simulatable encodings in the language of sheaf codes. The following definitions are modified from [GOS24], but put into the language of sheaf codes. Instead of point queries, the simulator accepts local codes in a cover of the sheaf code. In this subsection, $\mathcal{F}, P$ is a sheaf of linear codes and it's associated coded space.

Definition 5.1 (Linear randomised encoding function). *We say that $\text{ENC} : M \rightarrow \mathcal{F}$ is a linear randomised encoding function if $\text{ENC}(m)$ is distributed as $\text{ENC}(m; \mu)$ where $\mu \sim U(F^r)$ and $\text{ENC} : M \times F^r \rightarrow \mathcal{F}(P)$ is a (deterministic) linear function (of both message and randomness).*

Definition 5.2 (Locally simulatable encoding). *Let $\text{ENC} : \mathcal{M} \rightarrow \mathcal{F}(P)$ be a randomized encoding function, let S be a cover of $\mathcal{F}(P)$, let $t : \mathbb{N} \rightarrow \mathbb{N}$, let $\ell : \mathbb{N} \rightarrow \mathbb{N}$, and let $q \in \mathbb{N}$. We say that (ENC, S) is (t, ℓ, q) -locally simulatable if there exists a probabilistic algorithm Sim , called a local simulator, which receives oracle access to a message $m \in \mathcal{M}$, a set $T \subseteq S$ of local codes such that $|T| = n$, a function $f : P_{U_T} \rightarrow \mathbb{F}$, and a new query $\alpha \in S$, runs in time $t(n)$, and makes at most $\ell(n)$ queries to m , satisfying:*

$$\Pr[\mathcal{S}^m(T, \alpha) = \beta] = \Pr[\text{ENC}(m)|_\alpha = \beta \mid \text{ENC}(m)|_\rho = f(\rho) \ \forall \rho \in T],$$

for all $m \in \mathcal{M}$, $\alpha \in \mathcal{D}$, $\beta \in \Sigma$, and for all T such that $|T| < q$ and there exists $c \in \text{supp}(\text{ENC}(m))$ with $c_\rho = y$ for all $\rho \in T$.

The following is the definition of κ -local cover (Definition 4.15) from [BCFGRS17], rewritten to fit this context.

Definition 5.3 (κ -extendable cover). *Let $\mathcal{F}$ be a sheaf of linear codes on P . Let $S \subseteq P$ be a cover of P (Definition 3.12). Then S is a κ -extendable cover with respect to $\mathcal{F}$ if, for every $S' \subseteq S$ such that $|S'| \leq \kappa$, $U_{S'}$ is globally extendable (Definition 3.10) with respect to $\mathcal{F}$.*

5.2 Extendability Implies Local Simulatability

This is a short section describing how to go from sheaf codes to locally simulatable codes. We show that it suffices to find sheafs with good extendability properties.

Definition 5.4 (Extension Sheaf Code). *Let $U \subseteq P$ be an upper set and let $d \in \mathcal{F}(U)$. An extension code for d , which we denote $\mathcal{F}_{U=d}$, is the function mapping each upper set, $V \subseteq P$, to a subset of $\mathcal{F}(V)$ as follows.*

$$\mathcal{F}_{U=d}(V) = \{c \in \mathcal{F}(V) : c|_{P_V \cap P_U} = d|_{P_V \cap P_U}\}.$$

We refer to $\mathcal{F}_{U=d}(P)$ as an extension sheaf code. When $U = P_{\geq \sigma}$ for some $\sigma \in P$, we use the notation $\mathcal{F}_{\sigma=d}$.

Lemma 5.5 (Efficiently Computable Extensions). *Let U, V be two upper sets in P . Let $f \in \mathcal{F}(U)$. If U is extendable to $U \cup V$, then there exists an algorithm that takes as input $f|_{P_U \cap V}$ and outputs a uniformly random element of $\mathcal{F}_{U=f}(V)$ in time $\text{poly}(|P_V|)$.*

Proof. Let B be a basis for $\mathcal{F}(V)$, let $f \in \mathcal{F}(U)$, and let $\{X_b : b \in B\}$ and $\{Y_x : x \in P_V \setminus P_U\}$ be formal variables. Consider the following system of equations.

$$S := \left\{ \sum_{b \in B} X_b \cdot b[x] = Y_x : x \in P_V \setminus P_U \right\} \cup \left\{ \sum_{b \in B} X_b \cdot b[x] = f(x) : x \in P_U \cap P_V \right\},$$

Choose uniformly a solution, $\{x_b : b \in B\} \cup \{y_x : x \in P_V \setminus P_U\}$, to S , which must exist by definition of extendability (Definition 3.10) and because U is extendable to $U \cup V$. Define $\tilde{f} \in \mathcal{F}(V)$ to be the function such that $\tilde{f}(x) = y_x$ for all $x \in P_V \setminus P_U$ and $\tilde{f}(x) = f(x)$ for all $x \in P_U \cap P_V$. Since B is a basis for $\mathcal{F}(V)$, it follows by definition of S that $\tilde{f} \in \mathcal{F}(V)$, and so by definition of extension sheaf code (Definition 5.4), $\tilde{f} \in \mathcal{F}_{U=f}(V)$. $\square$

Definition 5.6 (Randomized Encoding For Extension Sheaf Codes). *Let $\sigma \in P$. Denote by ENC_σ the distribution that takes a codeword $c \in \mathcal{F}_\sigma$ as input and outputs a uniformly sampled codeword in $\mathcal{F}_{\sigma=c}(P)$.*

Theorem 5.7. *Let $\sigma \in P$, let $q \in \mathbb{N}$ and $S \subseteq P$ such that for all $c \in \mathcal{F}_\sigma$, S is a q -extendable cover for $\mathcal{F}_{\sigma=c}$. Let $l_S, t_S : \mathbb{N} \rightarrow \mathbb{N}$ be functions satisfying the following.*

$$l_S(k) \leq k \cdot \max_{\tau \in S} |P_\tau \cap P_\sigma| \quad t_S(k) \in \text{poly}(k \cdot \max_{\tau \in S} |P_\tau|)$$

Then (ENC_σ, S) is (t_S, l_S, q) -locally simulatable.

Proof. We build a simulator for ENC_σ . The simulator has oracle access to $c \in \mathcal{F}_\sigma$, and is additionally given $S, \rho \in S$, subset $Q \subseteq S$ such that $|Q| < q$, and function $f : P_{U_Q} \rightarrow \mathbb{F}$.

Construction 5.8 ($\text{Sim}^c(f, Q, \rho, S)$).

- If $\rho \in Q$, output $f|_\rho$.
- Fix $A := U_Q$ and $A^* := A \cup P_{\geq \rho}$.
- Uniformly sample $\tilde{f} \leftarrow \mathcal{F}_{\sigma=c}(A^*)$ such that $\tilde{f}|_{P_A} = f|_{P_A}$.
- Set $f := \tilde{f}$. Output $\tilde{f}|_{P_\rho}$.

We prove that the output of the simulator is identically distributed to the output of the randomized encoding.

Lemma 5.9. Let ρ, f, Q, S be the input to the simulator. Fix $A := U_Q$ and $A^* = A \cup P_{\geq \rho}$. Let $c \in \mathcal{F}_\sigma$ be the input to the randomized encoding. Then,

$$\Pr_{\pi \leftarrow \text{Sim}^c(f, Q, \rho)}[\pi = \pi^*] = \Pr_{\tilde{f} \leftarrow \text{ENC}_\sigma(c)}[\tilde{f}|_{P_\rho} = \pi^* \mid \tilde{f}|_{P_A} = f|_{P_A}], \quad (7)$$

where $\pi^* \in \mathbb{F}^{P_\rho}$.

Proof. We start with the following claim.

Claim 5.10. A^* is globally extendable with respect to $\mathcal{F}_{\sigma=c}$.

Proof. Let $Q' := Q \cup \{\rho\}$. By definition, $A^* = U_{Q'}$. Since $Q' \subseteq S$ and $|Q'| \leq q$, and since by assumption, S is a q -extendable cover for $\mathcal{F}_{\sigma=c}$, it follows by Definition 5.3 that A^* is globally extendable with respect to $\mathcal{F}_{\sigma=c}$. $\square$

Using this fact, we have the following.

$$\begin{aligned} & \Pr_{\pi \leftarrow \text{Sim}^p(f, Q, \rho)}[\pi = \pi^*] \\ &= \Pr_{\tilde{f} \leftarrow \mathcal{F}_{\sigma=c}(A^*)}[\tilde{f}|_{P_\rho} = \pi^* \mid \tilde{f}|_{P_A} = f|_{P_A}] \quad (\text{By definition of simulator}) \\ &= \Pr_{\tilde{f} \leftarrow \mathcal{F}_{\sigma=c}(P)|_{P_{A^*}}}[\tilde{f}|_{P_\rho} = \pi^* \mid \tilde{f}|_{P_A} = f|_{P_A}] \quad (\text{By Claim 5.10 and definition of extendability}) \\ &= \Pr_{\tilde{f} \leftarrow \text{ENC}_\sigma(c)}[\tilde{f}|_{P_\rho} = \pi^* \mid \tilde{f}|_{P_A} = f|_{P_A}] \quad (\text{By definition of ENC}_\sigma) \end{aligned}$$

$\square$

Next, we measure the simulator runtime and number of queries the simulator makes to the input for each query.

Lemma 5.11. Let ρ, f, Q be input to the simulator. The simulator makes

$$|P_\rho \cap P_\sigma|$$

queries to input c and runs in time $\text{poly}(|P_\rho|)$.

Proof. Let $U := P_{\geq \sigma} \cup A$ and define $g \in \mathcal{F}(U)$ to be the codeword satisfying $g|_{P_A} = f$ and $g|_{P_\sigma} = c$. We prove the following fact.

Fact 5.12.

$$\tilde{f} \in \mathcal{F}_{U=g}(A^*).$$

Proof. By definition of $\tilde{f}$, $\tilde{f}$ is an extension of f with respect to $\mathcal{F}_{\sigma=c}$. Therefore it is in the set $(\mathcal{F}_{\sigma=c})_{A=f}(A^*)$. By definition of extension sheaf code, we have the following.

$$\begin{aligned} & (\mathcal{F}_{\sigma=c})_{A=f}(A^*) \\ &= \{d \in \mathcal{F}_{\sigma=c}(A^*) : d|_{P_A \cap P_{A^*}} = f|_{P_A \cap P_{A^*}}\} \\ &= \{d \in \mathcal{F}(A^*) : d|_{P_\sigma \cap P_{A^*}} = c|_{P_\sigma \cap P_{A^*}} \wedge d|_{P_A \cap P_{A^*}} = f|_{P_A \cap P_{A^*}}\} \\ &= \{d \in \mathcal{F}(A^*) : d|_{P_U \cap P_{A^*}} = g|_{P_U \cap P_{A^*}}\} \\ &= \mathcal{F}_{U=g}(A^*) \end{aligned}$$

$\square$

By Lemma 5.5 and Fact 5.12, $\tilde{f}$ can be computed in time $\text{poly}(|P_{A^*}|)$ with input $g|_{U \cap P_{A^*}}$. $U \cap P_{A^*} = P_{\geq \sigma} \cap P_{A^*} \cup A \cap A^*$. $g|_{A \cap A^*}$ was already computed. So the simulator needs to find $g|_{P_{\geq \sigma} \cap (P_{A^*} \setminus P_A)} = g|_{P_{\geq \sigma} \cap P_{\geq \rho}} = c|_{P_{\geq \sigma} \cap P_{\geq \rho}}$. This completes the proof. $\square$

$\square$

5.3 Proving κ -Extendability in Sheaf Codes

In the previous subsection, we observed that κ -extendability in extension sheaf codes implies a locally simulatable encoding. In this section, we provide the tools needed to later prove that covers of explicit extension sheaf codes are κ -extendable. We start by relating κ -extendability of extension sheaf codes to κ -extendability of regular (non extension) sheaf codes. Then, we provide various lemmas that show how to reduce one extendability problem to a simpler extendability problem. These facts will later be used to prove κ -extendability of different sheaf codes.

5.3.1 Extendability in Extension Sheaf Codes

In the following two lemmas, we related this property to κ -extendability in (non-extension) sheaf codes so that going forward we can focus on the more natural setting of general sheaf codes.

Lemma 5.13. *Let $V \subseteq P$. Suppose that for all $d \in \mathcal{F}(V)$, $U \subseteq P$ is globally extendable with respect to $\mathcal{F}_{V=d}$. Then U is globally extendable with respect to $\mathcal{F}$.*

Proof. Let $f \in \mathcal{F}(U)$ and let $d \in \mathcal{F}(V)$ such that $d|_{P_V \cap P_U} = f|_{P_V \cap P_U}$. By assumption of the lemma, U is globally extendable with respect to $\mathcal{F}_{V=d}$. Let $\tilde{f}$ be the extension of f in $\mathcal{F}_{V=d}(P)$. Then $\tilde{f} \in \mathcal{F}(P)$ and $\tilde{f}|_{P_U} = f \in \mathcal{F}(U)$ by definition of extension. This completes the proof. $\square$

Lemma 5.14. *Let $V \subseteq P$, $d \in \mathcal{F}(V)$. Let U be an upper set in P . If $U \cup V$ is globally extendable with respect to $\mathcal{F}$ then it is globally extendable with respect to $\mathcal{F}_{V=d}$.*

Proof. Let $f \in \mathcal{F}_{V=d}(U \cup V)$. Then $f \in \mathcal{F}(U \cup V)$ and $f|_{P_V} = d$. Since $U \cup V$ is globally extendable with respect to $\mathcal{F}$, there exists a codeword $f^* \in \mathcal{F}(P)$ such that $f^*|_{P_{U \cup V}} = f$. Therefore $f^*|_{P_V} = f|_{P_V} = d$. Therefore $f^* \in \mathcal{F}_{V=d}(P)$, which completes the proof. $\square$

5.3.2 Basic Facts About Extendability

We now turn our attention to proving extendability in (non-extension) sheaf codes. The following lemma relates unique extendability between U_1 and U_2 to the dimensions of $\mathcal{F}(U_1), \mathcal{F}(U_2)$.

Lemma 5.15. *Let U_1, U_2 be two upper sets in P such that U_1 is extendable to U_2 . If $\dim(\mathcal{F}(U_1)) = \dim(\mathcal{F}(U_2))$, then U_1 is uniquely extendable to U_2 .*

Proof. Let $r : \mathcal{F}(U_2) \rightarrow \mathcal{F}(U_1)$ be the restriction map defined as $r(f) = f|_{P_{U_1}}$. Since U_1 is extendable in U_2 it follows by definition of extendability (Definition 3.10) that r is surjective. Since $|\mathcal{F}(U_1)| = |\mathcal{F}(U_2)|$, it follows that r is also injective. By definition of injectivity, for all $f_1, f_2 \in \mathcal{F}(U_2)$, $r(f_1) = r(f_2)$ implies $f_1 = f_2$, which implies that U_1 is uniquely extendable to U_2 . $\square$

In the following lemma, we show that if $U \cap V$ is extendable to V then U is extendable to $U \cup V$. We also show the conditions under which U is uniquely extendable.

Lemma 5.16. *Let U, V be upper sets. If $U \cap V$ is extendable to V then U is extendable to $U \cup V$. Furthermore, if $\overline{\mathcal{F}(V)}^\perp \cap \overline{\mathcal{F}(U)}^\perp = \vec{0}$, and $\dim(\mathcal{F}(V)) = |P_U \cap P_V|$, then U is uniquely extendable to $U \cup V$.*

Proof. Let $f \in \mathcal{F}(U)$. Let $\tilde{f}$ be the extension of $f|_{P_{U \cap V}}$ in $\mathcal{F}(V)$. Then we construct function $f^* \in \mathcal{F}(U \cup V)$ as follows. $f^*|_{P_U} = f$ and $f^*|_{P_V} = \tilde{f}$. First, f^* is well-defined because $f|_{P_{U \cap V}} = \tilde{f}|_{P_{U \cap V}}$ by definition of extension. Next, $f^* \in \mathcal{F}(U \cup V)$ because $f^*|_{P_U} = f \in \mathcal{F}(U)$ and $f^*|_{P_V} = \tilde{f} \in \mathcal{F}(V)$. To prove unique extendability, we have the following.

$$\begin{aligned} \dim(\mathcal{F}(U \cup V)) &= \dim(\mathcal{F}(V)) + \dim(\mathcal{F}(U)) - |P_{U \cap V}| = \dim(\mathcal{F}(V)) + \dim(\mathcal{F}(U)) - \dim(\mathcal{F}(V)) \\ &= \dim(\mathcal{F}(U)). \end{aligned}$$

The first equality follows by Corollary 3.19 because $\overline{\mathcal{F}(V)}^\perp \cap \overline{\mathcal{F}(U)}^\perp = \vec{0}$, and the second equality follows because $|P_{V \cap U}| = \dim(\mathcal{F}(V))$. By Claim 5.15, U is uniquely extendable to $V \cup U$, which completes the proof. $\square$

Finally, in the next lemma, we show that if the dimension of the restriction of $\mathcal{F}(V)$ is equal to the size of the domain of $\mathcal{F}(U)$, then U must be extendable to V .

Lemma 5.17 (Full Rank Restrictions Are Extendable). *Let $V \subseteq P$ be a upper set and let $U \subseteq V$. If $\dim(\mathcal{F}(V)|_{P_U}) = |P_U|$, then U is extendable to V .*

Proof. Since $\mathcal{F}(V)|_{P_U} \subseteq \mathbb{F}^{P_U}$, it follows by assumption on the dimensions of the two spaces that $\mathcal{F}(V)|_{P_U} = \mathbb{F}^{P_U}$. Let $f \in \mathcal{F}(U)$. By definition sheaf code, $\mathcal{F}(U) \subseteq \mathbb{F}^{P_U}$ and so $f \in \mathbb{F}^{P_U}$. Therefore $f \in \mathcal{F}(V)|_{P_U}$ and so there exists $g \in \mathcal{F}(V)$ such that $g|_{P_U} = f$, which completes the proof. $\square$

5.3.3 Extendability Reductions

In this section, we show how to reduce the extendability between two upper sets to a set of simpler extendability problems. This will be very useful when proving extendability in Sections 6 and 7.

Lemma 5.18 (Extendability To Disjoint Upper Sets). *Let P be a coded space and $\mathcal{F}$ a sheaf of codes on P . Let U_1, U_2, V be upper sets in P such that $P_{U_1} \cap P_{U_2} = \emptyset$. If V is (uniquely) extendable to both $V \cup U_1$ and $V \cup U_2$, then V is (uniquely) extendable to $V \cup U_1 \cup U_2$.*

Proof. Let $f \in \mathcal{F}(V)$. Let f_1 be an (unique) extension of f in $U_1 \cup V$ and let f_2 be an (unique) extension of f in $U_2 \cup V$. Let $f^* \in \mathbb{F}^{P_{U_1 \cup U_2 \cup V}}$ be the following vector.

$$f_{P_V}^* = f \quad f_{P_{U_1 \setminus V}}^* = f_1|_{P_{U_1 \setminus V}} \quad f_{P_{U_2 \setminus V}}^* = f_2|_{P_{U_2 \setminus V}}$$

f^* is well defined in terms of f_1, f_2 since $V, U_1 \setminus V$ and $U_2 \setminus V$ are pairwise disjoint. We show that $f^* \in \mathcal{F}(V \cup U_1 \cup U_2)$. By definition of f^* , $f^*|_{P_V} \in \mathcal{F}(V)$. Next, we prove the following.

Fact 5.19.

$$f^*|_{P_{U_1}} = f_1|_{P_{U_1}}.$$

Proof. $f^*|_{P_{V \cap U_1}} = f|_{P_{V \cap U_1}} = f_1|_{P_{V \cap U_1}}$, where the first equality follows by definition of f^* and the second equality follows because $V \cap U_1 \subseteq V \cup U_1$ and f_1 is an extension of f in $U_1 \cup V$. Also by definition of f^* , $f^*|_{P_{U_1 \setminus V}} = f_1|_{P_{U_1 \setminus V}}$. Therefore f^* agrees with f_1 on the upper set $U_1 \cap V$ as well as $U_1 \setminus V$ and so $f^*|_{P_{U_1}} = f_1|_{P_{U_1}}$. $\square$

Next, we prove the analogous statement for $f^*|_{P_{U_2}}$.

Fact 5.20.

$$f^*|_{P_{U_2}} = f_2|_{P_{U_2}}.$$

Proof. Similar to the argument in the previous fact, $f^*|_{P_{V \cap U_2}} = f|_{P_{V \cap U_2}} = f_2|_{P_{V \cap U_2}}$ and $f^*|_{P_{U_2 \setminus V}} = f_2|_{P_{U_2 \setminus V}}$. Therefore, $f^*|_{P_{U_2}} = f_2|_{P_{U_2}}$. $\square$

Since $f^*|_{P_{U_1}} = f_1|_{P_{U_1}} \in \mathcal{F}(U_1)$ and $f^*|_{P_{U_2}} = f_2|_{P_{U_2}} \in \mathcal{F}(U_2)$, and since by definition of f^* , $f^*|_{P_V} \in \mathcal{F}(V)$, it follows that $f^* \in \mathcal{F}(V \cup U_1 \cup U_2)$. Finally, if V is uniquely extendable to both $V \cup U_1$ and $V \cup U_2$, then there is only one choice for both f_1 and f_2 and therefore f^* is the only extension of f in $U_1 \cup U_2 \cup V$, and so V is uniquely extendable to $U_1 \cup U_2 \cup V$. $\square$

In the following two corollaries, $n \in \mathbb{N}$, $V, U_0, \dots, U_{n-1}$ are distinct upper sets in P , and V is an upper set in P such that V is (uniquely) extendable to $V \cup U_i$ for all $i \in [n]$.

Corollary 5.21. *Suppose that for each $i, j \in [n]$, $P_{U_i} \cap P_{U_j} = \emptyset$. Then V is (uniquely) extendable to $V \cup \bigcup_{i \in [n]} U_i$.*

Proof. We proceed with proof by induction on n . When $n = 0$ and $n = 1$, the statement follows trivially. When $n = 2$, the statement follows by Lemma 5.18. Next, let $n \in \mathbb{N}$ and let $U^* = \bigcup_{i \in [n-1]} U_i$. Then $\bigcup_{i \in [n]} U_i = U^* \cup U_{n-1}$. Furthermore, since $U_{n-1} \cap U_i = \emptyset$ for all $i \in [n-1]$, it follows by Claim 2.1 that $U_{n-1} \cap U^* = \emptyset$. By inductive hypothesis, V is (uniquely) extendable to $V \cup U^*$ and by assumption, V is (uniquely) extendable to $V \cup U_{n-1}$. Therefore, by Lemma 5.18, V is (uniquely) extendable to $V \cup U^* \cup U_{n-1} = V \cup \bigcup_{i \in [n]} U_i$, which completes the proof. $\square$

Corollary 5.22. *Suppose that for each $i, j \in [n]$, $P_{U_i \setminus V} \cap P_{U_j \setminus V} = \emptyset$. Then V is (uniquely) extendable to $V \cup \bigcup_{i \in [n]} U_i$.*

Proof. By assumption, V is (uniquely) extendable to $V \cup U_i$ for all $i \in [n]$. Since $V \cup U_i = V \cup (U_i \setminus V)$, it follows that V is (uniquely) extendable to $V \cup (U_i \setminus V)$ for all $i \in [n]$. Therefore, by Corollary 5.21, V is (uniquely) extendable to $V \cup \bigcup_{i \in [n]} (U_i \setminus V) = \bigcup_{i \in [n]} (U_i \setminus V) \cup V = \bigcup_{i \in [n]} U_i \cup V = V \cup \bigcup_{i \in [n]} U_i$, which completes the proof. $\square$

The following lemma states that if we have three upper sets U, V, W , and we know that U is *uniquely* extendable to V and extendable to W , then V is also extendable to W .

Lemma 5.23 (triangle of extendability). *Let U, V, G be upper sets in P such that $U \subseteq V \subseteq G$. If U is uniquely extendable to V and extendable to G , then V is extendable to G .*

Proof. Let $f : \mathcal{F}(G) \rightarrow \mathcal{F}(V)$ be the restriction map of G to V and let $h : \mathcal{F}(U) \rightarrow \mathcal{F}(G)$ be any extension map of U to G . Then $f \circ h$ is an extension map of U to V . Since U is uniquely extendable to V , $f \circ h$ is a bijection by Proposition 3.11. Therefore, for each $v \in V$ there exists $u \in \mathcal{F}(U)$ such $f(h(u)) = v$. Since $h(u) \in \mathcal{F}(G)$, this completes the proof. $\square$

Lemma 5.24 (extendability is transitive). *Let $U_1, U_2, U_3 \subseteq P$ with $U_1 \subseteq U_2 \subseteq U_3$. If U_1 is (uniquely) extendable to U_2 and U_2 is (uniquely) extendable to U_3 , then U_1 is (uniquely) extendable to U_3 .*

Proof. Let $f_1 \in \mathcal{F}(U_1)$ and let $f_2 \in \mathcal{F}(U_2)$ be an extension of f_1 in $\mathcal{F}(U_2)$. Let $f_3 \in \mathcal{F}(U_3)$ be an extension of f_2 in $\mathcal{F}(U_3)$. Then $f_3|_{P_{U_1}} = (f_3|_{P_{U_2}})|_{P_{U_1}} = f_2|_{P_{U_1}} = f_1$ and so f_3 is an extension of f_1 in U_3 . $\square$

5.3.4 Extendability For Maximum Distance Separable Codes

In this subsection, we apply the results from the previous subsections to sheaf codes satisfying the maximum distance separable property 2.17. Since our the local codes of our bivariate Reed-Muller sheaf codes are Reed-Solomon codes, and therefore MDS, we will use this lemma when proving κ -extendability in Section 7.

Lemma 5.25. *Suppose that P contains a local code $\sigma \in P$ such that $\mathcal{F}_\sigma$ is a maximum distance separable code. Let $U \subseteq P$ be an upper set. If $|P_\sigma \cap P_U| \leq \dim(\mathcal{F}_\sigma)$ (resp. $= \dim(\mathcal{F}_\sigma)$), then U is extendable (resp. uniquely extendable) to $U \cup P_{\geq \sigma}$.*

Proof. Because $|P_U \cap P_\sigma| \leq \dim(\mathcal{F}_\sigma)$ it follows by Lemma 2.18 and the fact that $\mathcal{F}_\sigma$ is MDS that $\dim(\mathcal{F}_\sigma|_{P_U \cap P_\sigma}) = |P_U \cap P_\sigma|$ and so by Lemma 5.17, $P_{\geq \sigma} \cap U$ is extendable to $P_{\geq \sigma}$. Therefore by Lemma 5.16, U is extendable to $U \cup P_{\geq \sigma}$. Next, we prove that when $|P_U \cap P_\sigma| = \dim(\mathcal{F}_\sigma)$ then U is *uniquely* extendable to $U \cup P_{\geq \sigma}$. To prove this, we apply the second clause of Lemma 5.16, which requires us to prove the following claim.

Claim 5.26.

$$\overline{\mathcal{F}(U)}^\perp \cap \overline{\mathcal{F}_\sigma}^\perp = \vec{0}.$$

Proof. For each $v \in \overline{\mathcal{F}(U)}^\perp \cap \overline{\mathcal{F}_\sigma}^\perp$, the following must be satisfied. $v \in \overline{\mathcal{F}_\sigma}^\perp$ and $\text{supp}(v) \subseteq P_U \cap P_\sigma$. By definition of padded vector space (Definition 2.9), for all $v \in \overline{\mathcal{F}_\sigma}^\perp$, $|\text{supp}(v)| = |\text{supp}(v|_{P_\sigma})|$ and $v|_{P_\sigma} \in \mathcal{F}_\sigma^\perp$. By Lemma 2.19, $|\text{supp}(v|_{P_\sigma})| > \dim(\mathcal{F}_\sigma)$. This is a contradiction and completes the proof. $\square$

$\square$

6 Perfect Zero-Knowledge PCPPs From Sheaf Codes

In the previous section, we have shown that any sheaf code with a κ -extendable cover is a locally simulatable code. In this section, we construct a PCPP where the space of honest prover outputs is a sheaf code with a κ -extendable cover. This implies that the prover is a locally simulatable encoding, and therefore that the PCPP is perfect zero-knowledge.

We build up to our target sheaf code in stages. Recall in Section 4, we defined a d -testable sheaf code, which has a recursive structure that enables efficient local tests. In Subsections 6.1, we build a PCPP for d -testable sheaf codes. Finally, in Subsection 6.3, we construct a d -testable sheaf code from a class of code families. We show that for certain code families, the sheaf code also has a κ -extendable cover.

Looking ahead to Section 7, the PCPP from [BS08] can be recovered by instantiating Construction 6.21 with the class of Reed-Solomon codes in [BS08]. We identify a different class of Reed-Solomon codes, which force the output sheaf code of Construction 6.21 to have a κ -extendable cover. Combining that finding with the results of this section yields our perfect zero-knowledge PCPP for Reed-Solomon codes.

6.1 Perfect Zero-Knowledge PCPPs From d -Testable Sheaf Codes with κ -extendable covers

In this subsection, we construct PCPPs for d -testable sheaf codes. We show that if the sheaf code also has a κ -extendable cover, then the PCPP satisfies perfect zero-knowledge. The results of this section reduce the problem of PCPP construction to the task of finding sheaf codes with certain properties.

We begin by relating the definition of locally simulatable encoding to that of a perfect zero-knowledge PCPP. Recall that the local simulator in Definition 5.2 accepts elements of a coded space as queries, and simulates entire *local* codewords. In the following lemma, we build a simulator that simulates points instead of codewords by using the simulator from Definition 5.2.

Lemma 6.1 (Local simulatability implies black-box zero-knowledge PCPP). *Let $q \in \mathbb{N}$ and let $t(n)$ be a function in $\text{poly}(n, q)$. Let Π be a PCPP such that the prover of Π is a (t, l, q) -locally simulatable encoding. Then Π is black-box, l -local, perfect zero-knowledge PCPP with query bound q .*

Proof. Let $\mathcal{S}$ be the local simulator for the prover of the PCPP, which is guaranteed by Definition 5.2. We define a simulator $\mathcal{S}^*$ for the PCPP as follows. The simulator has oracle access to a message $m \in \mathcal{M}$ and is additionally given $x \in P_*$, $f : P_{U_T} \rightarrow \mathbb{F}$ for some $T \subseteq S$ which records the previous queries of the simulator.

Construction 6.2 (Simulator).

- If $x \in P_{U_T}$, then output $f(x)$.
- Otherwise, find $\rho \in S$ such that $x \in \rho$, query $\beta \leftarrow \mathcal{S}^m(\rho)$
- Let $\tilde{f} : P_{U_T \cup \{\rho\}} \rightarrow \mathbb{F}$ such that $\tilde{f}|_{P_{U_T}} = f$ and $\tilde{f}|_\rho = \beta$.
- Set $f := \tilde{f}$.
- Output $\beta(x)$

We show that the above simulator is identically distributed to the view of the PCPP. Denote by $\Pi.P$ the prover of the PCPP.

$$\begin{aligned} \Pr_{y \leftarrow S^*(f, T, x)}[y = y^*] &= \Pr_{\beta \leftarrow S(f, T, \alpha)}[\beta|_x = y^*] \\ &= \Pr_{\tilde{f} \leftarrow \Pi.P(m)}[\tilde{f}|_x = y^* \mid \tilde{f}|_{P_{U_T}} = f|_{P_{U_T}}] \end{aligned}$$

The second to last inequality follows because S is a locally simulatable encoding and the final term is the distribution of the PCPP view according to Definition 2.26. Furthermore, the simulator uses only black-box access to the verifier, tolerates up to q verifier queries, and has the same running time as S . Each query requires at most $l(n)$ input points, and runs in time $t(n)$. This completes the proof. $\square$

In the following construction, we build a PCPP using the local codeword test in Construction 4.6. The key difference in this construction is that the verifier only has oracle access to the input, which is the local code $\mathcal{F}_\sigma$. As such, the verifier is testing a virtual function, which is partly made up of the proof string and partly made up of the input elements.

Construction 6.3 (PCPP). In this construction, d is in $\mathbb{N}$, $\mathcal{F}$ is a d -testable sheaf of linear codes for some code C , P is the coded space for $\mathcal{F}$, $\sigma \in P$ such that $\mathcal{F}_\sigma = C$, and $S \subseteq P$ is a d -testable cover for $\mathcal{F}_\sigma$. Furthermore, we denote by T the test from Construction 4.6.

• **Prover:**

- *Input:* $\mathcal{F}$, $c \in \mathcal{F}_\sigma$
- *Output:* A uniformly distributed sample from $\mathcal{F}_{\sigma=c}(P)$.

• **Verifier:**

- *Input:* $f : \mathbb{F}^{P_\sigma} \rightarrow \mathbb{F}$, and explicit input P , $g : \mathbb{F}^{P^*} \rightarrow \mathbb{F}$, σ and S .
- *Output:* Let $\hat{g} : \mathbb{F}^{P^*} \rightarrow \mathbb{F}$ be the function such that $\hat{g}|_{P^* \setminus P_\sigma} = g|_{P^* \setminus P_\sigma}$ and $\hat{g}|_{P_\sigma} = f$. The verifier outputs $T(\hat{g}, \sigma, S)$.

Theorem 6.4. *If there exists $\kappa \in \mathbb{N}$ such that $\mathcal{F}_{\sigma=c}$ has a κ -extendable cover, then for all $\epsilon \in [0, 1]$, Construction 6.3 is a black-box l_S -local (where l_S is a function defined in Theorem 5.7), perfect zero-knowledge PCPP for $\mathcal{F}_\sigma$ with query complexity $\max_{\alpha \in S_1} |\alpha|$ (where S_1 is from Definition 4.3), query bound κ , soundness error $(1 - \epsilon)$, and proximity parameter $c^d \cdot \epsilon$.*

Proof. Completeness and soundness follow immediately from Theorem 4.7, and perfect zero-knowledge follows because by Theorem 5.7, and since $\mathcal{F}_{\sigma=c}$ has a κ -extendable cover, it follows that the prover is a locally simulatable encoding. Applying Lemma 6.1 completes the proof. $\square$

6.2 Constructing d -testable and κ -extendable covers through merging

In this subsection, we construct a d -testable sheaf code, which generalizes the recursive structure of the PCPP from [BS08]. We utilize the sheaf code merging (Definition 3.26) to “glue” together a set of $(d - 1)$ -testable sheaf codes, along with a 1-testable sheaf code with testable cover S . Each of the $(d - 1)$ -testable sheaf codes gives a test for one of the codes in S . We place our sheafs intentionally, so that two sheafs only intersect on codes that are isomorphic to each other, and furthermore, only on the codes in the cover S . The following construction takes as input a main sheaf which contains the code, σ , that we would like to test, and which we denote $\mathcal{F}'$; a testable cover, S for local code $\sigma \in P'$; and for each $\rho \in S$, a $d - 1$ -testable sheaf code which we denote by $\mathcal{F}[\rho]$. For each $\rho \in S$, we additionally require a sheaf code isomorphism h_ρ that maps ρ to a local code $h_\rho(\rho)$ in $\mathcal{F}[\rho]$. That way, when we run the $d - 1$ -test for $\mathcal{F}[\rho]_{h_\rho(\rho)}$, we end up testing P'_ρ as well, which gives us our final d -test.

Construction 6.5 (*d*-Testable Sheaf Codes From $d - 1$ Testable Sheaf Codes). The construction takes the following inputs: a sheaf, $\mathcal{F}'$, defined over P' , a local code domain $\sigma \in P'$, testable cover $S \subseteq P'$ for code $\mathcal{F}'_\sigma$, and a set $\{\mathcal{F}[\rho] : \rho \in S\}$ such that for each $\rho \in S$, there exists $\sigma_\rho \in P[\rho]$ such that $\mathcal{F}[\rho]$ is a $(d - 1)$ -testable sheaf code for $\mathcal{F}[\rho]_{\sigma_\rho}$ and finally, there exists a sheaf code isomorphism (Definition 3.20) $h_\rho : P'_\rho \rightarrow P[\rho]_{\sigma_\rho}$ at upper set $P'_{\geq \rho}$ where $h_\rho(\rho) := \sigma_\rho$.

- Let $\{g' : P'_* \rightarrow \mathcal{U}\} \cup \{g_\rho : P[\rho]_* \rightarrow \mathcal{U}\}$ be a set of injections such that the following holds: for all $\rho_1 \neq \rho_2 \in P[\rho]$,

$$\text{Im}(g_{\rho_1}) \cap \text{Im}(g_{\rho_2}) \subseteq \text{Im}(g'),$$

and for all $\rho \in S$,

$$\text{Im}(g_\rho) \cap \text{Im}(g') = g'(\rho) \quad h_\rho = g_\rho^{-1} \circ g'|_\rho$$

- Set $A := \{(\mathcal{F}[\rho], g_\rho) : \rho \in S\} \cup \{(\mathcal{F}', g')\}$. Output $M(A)$.

For the remainder of the section, $\mathcal{F}'$ is a sheaf code with coded space P' , $\sigma \in P'$, $S \subseteq P'$ is a testable cover for $\mathcal{F}'_\sigma$, and $\{\mathcal{F}[\rho] : \rho \in S\}$ are as specified in Construction 6.5. We refer to $\mathcal{F}$ as the corresponding output of Construction 6.5.

6.2.1 Correctness of Construction 6.5

In this paragraph, we prove that Construction 6.5 is correct and outputs a merged sheaf code. To prove this, we show that A (from the last step of the construction) satisfies the conditions required by the merge sheaf code definition (Definition 3.26)

Lemma 6.6. *For any $\mathcal{F}'$, σ , S , $\{\mathcal{F}[\rho] : \rho \in S\}$, Construction 6.5 outputs a merged sheaf, i.e. a sheaf satisfying Definition 3.26.*

Proof. We will prove that A satisfies the conditions of Definition 3.26. Let $(\hat{\mathcal{F}}_1, \hat{g}_1), (\hat{\mathcal{F}}_2, \hat{g}_2) \in A$ such that $\hat{g}_1(\sigma) \supseteq \hat{g}_2(\tau)$ for some $\sigma \in \hat{P}_1, \tau \in \hat{P}_2$. We use the following three claims, one for each choice of pair $\hat{g}_1, \hat{g}_2$. In the following claims, ρ and β are codes in S . We use several facts throughout the proof, whose proofs we defer to Section B.

Claim 6.7. *Let $\sigma \in P', \tau \in P[\rho]$. If $g_\rho(\tau) \subseteq g'(\sigma)$ then $g'^{-1} \circ g_\rho|_\tau$ is a sheaf code isomorphism at upper set $P[\rho]_{\geq \tau}$.*

Proof. We make use of the following fact.

Fact 6.8.

$$\tau \geq h_\rho(\rho)$$

Since h_ρ is a sheaf code isomorphism at $P'_{\geq \rho}$, it follows by Proposition 3.21 that h_ρ^{-1} is a sheaf code isomorphism at upper set $P[\rho]_{\geq h_\rho(\rho)}$. By Fact 6.8, $\tau \geq h_\rho(\rho)$, and so by Proposition 3.22, $h_\rho^{-1}|_\tau$ is a sheaf code isomorphism at $P[\rho]_{\geq \tau}$. By definition of h_ρ, g' , and g_ρ , $h_\rho^{-1}|_\tau = g'^{-1} \circ g_\rho|_\tau$, which completes the proof. $\square$

Claim 6.9. *Let $\sigma \in P[\rho], \tau \in P'$. If $g'(\tau) \subseteq g_\rho(\sigma)$ then $g_\rho^{-1} \circ g'|_\tau$ is a sheaf code isomorphism at upper set $P'_{\geq \tau}$.*

Proof of Claim 6.9. We make use of the following fact.

Fact 6.10.

$$\tau \geq \rho$$

Since h_ρ is a sheaf code isomorphism at upper set $P'_{\geq \rho}$ and since by Fact 6.10, $\tau \geq \rho$, it follows that $h_\rho|_\tau$ is a sheaf code isomorphism at $P'_{\geq \tau}$. Since $h_\rho|_\tau = g_\rho^{-1} \circ g'|_\tau$, this completes the proof. $\square$

Claim 6.11. *Let $\sigma \in P[\rho], \tau \in P[\beta]$. If $g_\beta(\tau) \subseteq g_\rho(\sigma)$ for some $\sigma \in P[\rho], \tau \in P[\beta]$ then $g_\rho^{-1} \circ g_\beta|_\tau$ is a sheaf code isomorphism at upper set $P[\beta]_{\geq \tau}$.*

Proof of Claim 6.11. We prove the following two facts.

Fact 6.12.

$$h_\beta^{-1}(P[\beta]_{\geq \tau}) \subseteq P'_{\geq \rho}.$$

Fact 6.13.

$$\tau \geq h_\rho(\rho)$$

By definition, h_ρ is a sheaf code isomorphism at upper set $P'_{\geq \rho}$ and h_β^{-1} is one at upper set $P[\beta]_{\geq h_\rho(\rho)}$. By Fact 6.13, $\tau \geq h_\rho(\rho)$ and therefore by Proposition 3.22, $h_\beta^{-1}|_\tau$ is a sheaf code isomorphism at $P[\beta]_{\geq \tau}$. By Fact 6.12, $h_\beta^{-1}(P[\beta]_{\geq \tau}) \subseteq P'_{\geq \rho}$. Therefore, by Proposition 3.23, $h_\rho \circ h_\beta^{-1}|_\tau$ is a sheaf code isomorphism at upper set $P[\beta]_{\geq \tau}$. $h_\rho \circ h_\beta^{-1}|_\tau = g_\rho^{-1} \circ g'|_\tau \circ g_\beta^{-1} \circ g_\beta|_\tau = g_\rho^{-1} \circ g' \circ g_\beta^{-1} \circ g_\beta|_\tau = g_\rho^{-1} \circ g_\beta|_\tau$, where the second to last equality follows because $\Im(h_\beta^{-1}|_\tau) \subseteq P'_\rho$. This completes the proof. $\square$

$\square$

6.2.2 Testability of Construction 6.5

Lemma 6.14. *$\mathcal{F}$ is a d -testable sheaf code for $\mathcal{F}_{g'(\sigma)}$.*

Proof. By Construction 6.5, $S \subseteq P'$ is a 1-testable cover for $\mathcal{F}'_\sigma$. By Proposition 3.29, g' is a sheaf code isomorphism between $\mathcal{F}'$ and $\mathcal{F}$ at upper set P' . Therefore, by Lemma 4.4, $g'(S)$ is a 1-testable cover for $g'(\mathcal{F}'_\sigma) = \mathcal{F}_{g'(\sigma)}$. Next, let $\rho \in g'(S)$. We show that $\mathcal{F}$ contains a $d - 1$ testable cover for $\mathcal{F}_\rho$. Set $\alpha := g'^{-1}(\rho) \in S$. Then, $\mathcal{F}[\alpha]$ contains a $d - 1$ testable cover for $\mathcal{F}[\alpha]_{h_\rho(\rho)}$. Therefore, by Proposition 3.29 and Corollary 4.5, $g_\alpha(\mathcal{F}[\alpha])$ contains a $d - 1$ testable cover for $g_\alpha(\mathcal{F}[\alpha]_{h_\rho(\rho)})$.

$$g_\alpha(\mathcal{F}[\alpha]_{h_\rho(\rho)}) = \mathcal{F}_{g_\alpha(h_\rho(\rho))} = \mathcal{F}_{g'(\alpha)} = \mathcal{F}_\rho.$$

The first equality follows by Lemma 6.6 and by definition of merged sheaf (Definition 3.26), the second equality follows by definition of α and g' . By Proposition 3.29, g_α is a sheaf code isomorphism between $\mathcal{F}[\alpha]$ and $\mathcal{F}$, and therefore $g_\alpha(\mathcal{F}[\alpha])$ is a subspace of $\mathcal{F}$, which completes the proof. $\square$

6.2.3 Extendability of Construction 6.5

We prove the following proposition, which will be useful when arguing that Construction 6.5 is κ -extendable.

Proposition 6.15. *Suppose that $P[\rho]_{\geq h_\rho(\rho)}$ is globally extendable with respect to $\mathcal{F}[\rho]$ for all $\rho \in S$. Let $U \subseteq P'$ be a globally extendable upper set with respect to $\mathcal{F}'$ and let $K \subseteq S$ be a subset. Then, $g'(U)$ is extendable to*

$$g'(P') \cup \bigcup_{\rho \in K} g_\rho(P[\rho])$$

Proof. Since U is extendable to P' with respect to $\mathcal{F}'$, it follows by Lemma 3.25 that $g'(U)$ is extendable to $g'(P')$ with respect to $\mathcal{F}$. Next, we show the following.

Claim 6.16. *For each $\rho \in S$, $g'(P')$ is extendable to $g'(P') \cup g_\rho(P[\rho])$.*

Proof. By Construction 6.5, $g'(P') \cap g_\rho(P[\rho]) = g'(P'_{\geq \rho}) = g_\rho(P[\rho]_{\geq h_\rho(\rho)})$, where both equalities follow by definition of g', g_ρ . By assumption, $P[\rho]_{\geq h_\rho(\rho)}$ is globally extendable with respect to $\mathcal{F}[\rho]$. Therefore by Lemma 3.25, $g_\rho(P[\rho]_{\geq h_\rho(\rho)})$ is extendable to $g_\rho(P[\rho])$ with respect to $g_\rho(\mathcal{F}[\rho])$. Therefore by Lemma 5.16, since $g'(P') \cap g_\rho(P[\rho])$ is extendable to $g_\rho(P[\rho])$, it follows that $g'(P')$ is extendable to $g'(P') \cup g_\rho(P[\rho])$. $\square$

Furthermore, by definition of g_ρ and g' , for each $\rho_1, \rho_2 \in S$ with $\rho_1 \neq \rho_2$,

$$(g_{\rho_1}(P[\rho_1]) \setminus g'(P')) \cap (g_{\rho_2}(P[\rho_2]) \setminus g'(P')) = \emptyset.$$

Therefore, applying Lemma 5.22, with Claim 6.16, it follows that $g'(P')$ is extendable to $g'(P') \cup \bigcup_{\rho \in K} g_\rho(P[\rho]) = g'(P') \cup \bigcup_{\rho \in K} g_\rho(P[\rho])$. Applying Lemma 5.24 (transitivity of extendability) completes the proof. $\square$

Corollary 6.17. *Suppose that $P[\rho]_{\geq h_\rho(\rho)}$ is globally extendable with respect to $\mathcal{F}[\rho]$ for all $\rho \in S$. Let $U \subseteq P'$ be a globally extendable upper set with respect to $\mathcal{F}'$. Then $g'(U)$ is globally extendable with respect to $\mathcal{F}$.*

Proof. By definition of merged sheaf and Construction 6.5, $P = \bigcup_{\rho \in S} g_\rho(P[\rho]) \cup g'(P')$. Therefore, applying Proposition 6.15 completes the proof. $\square$

The following shows that if S is a κ -extendable cover for input $\mathcal{F}'$, and if $P[\rho]_{\geq h_\rho(\rho)}$ is extendable in $\mathcal{F}[\rho]$ for each $\rho \in S$, then the output sheaf code, $\mathcal{F}$, has a κ -extendable cover, where each code in the cover consists of the codes in S , along with their $d - 1$ testable sheaf codes.

Theorem 6.18 (κ -extendable covers in $\mathcal{F}$). *Suppose that for all $c \in \mathcal{F}'_{\sigma=c}$, S is a κ -extendable cover for $\mathcal{F}'_{\sigma=c}$. Furthermore, suppose that for all $\rho \in S$, $P[\rho]_{\geq h_\rho(\rho)}$ is globally extendable with respect to $\mathcal{F}[\rho]$. Then for all $c \in C$, $\{g_\rho(P[\rho]_*) : \rho \in S\}$ is a κ -extendable cover for $\mathcal{F}_{g'(\sigma)=g'(c)}$.*

Proof. Let $K \subseteq S$ such that $|K| \leq \kappa$. Define U_K as follows.

$$U_K := \bigcup_{\rho \in K} g_\rho(P[\rho]).$$

To complete the proof, we will apply Lemma 5.14, which requires us to show that $U_K \cup P_{\geq g'(\sigma)}$ is globally extendable with respect to $\mathcal{F}$. First, by Construction 6.5, $P = U_K \cup g'(P') \cup U_{K^c}$. Let $U_K^* := U_K \cup P_{\geq g'(\sigma)}$. We prove the following.

Fact 6.19.

$$U_K^* \cap (g'(P') \cup U_{K^c}) = \bigcup_{\rho \in K} g'(P'_\rho) \cup P_{g'(\sigma)}$$

Let $V := \bigcup_{\rho \in K} P'_\rho \cup P'_\sigma$. By Definition 2.7,

$$\bigcup_{\rho \in K} g'(P'_\rho) \cup P_{g'(\sigma)} = \bigcup_{\rho \in K} g'(P'_\rho) \cup g'(P'_\sigma) = g'(\bigcup_{\rho \in K} P'_\rho \cup P'_\sigma) = g'(V)$$

By assumption of the lemma, S is a κ -extendable cover for $\mathcal{F}'_{\sigma=c}$. Therefore, since $|K| \leq \kappa$, it follows that V is globally extendable with respect to $\mathcal{F}'$. Therefore, by Proposition 6.15, $g'(V)$ is extendable to $g'(P') \cup U_{K^c}$. Therefore, by Lemma 5.16, U_K^* is extendable to $U_K^* \cup g'(P') \cup U_{K^c}$. Since $U_K \cup U_{K^c}$ contains all codes greater than or equal to $\{g_\rho(P[\rho])\}$ for each $\rho \in S$ it follows that $U_K^* \cup g'(P') \cup U_{K^c} = P$. This completes the proof. $\square$

6.3 d -Testable Sheaf Codes with κ -Extendable Covers From Recursively Testable Codes

In the following construction, we build a d -testable sheaf code via recursive calls to Construction 6.5, where the inputs to each call is dictated by the next code in a “recursively testable code family” (Definition 6.20). This sets us up for our ultimate goal: an explicit perfect zero-knowledge PCPP, which we will obtain by instantiating Construction 6.21 with a well-chosen family of Reed-Solomon codes.

Definition 6.20 (Recursively testable code family). *Let $n \in \mathbb{N}$. A code family (Definition 2.12), $\{C_i\}_{i \geq 1}$, is recursively testable if for all $i \geq 1$, there exists a testable sheaf, $\mathcal{F}'[i]$, for C_i with testable cover S_i such that for all $i \in [n]$ and $\rho \in S_i$, $\mathcal{F}'[i]_\rho = g(C_{j(\rho)})$ for some $j(\rho) < i$ and some permutation mapping g (Definition 2.7). Furthermore, for $\sigma_\rho \in P'[j(\rho)]$ such that $C_{j(\rho)} = \mathcal{F}'[j(\rho)]_{\sigma_\rho}$, $\mathcal{F}'[j(\rho)]_{\sigma_\rho}$ and $\mathcal{F}'[i]_\rho$ are isomorphic as sheaf codes.*

Construction 6.21. Let $\{C_i\}_{i \geq 1}$ be a recursive testable code family, with associated sheafs $\{\mathcal{F}'[i] : i \geq 1\}$. Let $n, d \geq 1$. We construct a d -testable sheaf code for C_n , which we denote by $\mathcal{F}_d[n]$, as follows.

- $\mathcal{F}_1[n] = \mathcal{F}'[n]$
- For $d > 1$, for each $\rho \in S$, let $h_\rho : P'[n]_\rho \rightarrow P'[j(\rho)]_{\sigma_\rho}$ be a sheaf code isomorphism between $\mathcal{F}'[n]$ and $\mathcal{F}'[j(\rho)]$ at upper set $P'[n]_{\geq \rho}$ for some $\sigma_\rho \in P'[j(\rho)]$ where $\mathcal{F}'[j(\rho)]_{\sigma_\rho} = C_{j(\rho)}$. Let $\sigma \in P'[n]$ such that $\mathcal{F}'[n]_\sigma = C_n$. Then, $\mathcal{F}_d[n]$ is the output of Construction 6.5 on inputs $\sigma, \mathcal{F}'[n], S_n$, and

$$\{(\mathcal{F}_{d-1}[j(\rho)], h_\rho) : \rho \in S_n\}$$

Lemma 6.22 (Construction 6.21 is Correct). *The output of Construction 6.21 is well defined.*

Proof. Proof by induction. When $d := 1$, the output is just $\mathcal{F}'[n]$. Next, let $d > 1$. By construction, h_ρ is a sheaf code isomorphism at $P'[n]_{\geq \rho}$ and by inductive hypothesis and Construction 6.5, there is a sheaf code isomorphism $g : P'[j(\rho)]_{\sigma_\rho} \rightarrow P_{d-1}[j(\rho)]_{g'(d-1, j(\rho))(\sigma_\rho)}$. Therefore, it follows by Proposition 3.23 (composition of sheaf isomorphisms) that there is a sheaf code isomorphism between $\mathcal{F}'[n]$ and $\mathcal{F}_{d-1}[j(\rho)]$ on $P'[n]_{\geq \rho}$. $\square$

For the remainder of the section, let $g'[d, n]$ be the sheaf code isomorphism between $\mathcal{F}'[n]$ and $\mathcal{F}_d[n]$ on upper set $P'[n]$ as defined in Construction 6.5.

Proposition 6.23. *Let $n, d \geq 1$. If for all $j \leq n$, $P'[j]_{\geq \alpha_j}$ is globally extendable with respect to $\mathcal{F}'[j]$, then $P_d[n]_{\geq g'[d, n](\alpha_n)}$ is globally extendable with respect to $\mathcal{F}_d[n]$.*

Proof. Proof by induction. When $d = 1$, $\mathcal{F}'[n] = \mathcal{F}_d[n]$ and the proof follows trivially. Next, when $d > 1$, since $j_\rho < n$, we can apply the inductive hypothesis and conclude that for all $\rho \in S_n$, $g'[d-1, j_\rho](P'[j_\rho]_{\geq \alpha_{j_\rho}})$ is globally extendable with respect to $\mathcal{F}_{d-1}[j_\rho]$. Therefore, by Proposition 6.15, since $P'[n]_{\geq \alpha_n}$ is globally extendable with respect to $\mathcal{F}'[n]$, it follows that $g'[d, n](P'[n]_{\geq \alpha_n}) = P_d[n]_{\geq g'[d, n](\alpha_n)}$ is globally extendable with respect to $\mathcal{F}_d[n]$, which completes the proof. $\square$

Lemma 6.24 ($\mathcal{F}_{d, n}$ Contains A d -Testable Cover For C_n). *Let $d, n \in \mathbb{N}$ and let $\{C_i\}_{i \geq 1}$ be a recursively testable code family. Then for every integer $n \geq 1$, $\mathcal{F}_d[n]$ contains a d -testable cover for C_n .*

Proof. By Lemma 6.22, $\mathcal{F}_d[n]$ is an output of Construction 6.5. Therefore the proof follows directly from Lemma 6.14 and by Construction 6.21. $\square$

Lemma 6.25 (κ -extendable covers for $\mathcal{F}_{d, i}$). *Let $\kappa, d, n \in \mathbb{N}$ and let $\{C_i\}_{i \geq 1}$ be a recursively testable code family. For all $n \geq 1$, if for all $c \in C_n$, S_n is a κ extendable cover for $(\mathcal{F}'[n])_{\alpha_n=c}$, then $(\mathcal{F}_d[n])_{g'(\alpha_n)=g'(c)}$ has a κ -extendable cover.*

Proof. Fix $g := g'[d, n]$. When $d = 1$, the statement follows trivially. Otherwise, we apply Theorem 6.18, which requires us to prove that for all $c \in C_n$, S is a κ extendable cover in $(\mathcal{F}'[n])_{\alpha_n=c}$ (which we know by assumption of the lemma), and that for each $\rho \in S$, $P_{d-1}[j_\rho]_{\geq g(\alpha_{j_\rho})}$ is globally extendable with respect to $\mathcal{F}_{d-1}[j_\rho]$. This follows from Proposition 6.23. $\square$

7 Short, Perfect Zero-Knowledge PCPP For Reed-Solomon Codes

7.1 A Useful Generalization Of The BS08 Bivariate Reed-Muller Code

In this section, $\mathbb{F} = \mathbb{F}_{2^l}$ is a binary field, which we can view as a vector space over $\mathbb{F}_2$ of dimension l . For $L \subseteq \mathbb{F}$, denote by q_L the polynomial of minimal degree that vanishes on L , i.e.,

$$q_L(z) := \prod_{\alpha \in L} (z - \alpha).$$

Definition 7.1. Let $k, \hat{k} \in \mathbb{N}$, such that $k < l$, $k/2 + \hat{k} + 2 \leq l$. For a subspace $L \subseteq \mathbb{F}$ of dimension k , fix a canonical basis $b_1, \dots, b_k$ of L , and define

$$D_{\hat{k}}(L) := \bigcup_{\beta \in L_1} (L_\beta \times \{q_{L_0}(\beta)\}) \subseteq L \times q_{L_0}(L_1)$$

where L_β, L_1, L_0 are defined as follows:

$$\begin{aligned} L_0 &:= \text{span}(b_1, \dots, b_{k/2}) & L_1 &:= \text{span}(b_{k/2+1}, \dots, b_k) \setminus L_R \\ L_R &:= \text{span}(b_{k/2+1}, \dots, b_{k/2+\hat{k}+2}) & L_\beta &= \text{span}(L'_0, \beta) \quad \forall \beta \in L_1 \\ L'_0 &:= L_0 + L_R \end{aligned}$$

Define the coded space $P_{\hat{k}}(L) := D_{\hat{k}}(L) \cup \mathcal{R} \cup \mathcal{K} \cup \{T\}$, consisting of

- rows $\mathcal{R} := \{R_x : x \in L\}$ where $R_x = \{x\} \times \{y : (x, y) \in D_{\hat{k}}(L)\}$;
- columns $\mathcal{K} := \{K_y : y \in q_{L_0}(L_1)\}$, where $K_y = \{x : (x, y) \in D_{\hat{k}}(L)\} \times \{y\}$;
- a curve $T := \{(x, q_{L_0}(x)) : x \in L \setminus L_0\}$.

Finally, for degrees $d_X, d_Y \in \mathbb{N}$ where d_X is a power of two, define the sheaf of linear codes $\mathcal{F} = \mathcal{F}_{d_X, d_Y}(L)$ on $P_{\hat{k}}(L)$ where $\hat{k} := \log_2(d_X + 1) - k/2$:

- for $p \in D_{\hat{k}}(L)$, $\mathcal{F}_p = \mathbb{F}$;
- for $R_x \in \mathcal{R}$, $\mathcal{F}_{R_x} = \text{RS}[\mathbb{F}, R_x, d_X]$;
- for $K_y \in \mathcal{K}$, $\mathcal{F}_{K_y} = \text{RS}[\mathbb{F}, K_y, d_Y]$;
- and $\mathcal{F}_T = \{c \in \mathbb{F}^T : (c(x, q_{L_0}(x)))_{x \in L \setminus L'_0} \in \text{RS}[\mathbb{F}, L \setminus L'_0, d_X + d_Y \cdot |L_0|]\}$.

Remark 7.2. When $\hat{k} = 0$ we recover the bivariate Reed-Muller code from [BS08].

Figure 1 depicts the domain $D_{\hat{k}}(L)$ in orange, with the subspace curve highlighted in blue.

7.2 κ -Extendable Covers In Bivariate Reed-Muller Sheaf Codes

Theorem 7.3. Let $L \subseteq \mathbb{F}$ be a subspace, let $d_X, d_Y \in \mathbb{N}$ such that $d_X \geq |L_0| + d_Y$, $d_Y \geq 1$, and let $\mathcal{F} := \mathcal{F}_{d_X, d_Y}(L)$ be a bivariate Reed-Muller sheaf code (Definition 7.1). For all $c \in \mathcal{F}_T$, $\mathcal{R} \cup \mathcal{K}$ is a d_Y -extendable cover for $\mathcal{F}_{T=c}$.

We prove Theorem 7.3 using results about extendability from Section 5.3. To build intuition, we start with a high-level overview that uses only elementary facts about bivariate polynomials. Then, we describe the same argument through the more general lens of upper set extendability, and continue with the formal proof. Figures 2 depicts the former viewpoint and Figure 3 depicts the latter.

Overview of the proof of Theorem 7.3 in the language of bivariate polynomials. Let $S \subseteq \mathcal{R} \cup \mathcal{K}$ be a collection of rows and columns in P such that $|S| < d_Y$. Let $S' := S \cup \{T\}$. Let $D_{S'} = U_{S'}^* \subseteq \mathbb{F} \times \mathbb{F}$ be the maximal set of $U_{S'}$, i.e. the union of the domains of the local codes in S' . We need to prove that $U_{S'}$ is globally extendable with respect to $\mathcal{F}$. Since $\mathcal{F}(P)$ is a bivariate Reed-Muller sheaf code

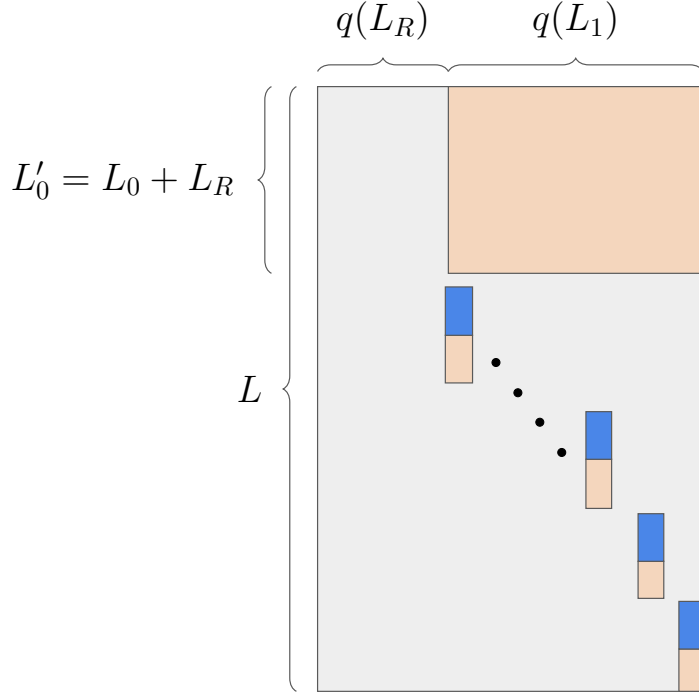


Figure 1: The domain $D_k(L)$, with the subspace curve highlighted in blue.

(Definition 7.1), this translates to proving that for any $f \in \mathcal{F}(U_{S'})$, there exists a bivariate polynomial $Q \in \mathbb{F}^{\leq [d_X, d_Y]}[X, Y]$ such that $Q(x, y) = f(x, y)$ for all $(x, y) \in D_{S'}$.

We start by considering some collection $W \subseteq P$ such that $S \subseteq W$, where W consists of exactly $d_X + 1$ rows, d_Y columns, and the curve T . Let $D_W = P_{U_W}$ be the maximal elements of U_W . Let R' be the rows and let K' be the columns in W . Let $D_{R'} = P_{U_{R'}}$, $D_{K'} = P_{U_{K'}}$ be the maximal elements of upper sets $U_{R'}$ and $U_{K'}$ respectively, and let $f \in \mathcal{F}(U_{R'})$. First, we find a bivariate polynomial $Q \in \mathbb{F}^{\leq [d_X, d_Y]}[X, Y]$ satisfying $Q|_{D_{R'}} = f|_{D_{R'}}$, which is proven to exist in Lemma H.1 in [BCFGRS17]. We will argue next that $Q|_{D_W} = f|_{D_W}$. We do this in two steps. First, $Q|_{D_{K'}}$ is equal to $f|_{D_{K'}}$ because for each column $\rho \in K$, $Q|_{\rho}$ and $f|_{\rho}$ are univariate polynomials of degree $\leq d_X$ and $Q|_{\rho}$ agrees with $f|_{\rho}$ on each of the $d_X + 1$ points in $P_{\rho} \cap D_{R'}$. By a similar argument, $Q|_T$ is equal to $f|_T$ because $f|_T$ and $Q|_T$ are both univariate polynomials of degree $\leq \Delta = d_X + d_Y \cdot |L_0|$ and $Q|_T$ agrees with $f|_T$ on each of the $\Delta + 1$ points in $T \cap (D_{K'} \cup D_{R'})$. Putting these two facts together, Q agrees with f on all of D_W .

Finally, we are ready to reason about the global extendability of $U_{S'}$. Let $f \in \mathcal{F}(U_{S'})$. Fix a codeword $g \in \mathcal{F}(W)$ such that $g|_{D_{S'}} = f|_{D_{S'}}$. We use the argument from the previous paragraph to find a bivariate polynomial $Q \in \mathbb{F}^{\leq [d_X, d_Y]}[X, Y]$ such that $Q|_{D_W} = g|_{D_W}$. Since $D_{S'} \subseteq D_W$, it follows that $Q|_{D_{S'}} = g|_{D_{S'}}$, which by definition is equal to $f|_{D_{S'}}$. This completes the proof.

Proof of Theorem 7.3. We are now ready for the formal proof of Theorem 7.3. Using the notation of the previous subsection, we start by proving that U_W is globally extendable. We observe that the above argument from bivariate polynomial interpolation is an instance of more abstract extendability properties as established by Lemma 5.23 (“triangle of extendability”) and Lemma 5.24 (“transitivity of extendability”). Lemma 5.23 tells us that if $U_{R'}$ is uniquely extendable to U_W and is itself globally extendable (which Lemma H.1 of [BCFGRS17] proves it is), then U_{T^*} must also be globally extendable. Similarly, Lemma 5.24 tells us that if U_W is globally extendable, and $U_{S'}$ is extendable to U_W , then $U_{S'}$ is globally extendable. Finally, we can prove the extendability of $U_{R'}$ and $U_{S'}$ to U_W by applying Lemmas 5.25 and 5.18 and then counting intersections of the different local domains.

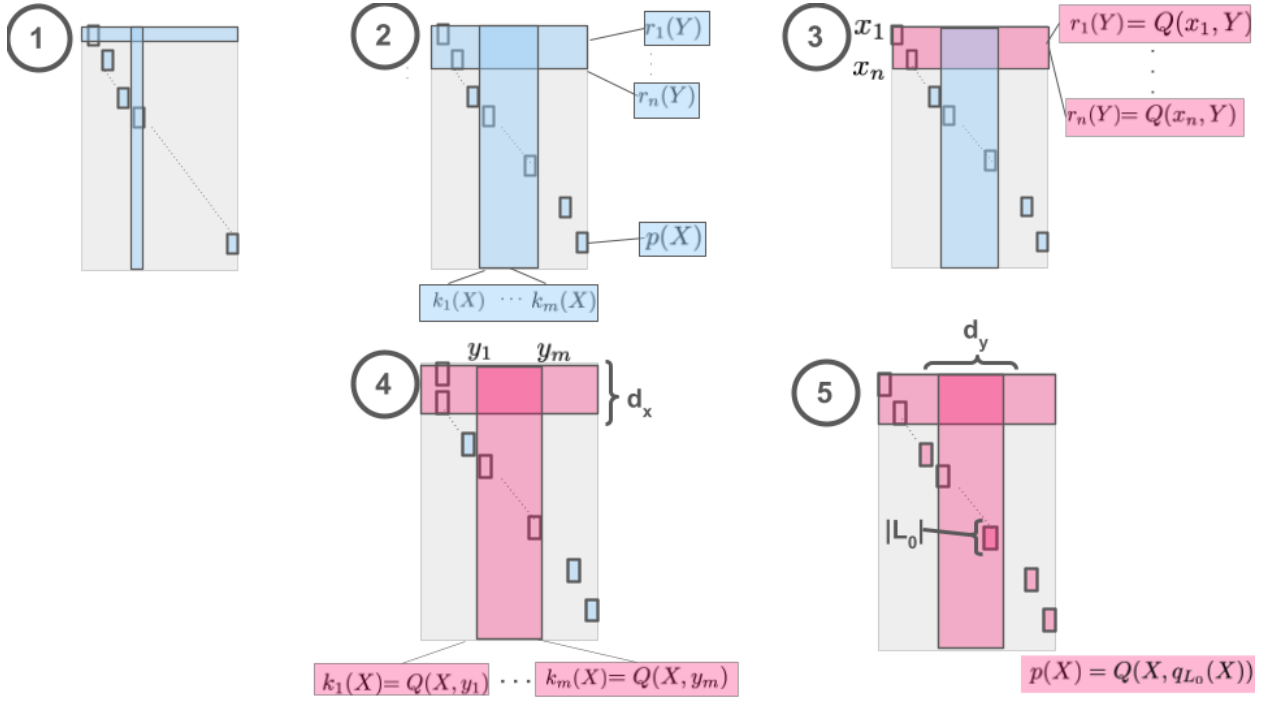


Figure 2: This image depicts the proof strategy described in Section 7.2, which uses polynomial interpolation. In Frame (1), we are given a function on a small domain consisting of one row, one column and the subspace curve. The function consists of a d_X univariate polynomial on the column, a d_Y univariate polynomial on the row, and a degree- Δ univariate polynomial on T . In Frame (2), we extend this domain, and assign to each new column a univariate polynomials of degree $\leq d_X$ and to each new row a univariate polynomials of degree $\leq d_Y$. In Frame (3), we find $Q \in \mathbb{F}^{\leq [d_X, d_Y]}[X, Y]$ that agrees with evaluations on the rows. Because each column is fully determined by the values in the rows, the columns must also agree with Q , which is depicted in Frame (4). Then, because there are $\Delta + 1$ points in T which are determined by the rows and columns in Frame (4), the degree- Δ polynomial on T must also agree with Q , which is depicted in Frame (5).

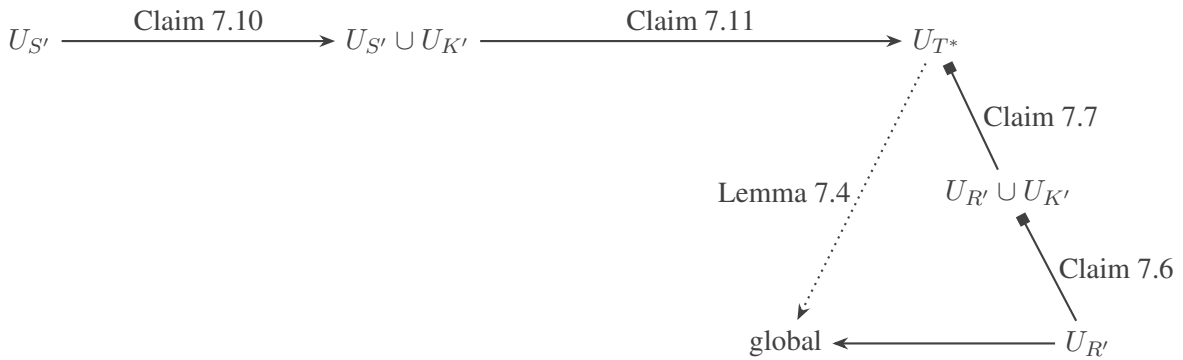


Figure 3: A diagram depicting the proof strategy of Theorem 7.3 using results from Section 5.3. Standard arrows denote extendability and square arrows denote unique extendability. $U_{S'}$ corresponds to Frame (1) in Figure 2. U_{T^*} refers to Frame (2) and the triangle describes the argument outlined by Frames (3), (4), and (5) in Figure 2.

Proof. Let $S \subseteq \mathcal{R} \cup \mathcal{K}$, $|S| \leq d_Y$, and let $S' := S \cup \{T\}$. To prove the theorem, it suffices by Lemma 5.14 to prove that $U_{S'}$ is globally extendable (Definition 3.10) with respect to $\mathcal{F}$. Let $R := S \cap \mathcal{R}$, $K := S \cap \mathcal{K}$ be the rows and columns in S , respectively. Choose any sets $R' \subseteq \mathcal{R}$, $K' \subseteq \mathcal{K}$ such that $R \subseteq R'$, $|R'| = d_X + 1$, $K \subseteq K'$, $|K'| = d_Y$. Let $W := K' \cup R' \cup T$. We prove the following two lemmas.

Lemma 7.4. U_W is globally extendable.

Lemma 7.5. $U_{S'}$ is extendable to U_W .

By transitivity of extendability (Lemma 5.24), Lemmas 7.4 and 7.5 together imply that $U_{S'}$ is globally extendable, which establishes the theorem. $\square$

Next, we present the proofs of these two lemmas.

Proof of Lemma 7.4. This lemma is a consequence of the following two claims, proved below.

Claim 7.6. $U_{R'}$ is uniquely extendable to $U_{R' \cup K'}$.

Claim 7.7. $U_{R' \cup K'}$ is uniquely extendable to U_W .

By transitivity of unique extendability (Lemma 5.24), it follows from Claims 7.6 and 7.7 that $U_{R'}$ is uniquely extendable to U_W . Furthermore, by Lemma H.1 in [BCFGRS17], $U_{R'}$ is globally extendable. Therefore, by Lemma 5.23 (triangle of extendability), it follows that U_W is globally extendable. $\square$

Next we prove Claims 7.6 and 7.7. Each claim is proven by decomposing one of the upper sets and analyzing the relevant intersections. We present the statements about intersections as facts whose proofs we defer to Appendix C.

Proof of Claim 7.6.

Fact 7.8. For all $y \in K'$,

$$|K_y \cap P_{U_{R'}}| = \dim(\mathcal{F}_{K_y}).$$

By Fact 7.8 and by Lemma 5.25, since $\mathcal{F}_{K_y}$ is MDS, $U_{R'}$ is uniquely extendable to $U_{R'} \cup P_{\geq K_y}$ for each $y \in K'$. Furthermore, for each $y_1 \neq y_2 \in K'$, $K_{y_1} \cap K_{y_2} = \emptyset$. Therefore, by Lemma 5.18, $U_{R'}$ is uniquely extendable to $U_{R'} \cup U_{K'} = U_{R' \cup K'}$. $\square$

Proof of Claim 7.7. Recall that $U_W = U_{R' \cup K'} \cup P_{\geq T}$. We use the following fact.

Fact 7.9.

$$|T \cap P_{U_{R' \cup K'}}| = \dim(\mathcal{F}_T)$$

By Lemma 5.25, since $\mathcal{F}_T$ is MDS, $U_{R'} \cup U_{K'}$ is uniquely extendable to U_W . $\square$

We now turn to the proof of Lemma 7.5.

Proof of Lemma 7.5. The lemma follows from the following two claims by transitivity (Lemma 5.24).

Claim 7.10. $U_{S'}$ is extendable to $U_{S'} \cup U_{K'}$.

Claim 7.11. $U_{S'} \cup U_{K'}$ is extendable to U_W . $\square$

Next, we prove the claims.

Proof of Claim 7.10. We use the following fact.

Fact 7.12. For all $y \in K' \setminus K$,

$$|K_y \cap P_{U_{S'}}| \leq \dim(\mathcal{F}_{K_y}).$$

By Fact 7.12 and by Lemma 5.25, since $\mathcal{F}_{K_y}$ is MDS, $U_{S'}$ is extendable to $U_{S'} \cup P_{\geq K_y}$ for each $y \in K'$. Furthermore, for each $y_1, y_2 \in K' \setminus K$, $K_{y_1} \cap K_{y_2} = \emptyset$. Therefore, by Lemma 5.18, $U_{S'}$ is extendable to $U_{S'} \cup U_{K'}$. $\square$

Proof of Claim 7.11. By definition, $U_W = U_{S'} \cup U_{K'} \cup U_{R' \setminus R}$. We use the following fact.

Fact 7.13. For all $x \in R' \setminus R$,

$$|R_x \cap P_{U_{S'} \cup U_{K'}}| = \dim(\mathcal{F}_{R_x}).$$

Therefore, by Fact 7.13 and by Lemma 5.25, $U_{S'} \cup U_{K'}$ is extendable to $U_{S'} \cup U_{K'} \cup P_{\geq R_x}$ for each $x \in R'$. Furthermore, for each $x_1, x_2 \in R'$, $R_{x_1} \cap R_{x_2} = \emptyset$. Therefore, by Lemma 5.18, $U_{S'} \cup U_{K'}$ is extendable to U_W . $\square$

7.3 Testing Reed-Solomon Codes With Bivariate Reed-Muller Sheaf Codes

In this section, we construct a recursively testable code family (Definition 6.20) for Reed-Solomon codes, using the bivariate Reed-Muller codes described in Section 7.1. For the target code of the family, C_n , we choose the degree bounds so that $(\mathcal{F}_n)_{\sigma=c}$ has a κ -extendable cover for any $c \in C_n$ so that the final PCPP satisfies perfect zero-knowledge. In particular, we choose degree bounds less than or equal to $(2|L_0| - 1, |L_0|/8 - 2)$ that satisfy the conditions of Theorem 7.3, i.e. $(2|L_0| - 1) \geq (|L_0|/8 - 2) + |L_0|$. We start by recalling Definition 7.9 from [BCGRS17].

Definition 7.14. We denote by RS^+ the linear code family indexed by tuples

$$n = (\mathbb{F}, L, d),$$

where $\mathbb{F}$ is an extension field of $\mathbb{F}_2$ and L is an $\mathbb{F}_2$ -linear subspace of $\mathbb{F}$ with

$$d \leq |L|/8,$$

and the n -th code consists of the codewords from the Reed-Solomon code

$$\text{RS}[\mathbb{F}, L, d].$$

Theorem 7.15. RS^+ is recursively testable.

Proof. Let $\hat{k} \geq 0$, $i \geq 1$, and let $(\mathbb{F}, L, d)$ be the i th tuple in RS^+ . Let $\mathcal{F}'[i]$ be the Reed-Muller sheaf code defined in Definition 7.1 over domain $D_{\hat{k}}(L_i)$. First, we show that $\mathcal{F}'[i]$ has a 1-testable cover for $C_i = \text{RS}[\mathbb{F}, L, d]$. Let $P'[i]$ be the coded space for $\mathcal{F}'[i]$. We show the following.

Lemma 7.16. For each $i \in [n]$, $\mathcal{R} \cup \mathcal{K}$ is a 1-testable cover for C_i .

The proof of Lemma 7.16 closely follows the soundness proof of the short PCPP from [BS08] so we defer its proof to Appendix D. Next, by definition of bivariate Reed-Muller sheaf code (Definition 7.1), for each $\rho \in \mathcal{R} \cup \mathcal{K}$, $\mathcal{F}'[i]_\rho$ is equal (up to permutation) to $\text{RS}[\mathbb{F}, L_j, d_j]$ for some $j < i$. This completes the proof. $\square$

7.4 Our Construction

We build our construction using the framework outlined in Section 6.

Construction 7.17 (Perfect Zero-Knowledge PCPP for Reed-Solomon Codes). Let $\mathcal{C} = \text{RS}^+$ be the Reed-Solomon code family (Definition 7.14) and let $C_i = \text{RS}[\mathbb{F}, L, n/8]$. Let $\mathcal{F}'[i]$ be the bivariate Reed-Muller sheaf code over coded space $P'[i]$ (Definition 7.1) over domain $D_1(L)$ such that $P'[i]_T = \text{RS}[\mathbb{F}, L \setminus L'_0, |L|/8 - 1]$. Let $\mathcal{F}, P$ be the output of Construction 6.21 when given code family $\mathcal{C}$, code index i , and recursive depth $\log(\log(|L|))$ as input.

• **Prover:**

- *Input:* Codeword $c \in \text{RS}[\mathbb{F}, L, |L|/8]$
- *Output:* The prover output from Construction 6.3 on inputs $\mathcal{F}, P, c$.

• **Verifier:**

- *Input:* Implicit input $f : \mathbb{F}^{P_\sigma} \rightarrow \mathbb{F}$ and explicit input $g : \mathbb{F}^{P_*} \rightarrow \mathbb{F}$
- *Output:* The verifier output from Construction 6.3 on inputs $\mathcal{F}, P, f, g, T, \mathcal{R} \cup \mathcal{K}$

Next, we use the above construction to prove the existence of a perfect-zero knowledge PCPP for Reed-Solomon codes.

Theorem 7.18. *For all $\epsilon, \delta \in [0, 1]$, there exists a black-box, $\sqrt{n}$ -local, perfect zero-knowledge PCPP for RS^+ with verifier query complexity $\text{polylog}(n)$, zero-knowledge query bound $\sqrt{n}/8 - 2$, soundness error δ , and proximity parameter ϵ .*

Proof. We start by showing that Construction 7.17 is black-box, $\sqrt{n}$ -local, perfect zero-knowledge PCPP with verifier query complexity $O(1)$, zero-knowledge query bound $\sqrt{n}/8 - 2$, soundness error $\epsilon/\text{polylog}(n)$ and proximity parameter ϵ . By Theorem 7.15, $\mathcal{C}$ is a recursive 1-testable code family. Therefore, since P is the output of Construction 6.21, it follows by Lemma 6.24 that P contains a d -testable cover for $\text{RS}[\mathbb{F}, L, |L|/8]$. Next, we apply Theorem 6.4, which just requires us to prove that $\mathcal{F}'[i]$ has a $(\sqrt{n}/8 - 2)$ -extendable cover. By Lemma 6.25, this follows as long as S_i is a $(|L_0|/8 - 2)$ -extendable cover for $\mathcal{F}_i$. This follows from Theorem 7.3 because by definition of our bivariate Reed-Muller sheaf code (Definition 7.1), $\log_2(d_X + 1) = \hat{k} + \log 2(|L|)/2 = \log_2(|L|)/2 + 1$ and so $d_X = 2^{\log_2(|L|)/2 + 1} = 2 \cdot |L_0| - 1$. Definition 7.1 also states that $\mathcal{F}'[i]_T$ has degree bound $d_X + d_Y \cdot |L_0|$. In our case $P'[i]_T$ has degree bound $|L|/8 - 1$. This implies that

$$d_Y = (|L|/8 - 1 - d_X)/|L_0| = |L_0|/8 - 1/|L_0| - d_X/|L_0| = |L_0|/8 - 1/|L_0| - 2 + 1/|L_0| = |L_0|/8 - 2.$$

We show that d_X, d_Y satisfy the conditions of Theorem 7.3 with the following.

$$d_X = 2|L_0| - 1 \geq 9|L_0|/8 - 2 = |L_0| + d_Y$$

By Proposition 2.9 from [BS08], this implies that for any $\epsilon, \delta \in [0, 1]$, we can obtain a PCPP with soundness error δ , proximity parameter ϵ and equivalent proof length to Construction 7.17 and query complexity

$$O\left(\frac{\log(1/\delta)}{(\epsilon/\text{polylog}(n))}\right) = \text{polylog}(n).$$

This completes the proof. □

8 Perfect Zero-Knowledge PCP For NTIME

In this section, we build a black-box perfect zero-knowledge PCP for NTIME using the zero-knowledge PCPP for Reed-Solomon codes from the previous section. Our argument follows that of section 8 in [BCFGRS17], which builds a perfect zero-knowledge k -round Interactive Oracle Proof for NTIME from a k -round perfect zero-knowledge Interactive Oracle Proof of Proximity for Reed-Solomon. Interactive Oracle Proofs (of Proximity) are generalizations of PCPs (resp. PCPPs), where the prover and verifier engage in $k \in \mathbb{N}$ rounds of interaction. Setting $k := 0$ yields the corresponding result for PCPs.

Another difference between our two settings is that they assume a perfect zero-knowledge IOPP whose simulator can tolerate unbounded verifier queries, and only queries one input symbol for every query to the proof. We modify their argument to handle l -local black-box perfect zero-knowledge PCPPs with an explicit query bound. The following is our main result.

Theorem 8.1 (Main Result). *For every polynomial $b, T: \mathbb{N} \rightarrow \mathbb{N}$, there exists a perfect zero-knowledge PCP system (P, V) for $\text{NTIME}(T)$ with the parameters:*

$$\left[\begin{array}{ll} \text{alphabet} & \mathbb{F}(n) = \mathbb{F}_2 \\ \text{proof length} & \ell(n) = \tilde{O}(T(n) + b(n)^2) \\ \text{query complexity} & q(n) = \text{polylog}(n) \\ \text{soundness error} & \varepsilon(n) = 1/2 \\ \text{query bound} & b(n) \end{array} \right].$$

Recall that Dinur [Din07] has shown that a PCP of length ℓ with query complexity q can be transformed to a PZK-PCP of length $\ell \cdot \text{poly}(q)$ with query complexity $O(1)$. In Appendix A, we observe that Dinur’s query reduction and gap amplification transformations are $\text{poly}(q(n))$ -locally computable, hence they preserve perfect zero-knowledge with query bound $b/\text{poly}(q)$. Applying gap amplification (Lemma A.1) to Theorem 8.1, instantiated with query bound $B(n) \cdot \log(n)^c$, for a sufficiently large constant $c > 0$, we obtain the following immediate corollary (note that the additional poly-logarithmic factor is absorbed in the $\tilde{O}$ notation).

Theorem 8.2 (Quasilinear-length PCPs with constant query complexity). *For every polynomial $b, T: \mathbb{N} \rightarrow \mathbb{N}$, there exists a perfect zero-knowledge PCP system (P, V) for $\text{NTIME}(T)$ with the parameters:*

$$\left[\begin{array}{ll} \text{alphabet} & \mathbb{F}(n) = \mathbb{F}_2 \\ \text{proof length} & \ell(n) = \tilde{O}(T(n) + b(n)^2) \\ \text{query complexity} & q(n) = O(1) \\ \text{soundness error} & \varepsilon(n) = 1/2 \\ \text{query bound} & b(n) \end{array} \right].$$

Proof overview. Our proof closely follows the proof of Theorem 8.1 in [BCFGRS17] closely, which itself follows [BCGV16]. Those works construct perfect zero-knowledge interactive oracle proofs (IOPs) for (randomizable) linear-algebraic constraint satisfaction problems ((R)LACSPs), which are shown in [BCGV16] to be NP-complete. IOPs are a multi-round generalization of PCPs. We obtain our main result by replacing the two-message zero knowledge IOP of proximity for Reed–Solomon codes used in those works with our ZKPCPP from Theorem 7.18.

Our ZKPCPP does not precisely fit the requirements laid out in [BCFGRS17], and so it is not a drop-in replacement. In particular, the simulator for the ZKPCPP has locality $\sqrt{d}$, where d is the degree of the polynomial being tested; in the prior work it is assumed that the ZKIOPP has locality 1. In addition, the simulator for our ZKPCPP fails after around $\sqrt{n}$ queries, which must be accounted for in the overall query bound. These modifications are minor and do not affect the structure of the proof.

8.1 Linear Algebraic Constraint Satisfaction Problems

We begin by recalling the definitions of LACSP and RLACSP [BCGV16; BCFGRS17].

Definition 8.3. *Let $g: \Sigma^n \rightarrow \Sigma^m$ be a function. We say that g is ℓ -local if for every $j \in [m]$ there exists $I_j \subseteq [n]$ with $|I_j| \leq \ell$ such that $g(w)_j$ depends only on $w|_{I_j}$. Moreover, we say that g is s -evasive if for every $I \subseteq [n]$ the probability that, for a uniform $j \leftarrow [m]$, I_j intersects I is at most $s \cdot \frac{|I|}{n}$.*

Definition 8.4 (LACSP). *Let $C_0(n)$ and $C_1(n)$ be descriptions of linear codes over $\mathbb{F}(n)$ with block length $\ell(n)$ and relative distance $\tau(n)$. The promise relation of linear algebraic CSPs (LACSPs)*

$$\text{LACSP}[\mathbb{F}(n), C_0(n), C_1(n), \ell(n), \tau(n), q(n),] = (R_{\text{LACSP}}^{\text{YES}}, L_{\text{LACSP}}^{\text{NO}})$$

considers instance–witness pairs (x, w) of the following form.

- An instance x is a tuple $(1^n, g)$, where:
 - $g: \mathbb{F}(n)^{\ell(n)} \rightarrow \mathbb{F}(n)^{\ell(n)}$ is $q(n)$ -local, $q(n)$ -evasive, and can be computed in time $\text{poly}(n)$;
 - $C_1(n) \cup g(C_0(n))$ has relative distance at least $\tau(n)$.
- A witness w is a tuple $w = (w_0, w_1)$, where $w_0, w_1 \in \mathbb{F}(n)^{\ell(n)}$.

The yes-relation $R_{\text{LACSP}}^{\text{YES}}$ consists of all pairs (x, w) as above for which the instance x and witness w jointly satisfy $w_0 \in C_0(n)$, $w_1 \in C_1(n)$, and $g(w_0) = w_1$. In particular, a witness $w = (w_0, g(w_0))$, with $w_0 \in C_0(n)$, satisfies x if and only if $g(w_0) \in C_1(n)$. The no-language $L_{\text{LACSP}}^{\text{NO}}$ consists of all instances x as above such that $x \notin L(R_{\text{LACSP}}^{\text{YES}})$.

Definition 8.5 (randomizability). An LACSP instance $(1^n, g)$ is $t(n)$ -**randomizable** (with respect to code families $C_0(n), C_1(n)$) if there exists a $t(n)$ -wise independent subcode $C' \subseteq C_0(n)$ such that if $(c_0, g(c_0))$ satisfies x then, for every $c'_0 \in c_0 + C'$, the witness $(c'_0, g(c'_0))$ also satisfies x .

Definition 8.6 (RLACSP). The promise relation of randomizable linear algebraic CSPs (RLACSPs) is

$$\text{RLACSP}[\mathbb{F}(n), C_0(n), C_1(n), \ell(n), \tau(n), q(n), c(n), t(n)] = (R_{\text{RLACSP}}^{\text{YES}}, L_{\text{RLACSP}}^{\text{NO}}),$$

where $R_{\text{RLACSP}}^{\text{YES}}$ is obtained by restricting $R_{\text{LACSP}}^{\text{YES}}$ to instances that are $t(n)$ -randomizable, and $L_{\text{RLACSP}}^{\text{NO}} := L_{\text{LACSP}}^{\text{NO}}$.

Theorem 8.7 ([BCGV16]). For any polynomials t, T and any constant $\tau \in (0, 1)$, $\text{RLACSP}[\mathbb{F}, C_0, C_1, \ell, \tau, q, t]$ is $\text{NTIME}(T)$ -hard where $\mathbb{F}$ is a sufficiently large binary field, C_0 is a Reed–Solomon code, C_1 is a subcode of Reed–Solomon that vanishes on a set H , $\ell = \tilde{O}(T + t)$; and $q = \text{polylog}(T)$.

8.2 From zkPCPP for Reed–Solomon Codes to zkPCP For NTIME

The following theorem is adapted from [BCFGRS17, Theorem 8.9].

Theorem 8.8. Suppose that there exists a black-box κ -local perfect zero-knowledge PCPP for the relation $C_0 \times C_1$ with the parameters:

$$\begin{bmatrix} \text{alphabet} & \mathbb{F}(n) \\ \text{proof length} & \hat{l}(n) \\ \text{query complexity} & \hat{q}(n) \\ \text{soundness error} & \hat{\varepsilon}(n) \\ \text{query bound} & \hat{b}(n) \end{bmatrix}.$$

Then there exists a PCP (P_R, V_R) , for relation $(\mathcal{R}_{\text{RLACSP}}^{\text{YES}}, \mathcal{L}_{\text{RLACSP}}^{\text{NO}})[\mathbb{F}, C_0, C_1, \ell, \tau, q, t]$ with parameters

$$\begin{bmatrix} \text{alphabet} & \mathbb{F}(n) \\ \text{proof length} & l(n) = \hat{l}(\ell(n)) + \ell(n) \\ \text{query complexity} & q(n) = \hat{q}(\ell(n)) + q(n) + 1 \\ \text{soundness error} & \varepsilon(n) = \max\{\hat{\varepsilon}(n), 1 - \tau(n) + 2\hat{\delta}(n)(q(n) + 1)\} \\ \text{query bound} & b(n) = \min\left(\frac{t(n)}{\kappa(\ell(n)) \cdot q(n)}, \hat{b}(\ell(n))\right) \end{bmatrix}.$$

Proof sketch. The proof follows [BCFGRS17, Theorem 8.9] very closely; here we outline the differences only, and refer the reader to the prior work for the remainder of the proof. The only modification is in the proof of perfect zero knowledge: for each query the verifier makes to the PCPP (up to $\hat{b}$, after which the zero knowledge guarantee may fail), its simulator will make at most κ queries to w_0, w_1 , resulting in at most $\kappa \cdot q$ queries to w_0 (since w_1 is q -locally computable from w_0). $\square$

Proof of Theorem 8.1. Let $L \in \text{NTIME}(T)$. We construct a ZKPCP for L by first reducing to RLACSP via Theorem 8.7, for sufficiently small constant τ and $t = \max(b(n)^2, T(n))q(n)^2$. A ZKPCPP for $C_0 \times C_1$ can be obtained by combining our Theorem 7.18 with standard techniques for testing vanishing subcodes of Reed–Solomon. That this achieves the specified parameters is easily verified. $\square$

References

- [ALMSS98] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. “Proof verification and the hardness of approximation problems”. In: *Journal of the ACM* 45.3 (1998). Preliminary version in FOCS ’92., pp. 501–555.
- [AS98] S. Arora and S. Safra. “Probabilistic checking of proofs: a new characterization of NP”. In: *Journal of the ACM* 45.1 (1998). Preliminary version in FOCS ’92., pp. 70–122.
- [BCFGRS17] E. Ben-Sasson, A. Chiesa, M. A. Forbes, A. Gabizon, M. Riabzev, and N. Spooner. “Zero Knowledge Protocols from Succinct Constraint Detection”. In: *Proceedings of the 15th Theory of Cryptography Conference*. TCC ’17. 2017, pp. 172–206.
- [BCGRS17] E. Ben-Sasson, A. Chiesa, A. Gabizon, M. Riabzev, and N. Spooner. “Interactive Oracle Proofs with Constant Rate and Query Complexity”. In: *Proceedings of the 44th International Colloquium on Automata, Languages and Programming*. ICALP ’17. 2017, 40:1–40:15.
- [BCGV16] E. Ben-Sasson, A. Chiesa, A. Gabizon, and M. Virza. “Quasilinear-Size Zero Knowledge from Linear-Algebraic PCPs”. In: *Proceedings of the 13th Theory of Cryptography Conference*. TCC ’16-A. 2016, pp. 33–64.
- [BS08] E. Ben-Sasson and M. Sudan. “Short PCPs with Polylog Query Complexity”. In: *SIAM Journal on Computing* 38.2 (2008). Preliminary version appeared in STOC ’05., pp. 551–607.
- [BV26] M. Bafna and N. Vyas. “Private PCPs from Product Expansion”. In: *ECCC TR26-150* (2026).
- [Din07] I. Dinur. “The PCP theorem by gap amplification”. In: *Journal of the ACM* 54.3 (2007), p. 12.
- [GOS24] T. Gur, J. O’Connor, and N. Spooner. “Perfect Zero-Knowledge PCPs for #P”. In: *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*. 2024, pp. 1724–1730.
- [GOS25] T. Gur, J. O’Connor, and N. Spooner. “A zero-knowledge PCP theorem”. In: *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*. 2025, pp. 986–994.
- [KPT97] J. Kilian, E. Petrank, and G. Tardos. “Probabilistically checkable proofs with zero knowledge”. In: *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*. STOC ’97. 1997, pp. 496–505.
- [PK24] P. Panteleev and G. Kalachev. *Maximally Extendable Sheaf Codes*. 2024. arXiv: 2403.03651 [cs.IT]. URL: <https://arxiv.org/abs/2403.03651>.

A Zero-Knowledge Gap Amplification

In this appendix, we observe that Dinur’s query reduction and gap amplification preserve perfect zero-knowledge, at the cost of reducing the query bound by a polynomial factor of the query complexity of the original PCP.

Dinur [Din07] showed that a PCP of length ℓ with query complexity q can be transformed to a PZK-PCP of length $\ell \cdot \text{poly}(q)$ with query complexity $O(1)$. We note that if the original PCP was zero-knowledge with respect to query bound b , then the resulting $O(1)$ -query PCP will have query bound $b/\text{poly}(q)$.

By the local simulatability lemma [GOS25, Lemma 3.2], it suffices to show that the gap amplification procedure is $\text{poly}(q(n))$ -locally computable from the original PCP. We verify this below, assuming familiarity with Dinur [Din07].

Lemma A.1. *Let (P, V) be a non-adaptive PZK-PCP system with constant-size alphabet, polynomial proof length $\ell(n)$, query complexity $q(n)$, query bound $b(n)$, polynomial decision complexity, perfect completeness, and constant soundness error.*

Then, there exist absolute constants $c_1, c_2 > 0$ and a PZK-PCP system (P', V') with binary alphabet, proof length $\ell(n) \cdot q(n)^{c_1}$, query complexity $O(1)$, query bound $b(n)/q(n)^{c_2}$, perfect completeness, and soundness error $1/2$.

Sketch of proof. The transformation and the resulting alphabet size, proof length, query complexity, completeness, and soundness are provided and analysed in [Din07, Theorem 1.5 and Section 8]. It remains to show the resulting PCP is perfect zero-knowledge with query bound $b(n)/q(n)^{c_2}$, for some absolute constant $c_2 > 0$.

Recall the that first step of the transformation are standard arity-reduction and alphabet-reduction steps by appending a large alphabet symbol to each $q(n)$ -tuple of coordinates that may be queried by the non-adaptive verifier and providing a PCPP, with an efficient deterministic prover, for the decision predicates. Clearly, this step is $q(n)$ -local, as each large-alphabet symbol can be obtained by querying $q(n)$ symbols of the original proof, and each PCPP oracle is a function of a single large-alphabet symbol.

Next, write down the PCP system as a constraint graph G of size $\ell(n) \cdot q(n)^{O(1)}$ over an constant-size alphabet Σ , with unsatisfiability gap $\gamma_0 = \Omega(1/q(n))$.

Following the notation in [Din07], recall that each iteration in Dinur’s gap amplification transformation has the form

$$G \rightarrow \text{prep}(G)^t \circ P_{\text{in}} \quad (8)$$

where, $\text{prep}(G)$ is the preparation transformation (degree reduction and expanderization), the powering degree t is a fixed constant, and P_{in} is the inner assignment tester (PCPP) for alphabet reduction. By [Din07, Theorem 1.5], each iteration has unsatisfiability gap

$$\gamma_{i+1} \geq \min\{2\gamma_i, \alpha\},$$

where $\alpha > 0$ is an absolute constant, and increases the proof length by a constant factor.

Observe that each iteration of the transformation in Eq. 8 yields an output proof oracle that is $O(1)$ -locally computable from its input proof oracle. To see this, note that $\text{prep}(G)$ is a $O(1)$ -local transformation by the description of the degree-reduction and expanderization steps, then the constant-degree graph is raised to the t ’th power, for a constant t , which is also $O(1)$ local, and the alphabet reduction is $O(1)$ -local, as observed above.

Finally, note that after $O(\log(1/\gamma(n)))$ iterations of the transformation in Eq. 8, we obtain a constant gap γ , and in turn, the constant soundness error; Moreover, we can further reduce the soundness error to $1/2$ by applying a constant number of repetitions and accepting if all invocations accepted. Since each iteration of the gap amplification procedure increased the proof length by at most a constant and decreased the query bound by at most a constant, we obtain the claimed parameters, where the constant alphabet size can be reduced to binary via standard $O(1)$ -locally computable alphabet reduction. $\square$

B Deferred Proofs Of Facts From Section 6.2

Proof of Fact 6.8.

$$g_\rho(\tau) \subseteq g_\rho(\tau) \cap g'(\sigma) \subseteq \text{Im}(g') \cap \text{Im}(g_\rho(\tau)) = g'(\rho),$$

where the first inclusion follows since by the assumption that $g_\rho(\tau) \subseteq g'(\sigma)$, the second inclusion follows by definition of image of a function and the equality follows by definition of g' and g_ρ . Therefore, $\tau \subseteq g_\rho^{-1}(g'(\rho)) = h_\rho(\rho) = h_\rho(\rho)$. Since τ and $h_\rho(\rho)$ are both in $P[\rho]$, it follows by definition of coded space that $\tau \geq h_\rho(\rho)$ and therefore $\tau \in P[\rho]_{\geq h_\rho(\rho)}$. $\square$

Proof of Fact 6.10.

$$g'(\tau) \subseteq g_\rho(\sigma) \subseteq \text{Im}(g') \cap \text{Im}(g_\rho) = g'(\rho).$$

The first inclusion follows by assumption of the claim, the second inclusion and equality follow by definition of g' , g_ρ . Therefore $\tau \geq \rho$. Since $\tau, \rho \in P'$, it follows by definition of coded space that $\tau \geq \rho$ and therefore $\tau \in P'_{\geq \rho}$. $\square$

Proof of Fact 6.12. Let $\alpha \in h_\beta^{-1}(P[\beta]_{\geq \tau})$. Then $\alpha = g'^{-1}(g_\beta(x))$ for some $x \in P[\beta]_{\geq \tau}$. The following is implied by the claim's assumption that $g_\beta(\tau) \subseteq g_\rho(\sigma)$, by the definitions of g' , g_β , and g_ρ as well as by basic properties of set intersection.

$$g_\beta(x) \subseteq g_\beta(\tau) \subseteq g_\beta(\tau) \cap g_\rho(\sigma) \subseteq g'(\rho) \cap g_\beta(\tau) \cap g_\rho(\sigma) \subseteq g_\rho(\sigma) \cap g'(\rho) \subseteq g'(\rho).$$

Therefore $\alpha = g'^{-1}(g_\beta(x)) \subseteq g'^{-1}(g'(\rho)) = \rho$. Since $\alpha, \rho \in P'$, it follows by the ordering on coded spaces that $\alpha \geq \rho$, which completes the proof. $\square$

Proof of Fact 6.13. The following is implied by the claim's assumption that $g_\beta(\tau) \subseteq g_\rho(\sigma)$, by the definitions of g' , g_β , and g_ρ as well as by basic properties of set intersection.

$$g_\beta(\tau) \subseteq g_\beta(\tau) \cap g_\rho(\sigma) \subseteq g_\beta(\tau) \cap g_\rho(\sigma) \cap \text{Im}(g') \subseteq g_\beta(\tau) \cap \text{Im}(g') \subseteq g'(\beta)$$

Therefore $\tau \subseteq h_\beta(\beta) = h_\rho(\rho)$. Since $h_\rho(\rho)$ and τ are both in $P[\beta]$, the proof follows by the ordering of coded spaces. $\square$

Proof of Fact 6.19.

$$\begin{aligned} U_K^* \cap (g'(P') \cup U_{K^c}) &= (U_K^* \cap g'(P')) \cup (U_K^* \cap U_{K^c}) = U_K^* \cap g'(P') \\ &= \left(\bigcup_{\rho \in K} g_\rho(P[\rho]) \cap g'(P') \right) \cup g'(P'_\sigma) \\ &= \left(\bigcup_{\rho \in K} g'(P'_\rho) \right) \cup g'(P'_\sigma). \end{aligned}$$

This completes the proof of the fact. $\square$

C Deferred Proofs Of Facts From Proof of Theorem 7.3.

In this section, set $\mathcal{F} := \mathcal{F}'[i]$ and $P := P'[i]$.

Fact C.1. For all $y \in q_{L_0}(L_1)$,

$$|(T \cap K_y)| = \sqrt{n}.$$

Proof. For each $y \in q_{L_0}(L_1)$, $|T \cap K_y| = \{(x, y) : q_{L_0}(x) = y\} = \sqrt{n}$, which completes the proof. $\square$

Fact C.2. For all $x \in L$,

$$|(T \cap R_x)| = 1.$$

Proof. An X -curve intersects with each row exactly once. $\square$

Proof of Fact 7.8.

$$\begin{aligned} |K_y \cap P_{U_{R'}}| &= |K_y \cap \bigcup_{x \in R'} R_x| = |\bigcup_{x \in R'} K_y \cap R_x| = \sum_{x \in R'} |K_y \cap R_x| = |R'| \cdot 1 \\ &= d_X + 1 = \dim(\mathcal{F}_{K_y}) \end{aligned}$$

The first equality follows by definition of U_1 , the second equality follows by distributivity of set intersection, the third equality follows because for each $x_1, x_2 \in R'$, $P_{R_{x_1}} \cap P_{R_{x_2}} = \emptyset$, the fourth equality follows because for each $(x, y) \in \mathbb{F}^{P*}$, $|K_y \cap R_x| = 1$, the fifth equality follows by definition of R' , and the final equality follows by definition of $\mathcal{F}$. $\square$

Proof of Fact 7.9.

$$\begin{aligned} |T \cap P_{U_{R'} \cup U_{K'}}| &= |T \cap (\bigcup_{x \in R'} R_x \cup \bigcup_{y \in K'} K_y)| \\ &= |T \cap \bigcup_{x \in R'} R_x| + |T \cap \bigcup_{y \in K'} K_y| = |(\bigcup_{x \in R'} T \cap R_x)| + |(T \cap (\bigcup_{y \in K'} K_y))| \\ &= \sum_{x \in R'} |T \cap R_x| + \sum_{y \in K'} |T \cap K_y| = |R'| \cdot 1 + |K'| \cdot |L_0| = (d_X + 1) + d_Y \cdot |L_0| \\ &= \dim(\mathcal{F}_T) \end{aligned}$$

The first equality follows definitions of $U_{R'}, U_{K'}$, the second equality follows by distributivity of set intersection, the third equality follows because for each $x \in R', y \in K'$, $R_x \cap K_y \cap T = \emptyset$, the fourth equality follows because for every $x_1, x_2 \in R'$ (resp. $y_1, y_2 \in K'$), $R_{x_1} \cap R_{x_2} = \emptyset$ (resp. $K_{y_1} \cap K_{y_2} = \emptyset$), the fifth equality follows by Facts C.2 and C.1, the sixth equality follows by definitions of R', K' , and the final two equalities follow by the definition of T . $\square$

Proof of Fact 7.12.

$$\begin{aligned} &|P_{K_{y'}} \cap P_{U_S \cup P_{\geq \sigma}}| \\ &= |K_y \cap (P_\sigma \cup \bigcup_{x \in R} R_x \cup \bigcup_{y' \in K} P_{K_{y'}})| \\ &= |(K_y \cap P_\sigma) \cup (K_y \cap \bigcup_{x \in R} R_x) \cup (K_y \cap \bigcup_{y' \in K} P_{K_{y'}})| \\ &= |(K_y \cap P_\sigma) \cup (\bigcup_{x \in R} K_y \cap R_x)| \\ &= |(K_y \cap P_\sigma)| + |(\bigcup_{x \in R} K_y \cap R_x)| \\ &= |K_y \cap P_\sigma| + \sum_{x \in R} |K_y \cap R_x| \\ &= |L_0| + |R| \cdot 1 \leq |L_0| + d_Y \leq d_X < \dim(\mathcal{F}_{K_y}) \end{aligned}$$

The first equality follows by definition of U_S , the second follows by distributivity of set intersection, the third follows by Claim 2.1 because for all $y_1, y_2 \in K$, $P_{K_{y_1}} \cap P_{K_{y_2}} = \emptyset$, the fourth follows because for each $x \in R', y \in K'$, $R_x \cap K_y \cap P_\sigma = \emptyset$, the fifth follows because for all $x_1, x_2 \in R$, $P_{R_{x_1}} \cap P_{R_{x_2}} = \emptyset$, the sixth follows by Fact C.1. On the last line, the first inequality follows by definition of R , the second inequality follows by assumption on d_X , and the last inequality follows by definition of $\mathcal{F}$. $\square$

Proof of Fact 7.13.

$$\begin{aligned}
& |R_x \cap P_{U_S \cup P_{\geq T \cup U_{K'}}}| \\
&= |R_x \cap (T \cup \bigcup_{x' \in R} P_{R_{x'}} \cup \bigcup_{y \in K'} K_y)| \\
&= |(R_x \cap T) \cup (R_x \cap \bigcup_{x' \in R} P_{R_{x'}}) \cup (R_x \cap \bigcup_{y \in K'} K_y)| \\
&= |(R_x \cap T) \cup (R_x \cap \bigcup_{y \in K'} K_y)| \\
&= |(R_x \cap T)| + |(R_x \cap \bigcup_{y \in K'} K_y)| \\
&= |(R_x \cap T)| + |\bigcup_{y \in K'} R_x \cap K_y| \\
&= |(R_x \cap T)| + \sum_{y \in K'} |R_x \cap K_y| \\
&= 1 + |K'| \cdot 1 = d_Y + 1 = \dim(\mathcal{F}_{R_x})
\end{aligned}$$

The first equality follows by definition of U_S , the second one follows by distributivity of set intersection, the third follows by Claim 2.1, because for every $x_1, x_2 \in R'$, $P_{R_{x_1}} \cap P_{R_{x_2}} = \emptyset$, the fourth follows because for each $x \in R'$, $y \in K'$, $R_x \cap K_y \cap P_\sigma = \emptyset$, the fifth follows by distributivity of set intersection, the sixth follows because for each $y_1 \neq y_2 \in K'$, $K_{y_1} \cap K_{y_2} = \emptyset$, the seventh follows by Fact C.2, and the final equalities follow by definition of K' and $\mathcal{F}$. $\square$

D Proof Of Lemma 7.16

Proof. We follow the proof of Lemma 6.11 in [BS08] closely. Let $\mathcal{F} := \mathcal{F}_i$, $P := P_i$, and $S := S_i$. Let $f \in \mathbb{F}^T$ and let

$$\epsilon = \mathbb{E}_{\rho \leftarrow \mathcal{D}(S_i)} [\Delta(f_{P_\rho}, \mathcal{F}_\rho)].$$

We prove that there exists constant $c^* \in \mathbb{N}$ such that $\Delta(f, \mathcal{F}_T) \leq c^* \cdot \epsilon$. We start with the following claim.

Claim D.1. *For all $f \in \mathbb{F}^P$, there exists a constant $c \in \mathbb{N}$ such that $\Delta(f, \mathcal{F}(P)) \leq c \cdot \epsilon$.*

Proof. We use the following constants, where $\hat{c}$ is a constant to be minimized and c_0 is the universal constant from Lemma 6.13 in [BS08].

$$c_1 := \frac{\hat{c}}{2} \quad c_2 := \frac{c_1}{3c_0} \quad c_3 := \frac{3c_1}{16(3c_0 + 2)}$$

We fix c to be the minimal $\hat{c}$ such that $c_3 \geq 2$ and $\frac{1}{c_3} + \frac{2}{c_1} < 1$. Let $\epsilon_{row} := \mathbb{E}_{\rho \leftarrow \text{row}(P)} [\Delta(f_{P_\rho}, \mathcal{F}_\rho)]$ and let $\epsilon_{col} := \mathbb{E}_{\rho \leftarrow \text{col}(P)} [\Delta(f_{P_\rho}, \mathcal{F}_\rho)]$. Then by definition of $\mathcal{D}(S)$, $(\frac{1}{2}) \cdot (\epsilon_{col} + \epsilon_{row}) = \epsilon$ and so $\epsilon_{col}, \epsilon_{row} \leq 2\epsilon$.

Next, we consider the restriction of f to a product set. Let $G = L'_0 \times q_{L_0}(L_1)$. Let $\mathcal{E}, Q$ be the sheaf and coded space for a bivariate Reed-Muller sheaf code over domain G . Let $S_G := \mathcal{R} \cup \mathcal{K}$. Let $\mathcal{D}(S_G) := \mathcal{D}(S_i)$. Let $\epsilon'_{col} := \mathbb{E}_{\rho \leftarrow \text{col}(Q)} [\Delta(f|_{P_\rho}, \mathcal{E}_\rho)]$ and let $\epsilon'_{row} := \mathbb{E}_{\rho \leftarrow \text{row}(Q)} [\Delta(f|_{P_\rho}, \mathcal{E}_\rho)]$. Since for each $\rho \in \text{col}(P)$, $|Q_\rho|/|P_\rho| = 2$, we have the following.

$$\epsilon'_{col} \leq 2\epsilon_{col} \leq 4\epsilon = 2c\epsilon \cdot \frac{1}{c_1}.$$

Furthermore, since for each $\rho \in \text{row}(P)$, $Q_\rho = P_\rho$, it follows that

$$\epsilon'_{row} = \epsilon_{row} \leq 2\epsilon = c\epsilon \frac{1}{c_1}.$$

By Lemma 6.13 in [BS08], we have the following. For all $f \in \mathbb{F}^Q$, there exists a constant c_0 such that

$$\Delta(f, \mathcal{E}(Q)) \leq c_0(\epsilon'_{row} + \epsilon'_{col}) = c\epsilon \frac{1}{c_2}.$$

Let $U_{col} := \bigcup_{y \in \text{col}(P)} P_{\geq y}$. Let d be the closest codeword in $\mathcal{E}(Q)$ to $f|_G$ and let e be the closest codeword in $\mathcal{E}(U_{col})$ to $f|_G$. We say β is bad if $e|_{Q_\beta} \neq d|_{Q_\beta}$. Otherwise β is good.

$$\Pr_{(x,y) \in P_*} [f(x,y) \neq d(x,y)] = \Pr_{\beta \in \text{col}(P)} [\beta \text{ is bad}] + \Pr_{(\alpha,\beta) \in P_*} [f(\alpha,\beta) \neq d(\alpha,\beta) \mid \beta \text{ is good}]. \quad (9)$$

First we analyze the left hand side of the sum. If β is bad, then $e|_{Q_\beta} \neq d|_{Q_\beta}$ and so $f|_{Q_\beta}$ is either $3/8$ -far from $e|_{Q_\beta}$ or $(3/8)$ -far from $d|_{Q_\beta}$, because otherwise, it follows from the triangle inequality that $\Delta(e|_{Q_\beta}, d|_{Q_\beta}) \leq 6/8 = 3/4$ which is a contradiction since the minimum distance of the code is $3/4$. $\text{col}(Q)$ is a cover for both $\mathcal{E}(Q)$ and $\mathcal{E}(\text{col}(Q))$. Each code in the cover is equal to $RS[\mathbb{F}, Q_*, 3/4]$ and every two codes in the cover have disjoint domains. Therefore, we apply Claim 4.10 and have the following.

$$\begin{aligned} & \Pr_{\beta \leftarrow \text{col}(P)} [\beta \text{ is bad}] \\ & \leq \Pr_{\beta \in \text{col}(Q)} [\Delta(f|_{Q_\beta}, d|_{Q_\beta}) > 3/8] + \Pr_{\beta \in \text{col}(Q)} [\Delta(f|_{Q_\beta}, e|_{Q_\beta}) > 3/8] \quad (\text{By the triangle inequality}) \\ & \leq \frac{8}{3} \cdot \Delta(f|_Q, d) + \frac{8}{3} \cdot \Delta(f|_Q, e) \quad (\text{By Claim 4.10}) \\ & = 2 \cdot \frac{8}{3} \cdot \max\left(\frac{1}{c_2}, \frac{2}{c_1}\right) \cdot \epsilon c \end{aligned} \quad (10)$$

$$\leq \frac{\epsilon c}{c_3} \quad (11)$$

Since we minimized $\hat{c}$ so that $c_3 \geq 2$ and since we can assume that $c\epsilon \leq 1$, we analyze the right hand side of the sum in Equation 9 as follows.

$$\begin{aligned} & \Pr_{(\alpha,\beta) \in P_*} [f(\alpha,\beta) \neq d(\alpha,\beta) \mid \beta \text{ is good}] = \Pr_{(\alpha,\beta) \in P_*} [f(\alpha,\beta) \neq e(\alpha,\beta) \mid \beta \text{ is good}] \\ & \leq \frac{\Pr[f(\alpha,\beta) \neq e(\alpha,\beta)]}{\Pr[\beta \text{ is good}]} = \frac{\Delta(f,e)}{1 - \Pr[\beta \text{ is bad}]} \leq 2\Delta(f,e) \leq 2 \cdot \frac{c\epsilon}{c_1} = \frac{2c\epsilon}{c_1} \end{aligned}$$

Summing up the above two equations we have that

$$\Pr_{(x,y) \in P_*} [f(x,y) \neq d(x,y)] = \left(\frac{1}{c_3} + \frac{2}{c_1}\right)c\epsilon$$

Since we assume $\frac{1}{c_3} + \frac{2}{c_1} \leq 1$, this completes the proof. $\square$

Finally we are ready to complete the proof of the lemma. By Definition 7.1, $|T \cap K_\beta| = |L_0|$ for all $\beta \in \text{col}(P)$. Therefore $|T| = |L_0| \cdot |L_1|$. Next,

$$\frac{|P_*|}{|T|} = \frac{\sum_{\beta \in \text{col}(P)} |L_\beta|}{|L_0| \cdot |L_1|} = \frac{|L_\beta| \cdot |L_1|}{|L_0| \cdot |L_1|} = \frac{|L_\beta|}{|L_0|}, \quad (12)$$

which is a constant. Therefore, since by Claim D.1, $\mathcal{R} \cup \mathcal{K}$ is a 1-testable cover for $\mathcal{F}(P)$, it follows by Corollary 4.9 and Equation 12 that $\mathcal{R} \cup \mathcal{K}$ is a 1-testable cover for

$$\mathcal{F}_T = RS[\mathbb{F}, T_X, \deg^*(T, d_X, d_Y)] = C_i.$$

$\square$