Interactive Oracle Proofs of Proximity to Algebraic Geometry Codes

Sarah Bordage[1,2] and Jade Nardi[2,1]⋆⋆

[1] LIX, CNRS UMR 7161, Ecole Polytechnique, Institut Polytechnique de Paris
sarah.bordage@lix.polytechnique.fr
[2] Inria
jade.nardi@univ-rennes1.fr

Abstract. The problem of testing proximity to an error-correcting code C consists in distinguishing between the case where an input word, given as an oracle, belongs to C and the one where it is far from every codeword of C. Algebraic Geometry (AG) codes are good candidates to construct *short* proof systems, but there exists no efficient proximity tests for them. We construct an Interactive Oracle Proof of Proximity (IOPP) for some families of AG codes by generalizing an IOPP for Reed-Solomon codes, known as the FRI protocol [6]. We identify suitable requirements for designing efficient IOPP systems for AG codes. Our approach relies on Kani's result that splits the Riemann-Roch space of any invariant divisor under a group action on a curve into several explicit Riemann-Roch spaces on the quotient curve [17]. Under some hypotheses, a proximity test to C is thus reduced to one to a simpler code C'. Iterating this process thoroughly, we end up with a membership test to a code with significantly smaller length.
In addition to proposing the first proximity test targeting AG codes, our IOPP admits quasilinear prover arithmetic complexity and sublinear verifier arithmetic complexity with constant soundness for meaningful classes of AG codes. As a concrete instantiation, we study AG codes on Kummer curves, which are potentially much longer than Reed-Solomon codes. For these curves, we extend our generic construction to reach a strictly linear proving time and a strictly logarithmic verification time.

1 Extended abstract

Under the generic term of *arithmetization* [20], algebraic techniques for constructing proof systems using properties of low-degree polynomials emerged from the study of interactive proofs (IPs, [14]). Arithmetization techniques have been enhanced and fruitfully applied to other families of proof systems since then, including probabilistically checkable proofs (PCPs, [4,2,1]). To construct a proof system for a non-deterministic relation $\mathcal{R}$, arithmetization turns any instance-witness pair (x, w) into a word that lies in a given error-correcting code C if $(x, w) \in \mathcal{R}$, and is very far from C otherwise.

⋆⋆ [0000−0003−0901−7266]

Since the seminal works of Kilian [19] and Micali [22], a lot of efforts have been put into making PCPs efficient enough to obtain *practical* sublinear non-interactive arguments for delegating computation. In search of reducing the work required to generate such probabilistic proofs, as well as the communication complexity of succinct arguments based on them, Interactive Oracle Proofs (IOPs) have been introduced as a generalization of both PCPs and IPs.

Considering for the first time univariate polynomials instead of multivariate ones, [13,**?**] constructed a PCP with quasilinear proof length and constant query complexity. Since then, efficient transparent and zero-knowledge non-interactive arguments have been designed by relying on Reed-Solomon (RS) codes, *e.g.* [7,9,18]. At some point, aforementioned arguments require a proximity test to RS codes. One can use a prover-efficient Reed-Solomon IOP of Proximity, which is an interactive variant of PCP of Proximity introduced by [8]. The state-of-the-art IOPP for RS codes is known as the FRI protocol, firstly introduced in [6].

In 2013, [11] construct a PCP with linear proof length and sublinear query complexity for boolean circuit satisfiability by using AG codes. For any $\varepsilon > 0$ and instances of size n, their PCP has length $2^{O(1/\varepsilon)}n$ and query complexity n^{ε}. When aiming at optimal proof length and query complexity as small as possible, this result remains the state-of-the-art PCP design. By using AG codes, the authors of [11] reduce the field size to a constant, which avoids a logarithmic blowup in proof bit-length (as in [13]). However, they are not able to apply proof composition [2] to reduce the query complexity of their PCP because decision complexity of the PCP verifier is too large (polynomial in the query complexity).

Improving on [11], [8] construct an interactive oracle proof (IOP, [10]) for boolean circuit satisfiability with linear proof length and constant query complexity. However, prover and verifier complexities are still super-linear. The IOP of [8] invokes the sumcheck protocol [20] on $O(1)$-wise tensor product of AG codes, which exponentially deteriorates the rate of the base code. Then, they use Mie's PCP of Proximity for non-deterministic languages [23] to test proximity to the tensored code. Both constructions benefit from AG codes to get constant size alphabet and linear proof bit-lengths. Still, prover and verifier running times prevent them from being implemented for verifying meaningful computations.

A recent work of [24] constructs an IOPP for any deterministic language which can be decided in time $\mathrm{poly}(n)$ and space $n^{o(1)}$, with constant round and query complexities, linear proof length and sublinear verification. This might be applied to test proximity to AG codes but prover running time is $\mathrm{poly}(n)$, which can obstruct implementations. By contrast, we exhibit specific families of AG codes for which we construct a proximity test with linear prover running time and logarithmic verification. Our construction is also simpler to implement, since the prover and the verifier mainly perform small degree univariate interpolations.

The FRI protocol for RS proximity testing admits linear prover time, logarithmic verifier time and logarithmic query complexity. A natural question is whether an IOPP targeting AG codes of similar efficiency can be constructed. Indeed, AG codes [15], as evaluations of a set of functions at some designated points

on a given curve, extend the notion of Reed-Solomon codes and inherit many of their interesting properties. A key feature for a family of codes to be suitable for arithmetization is a multiplication property [21], namely the component-wise multiplication of two codewords lies in a code with good minimum distance. AG codes not only feature this property but may also have arbitrary large length over a fixed finite field $\mathbb{F}$, unlike RS codes. For concrete efficiency, complexity measures (proof length, query complexity, prover/verifier time) are closely examined and reducing the size of the alphabet directly impacts the binary complexities.

Prover complexity is actually the main bottleneck in deploying zero-knowledge proof systems for large computations. The running time of the prover is bounded from below by the encoding time during arithmetization. In this direction, some one-point AG codes, *e.g.* on Kummer type curves, are especially appealing. For instance, there exist a quasilinear encoding algorithm for AG codes on the Hermitian curve, which is a special case of Kummer type curves [5].

This motivates a part of our study dedicated to the case of AG codes on Kummer type curves. To encourage the search for suitable families of AG codes, we study generic conditions to perform proximity testing. By constructing an efficient IOPP for AG codes, we hope that it opens up new possibilities for designing efficient probabilistic proof systems with short proofs, without requiring tensor product codes.

1.1 Definition of an IOPP for a code C

Let $C \subseteq \Sigma^S$ be an evaluation code with domain S of size n and alphabet Σ. We measure the distance between $u, u' \in \Sigma^S$ with the *relative Hamming distance* Δ, namely the ratio of coordinates in which they differ. For a code $C \subseteq \Sigma^S$, the distance of $u \in \Sigma^S$ from C, denoted by $\Delta(u, C)$, is the minimal distance between u and a codeword of C. If $\Delta(u, C) > \delta$, we say that u is δ-far from C, and δ-close otherwise. As mentioned earlier, we address the problem of proximity testing to a code C, *i.e.* given a code C and assuming a verifier has oracle access to a function $f : S \to \Sigma$, determine whether $f \in C$ or f is δ-far from C. Here, we focus on the case where C is an AG code. An algebraic geometry (AG) code $C = C(\mathcal{C}, \mathcal{P}, D)$ is a vector space formed by the evaluations on $\mathcal{P} \subset \mathcal{C}$ of functions in the Riemann-Roch space $L_{\mathcal{C}}(D)$. We address this problem in the IOP model, which proved to be particularly promising for the design of proof systems.

We are specifically interested in public-coin IOP of Proximity (IOPP) for a family of evaluation codes $\mathscr{C}$, thereby we specify our definition for this particular setting. An IOPP (P, V) for a code C is a pair of randomized algorithms, where both P (the prover) and V (the verifier) receive as explicit input the specification of a code $C \subseteq \Sigma^S$. We define the input size to be $n = |S|$. Furthermore, a purported codeword $f : S \to \Sigma$ is given as explicit input to P and as an oracle to V. The prover and the verifier interact over at most $r(n)$ rounds and during this conversation, P seeks to convince V that f belongs to the code C.

At each round, the verifier sends a message chosen uniformly and independently at random, and the prover answers with an oracle. After the end of the

interaction with the prover, verifier queries the prover's messages using public randomness. Thus, such an IOPP is a *public-coin* protocol (or Arthur-Merlin [3]).

Let us denote $\langle \mathsf{P} \leftrightarrow \mathsf{V} \rangle \in \{\mathsf{accept}, \mathsf{reject}\}$ the output of V after interacting with P. The notation V^f means that f is given as an oracle input to V. We say that a pair of randomized algorithms (P, V) is an IOPP system for the code $C \subseteq \Sigma^S$ with *soundness error* $s : (0, 1] \to [0, 1]$, if the following conditions hold:

Perfect completeness: If $f \in C$, then $\Pr[\langle \mathsf{P}(C, f) \leftrightarrow \mathsf{V}^f(C) \rangle = \mathsf{accept}] = 1$.
Soundness: For any function $f \in \Sigma^S$ such that $\delta := \Delta(f, C) > 0$ and any unbounded malicious prover P^*, $\Pr[\langle \mathsf{P}^* \leftrightarrow \mathsf{V}^f(C) \rangle = \mathsf{accept}] \leq s(\delta)$.

The length of any prover message is expressed in number of symbols of an alphabet $a(n)$. The sum of lengths of prover's messages defines the proof length $l(n)$ of the IOPP. The query complexity $q(n)$ is the total number of queries made by the verifier to both the purported codeword f and the oracle sent by the prover during the interaction. The prover complexity $t_p(n)$ is the time needed to generate prover messages during the interaction (which does not include the input function f). The verifier complexity $t_v(n)$ is the time spent by the verifier to make her decision when queries and query-answers are given as inputs.

Let $\mathcal{R}_{\mathscr{C}}$ be the relation consisting of instance-witness pairs (C, f) where $C \subset \Sigma^S$ lies in $\mathscr{C}$ and $f : S \to \Sigma$. We say that $\mathcal{R}_{\mathscr{C}}$ belongs to the complexity class $\mathsf{IOPP}[a, r, l, q, \delta, s]$ if on inputs of size n, there is an IOPP system testing proximity of f to C with alphabet $a(n)$, round complexity $r(n)$, proof length $l(n)$, query complexity $q(n)$, proximity parameter $\delta(n)$ and soundness error $s(n)$.

1.2 Our results

Construction of an IOPP for foldable AG codes. Firstly, we give a criterion for building an efficient IOPP for AG codes. Let $\mathcal{C}_0$ be a curve defined over a finite field $\mathbb{F}$, D_0 a divisor on the curve $\mathcal{C}_0$ and $\mathcal{P}_0 \subset \mathcal{C}(\mathbb{F})$. This defines an AG code $C_0 = C(\mathcal{C}_0, \mathcal{P}_0, D_0)$. We construct a sequence of curves

$$\mathcal{C}_0 \xrightarrow{\pi_0} \mathcal{C}_1 \xrightarrow{\pi_1} \mathcal{C}_2 \xrightarrow{\pi_2} \cdots \xrightarrow{\pi_{r-1}} \mathcal{C}_r,$$

and a sequence of AG codes $C_i := C(\mathcal{C}_i, \mathcal{P}_i, D_i)$ of decreasing length to turn the proximity test of the function $f^{(0)} = f$ to C_0 into a membership test of a function $f^{(r)}$ in C_r. In the above sequence of curve, the curve $\mathcal{C}_{i+1}$ arises as the quotient of the curve $\mathcal{C}_i$ by a cyclic group $\mathbb{Z}/p_i\mathbb{Z}$ under the projection π_i. We show that such a procedure is made possible by the action of a large solvable group $\mathcal{G}$ on the curve $\mathcal{C}_0$ and some hypotheses on the divisor D_0. A code fulfilling all the conditions we require will be called *foldable*. We design an IOPP for testing proximity to any foldable AG code $C(\mathcal{C}_0, \mathcal{P}_0, D_0)$ with linear proof length, sublinear query complexity and constant soundness. Efficiency parameters of this protocol, called AG-IOPP, are given by the following theorem.

Theorem 1 (informal). *Let $\mathcal{R}_C$ be the relation of instance-witness pairs $((\mathcal{C}_0, \mathcal{P}_0, D_0), f^{(0)})$ such that $C_0 = C(\mathcal{C}_0, \mathcal{P}_0, D_0)$ is a foldable AG code and $f^{(0)} \in C_0$. We denote $n = |\mathcal{P}_0|$. As C_0 is a foldable code, there is a solvable group $\mathcal{G}$ acting on $\mathcal{C}_0$. Assume there exists $e \in (0,1)$ such that $|\mathcal{G}| > n^e$. For every proximity parameter $\delta \in (0,1)$, there exists a public-coin IOPP system (P, V) with perfect completeness putting $\mathcal{R}_C$ in the complexity class*

$$\mathsf{IOPP}\left[\begin{array}{ll} \textit{alphabet} & a(n) = \mathbb{F} \\ \textit{randomness} & k(n) = O(\log n) \\ \textit{rounds} & r(n) = O(\log n) \\ \textit{proof length} & l(n) = O(n) \\ \textit{query complexity} & q(n) = O(n^{1-e}) \\ \textit{proximity parameter} & \delta(n) = \delta \\ \textit{soundness error} & s(n) = 1/2 \end{array}\right].$$

We emphasize that the larger is the group $\mathcal{G}$ acting on $\mathcal{C}_0$ compared to n, the smaller are the query complexity and the verifier decision complexity.

AG-IOPP with linear prover and logarithmic verifier on Kummer curves. When $\mathcal{C}_0$ is a Kummer curve of the form $y^N = f(x)$, we show how to choose $\mathcal{P}_0$ and D_0 to make the AG code $C_0 = C(\mathcal{C}_0, \mathcal{P}_0, D_0)$ foldable. We benefit from the action of the group $\mathbb{Z}/N\mathbb{Z}$ on $\mathcal{C}_0$ that yields a quotient curve $\mathcal{C}_0/(\mathbb{Z}/N\mathbb{Z})$ isomorphic to the *projective line*. This enables us to define a sequence of codes $(C_i)_{0 \le i \le s}$ such that the code C_s is a *RS code* of dimension $(\deg D_0)/N + 1$, which is itself a foldable AG code. Leveraging this fact, we extend the IOPP for generic foldable AG codes to construct a very effective AG-IOPP for codes on Kummer curves, with linear prover running time and strictly logarithmic verification (with respect to the blocklength of the first code). We get the following improvement.

Theorem 2 (informal). *Let $\mathcal{R}_{C'}$ be the relation of instance-witness pairs $((\mathcal{C}_0, \mathcal{P}_0, D_0), f^{(0)})$ such that $C_0 = C(\mathcal{C}_0, \mathcal{P}_0, D_0)$ is a foldable AG code, $\mathcal{C}_0$ is a Kummer curve of equation $\mathcal{C}_0 : y^N = f(x)$ such that $\deg f \equiv -1 \mod N$, N is a smooth integer, coprime with $|\mathbb{F}|$, and $f^{(0)} \in C_0$. We denote $n = |\mathcal{P}_0|$. For every proximity parameter $\delta \in (0,1)$, there exists a public-coin IOPP system (P, V) with perfect completeness putting $\mathcal{R}_{C'}$ in the complexity class*

$$\mathsf{IOPP}\left[\begin{array}{ll} \textit{alphabet} & a(n) = \mathbb{F} \\ \textit{randomness} & k(n) = O(\log n) \\ \textit{rounds} & r(n) = O(\log n) \\ \textit{proof length} & l(n) = O(n) \\ \textit{query complexity} & q(n) = O(\log n) \\ \textit{proximity parameter} & \delta(n) = \delta \\ \textit{soundness error} & s(n) = 1/2 \end{array}\right].$$

Prover complexity is $O(n)$ and verifier decision complexity is $O(\log n)$.

The Hermitian curve defined over $\mathbb{F}_{q^2}$ by $y^{q+1} = x^q + x$ satisfies the hypotheses of the theorem above. It is well known to have many rational points with respect to its geometry. We thus provide family of codes much longer than RS codes that are endowed with a proximity test as efficient as the FRI protocol.

1.3 Technical overview

Our IOPP construction relies on the generalization of the FRI protocol to AG codes. We recall some ideas behind the construction of FRI protocol (see e.g. [12] for details) and we describe how we tailor these ideas.

The FRI protocol for RS proximity testing. Let k be a positive integer and $\rho \in]0,1[$ such that $\rho = 2^{-k}$. The FRI protocol allows to check proximity to the Reed-Solomon code $\mathsf{RS}[\mathbb{F}, \mathcal{P}, \rho] := \left\{ f \in \mathbb{F}^{\mathcal{P}} \mid \deg f < \rho |\mathcal{P}| \right\}$ by testing proximity to $\mathsf{RS}[\mathbb{F}, \mathcal{P}', \rho]$ with $|\mathcal{P}'| < |\mathcal{P}|$. The FRI protocol considers a family of linear maps $\mathbb{F}^{\mathcal{P}} \to \mathbb{F}^{\mathcal{P}'}$ which randomly "fold" any function in $\mathbb{F}^{\mathcal{P}}$ into a function in $\mathbb{F}^{\mathcal{P}'}$. The following three key ingredients enable the FRI protocol to work.

(a) *Splitting of polynomials.* For any polynomial f of degree $\deg f < \rho n$, there exist two polynomials g, h of degree $< \frac{1}{2}\rho n$ such that

$$f(x) = g\left(x^2\right) + x \cdot h\left(x^2\right). \tag{1}$$

This decomposition means that of the space of polynomials of degree less than ρn into two copies of the space of polynomials of degree less than $\rho n/2$.

(b) *Randomized folding.* Choose $\mathcal{P}$ to be a multiplicative group of order 2^r generated by $\omega \in \mathbb{F}$. Then, define $\mathcal{P}' = \langle \omega^2 \rangle = \{x^2 \mid x \in \mathcal{P}\}$. Set $\pi : \mathbb{F} \to \mathbb{F}$ to be the map defined by $\pi(x) = x^2$, observe that $\pi(\mathcal{P}) = \mathcal{P}'$ and $|\mathcal{P}'| = |\mathcal{P}|/2$. The structure of the evaluation domain will allow to reduce the problem of proximity to one of half the size at each round of interaction.
Based on the decomposition (1), for any $z \in \mathbb{F}$ we define a *folding operator* $\mathbf{Fold}[\cdot, z] : \mathbb{F}^{\mathcal{P}} \to \mathbb{F}^{\mathcal{P}'}$ by $\mathbf{Fold}[f, z] := g + zh$. If $\deg f < \rho n$, both functions $g : \mathcal{P}' \to \mathbb{F}$ and $h : \mathcal{P}' \to \mathbb{F}$ belong to $\mathsf{RS}[\mathbb{F}, \mathcal{P}', \rho]$. Then for any random challenge $z \in \mathbb{F}_q$, the operator $\mathbf{Fold}[\cdot, z]$ maps $\mathsf{RS}[\mathbb{F}, \mathcal{P}, \rho]$ into $\mathsf{RS}[\mathbb{F}, \mathcal{P}', \rho]$.

(c) *Folding preserves distance.* Except with small probability over z, we have

$$\Delta(f, \mathsf{RS}[\mathbb{F}, \mathcal{P}, \rho]) \geq \delta \;\Rightarrow\; \Delta\left(\mathbf{Fold}[f, z], \mathsf{RS}[\mathbb{F}, \mathcal{P}', \rho]\right) \geq (1 - o(1))\delta.$$

The protocol goes as follows: the verifier sends a random challenge $z \in \mathbb{F}$ and the prover answers with an oracle function $f' : \mathcal{P}' \to \mathbb{F}$, which is expected to be $\mathbf{Fold}[f, z]$. At the next round, f' becomes the function to be folded, and the process is repeated for r rounds. Each round reduces the problem by half, eventually leading to a function $f^{(r)}$ evaluated over a small enough domain. This induces a sequence of RS codes of strictly decreasing length, but with constant code rate and relative minimum distance. The final test consists in checking that $f^{(r)}$ belongs to the last RS code.

Perfect completeness follows from Item (b). Prover and verifier efficiencies of the FRI protocol come from the possibility of determining any value of $\mathbf{Fold}[f, z]$ at a point $y \in \mathcal{P}'$ with exactly two values of f, namely on the set $\pi^{-1}(\{y\})$. Consequently, a single test of consistency between f and f' requires only two

queries to f and one query to f'.

Soundness of the protocol relies on Item (c). It is proved using results about distance preservation under random linear combinations, that could be roughly stated as follows: "*Let $V \subset \mathbb{F}_q^n$ be a linear code and $g, h \in \mathbb{F}_q^n$. As long as δ is small enough, if we have $\Delta(g + zh, V) \leq \delta$ for enough values $z \in \mathbb{F}_q$, then both g and h are δ'-close to V, where $\delta' = (1 - o(1))\delta$.*" (see [6,**?**,**?**,**?**]). Based on that, one can deduce that if $\mathsf{Fold}[f, z] = g + zh$ is δ-close to V for enough values of z, then both g and h are δ'-close from V. The proof of Item (c) consists in exhibiting a codeword which is δ-close from f, thanks to the decomposition (1).

Remark 1. We point out that Item (c) holds because the functions g and h in (1) have *exactly* the same degree. This arises from the crucial fact that the FRI protocol considers only RS code of dimension a power of 2. Eac RS code is defined by polynomials of degree at most an *odd* bound.

Let us observe what happens when f is expected to have degree at most $2d$. The degrees of the functions g and h in the decomposition of f (Item (a)) are respectively $\deg g \leq d$ and $\deg h \leq d - 1$. Therefore, knowing that $g + zh$ is a polynomial of degree $\leq d$ with high probability on z only tells us that both g and h have degree $\leq d$, which is not enough to deduce that f has degree $\leq 2d$ and not $2d + 1$. Moreover words corresponding to a polynomial of degree $2d + 1$ are among the *farthest* words from the RS code of degree $\leq 2d$. One can overcome this obstacle by supposing not only $\deg g, \deg h \leq d$ but also $\deg(\nu h) \leq d$ for a degree-1 polynomial function ν. This implies that $\deg h < d$, hence $\deg f \leq 2d$.

Our IOPP for AG proximity testing. Let $\mathcal{C}$ be a curve defined over a finite field $\mathbb{F}$ and $C = C(\mathcal{C}, \mathcal{P}, D)$ be an AG code. We aim to adapt the three ingredients of the FRI protocol to the AG context.

Group actions and Riemann-Roch spaces. The splitting of the polynomial f into an even and an odd part in Item (a) comes from the action of a multiplicative group of order 2 on its domain. This observation is also true with the actual FRI protocol, in which π is an affine subspace polynomial. This phenomenon occurs in a more general framework: if a group Γ acts on the curve $\mathcal{C}$, its action naturally extends on the functions on $\mathcal{C}$. The representation theory expresses any Riemann-Roch space associated to a Γ-invariant divisor on $\mathcal{C}$ as a sum of vector spaces that Kani [17] proved to arise from some Riemann-Roch spaces on the quotient curve $\mathcal{C}/\Gamma$ through the projection map $\pi : \mathcal{C} \to \mathcal{C}/\Gamma$.

Let us state Kani's result for a cyclic group $\Gamma = \langle \gamma \rangle$ of prime order p. The theorem first states that there exists a function μ on $\mathcal{C}$ such that $\gamma \cdot \mu = \zeta\mu$ where ζ is a primitive p^{th} root of unity. Then, for any divisor D that is Γ-invariant, any function f in the Riemann-Roch space $L_{\mathcal{C}}(D)$ can be uniquely written

$$f = \sum_{j=0}^{p-1} \mu^j f_j \circ \pi \text{ with } f_j \in L_{\mathcal{C}/\Gamma}(E_j) \text{ where } E_j = \left\lfloor \frac{1}{m} \pi_* \left(D + j(\mu) \right) \right\rfloor. \quad (2)$$

Assume that no point of $\mathcal{P}$ is fixed by Γ and set $\mathcal{P}' = \pi(\mathcal{P})$. Polynomial interpolation enables the determination of $f_j(P)$ for any point $P \in \mathcal{P}'$ with exactly p values of f, namely on the set $\pi^{-1}(\{P\})$. This means that the decomposition (2) can be written for any function in $\mathbb{F}^{\mathcal{P}}$, not only for elements of $L_{\mathcal{C}}(D)$.

Folding operator. From the decomposition (2), we aim to define folding operators $\mathbf{Fold}[\cdot, z] : \mathbb{F}^{\mathcal{P}} \to \mathbb{F}^{\mathcal{P}'}$ ($z \in \mathbb{F}$) and a code $C' = C(\mathcal{C}/\Gamma, \mathcal{P}', D')$ such that $\mathbf{Fold}[\cdot, z](C) \subseteq C'$.

In a first approach, one could choose to define the folding operators similarly to the FRI protocol by using the functions f_j in the decomposition (2) of $f \in \mathbb{F}^{\mathcal{P}}$ and setting for $z \in \mathbb{F}$, $\mathbf{Fold}[f, z] = \sum_{j=0}^{p-1} z^j f_j$. Then the code C' has to be associated to a divisor D' on $\mathcal{C}/\Gamma$ such that each Riemann-Roch space $L_{\mathcal{C}/\Gamma}(E_j)$ can be embedded into $L_{\mathcal{C}/\Gamma}(D')$. Note that we would like the rates of C and C' to be roughly equal to prevent the relative minimum distance from dropping. So we need $L_{\mathcal{C}/\Gamma}(D')$ not to be too large compared to the components $L_{\mathcal{C}/\Gamma}(E_j)$. In the best scenario, the space $L_{\mathcal{C}}(D)$ is decomposed in p "copies" of the same Riemann-Roch space, as for RS codes of dimension a power of 2. Unfortunately, it is unlikely that all divisors E_j are the same (or even equivalent) *if $\mathcal{C}$ is not the projective line*. We are then facing a similar issue than in Remark 1 on $\mathbb{P}^1$.

Therefore, such a choice of the folding operators does not guarantee the soundness of our protocol. We thus aim to adapt the idea at the end of Remark 1 to the AG setting. We introduce some *balancing* functions ν_j such that, for every $f_j \in C'$, if the product $\nu_j f_j$ also lies in C', then the function f_j belongs to the desired Riemann-Roch space $L_{\mathcal{C}/\Gamma}(E_j)$. Defining such a balancing function ν_j is tantamount to specify its pole order at the points supporting the divisor D'. The existence of all the functions ν_j thus depends on the *Weierstrass semigroup* of these points (see [16, Section 6.6] for definition) and does not hold for any divisor D'. If such functions exist for a divisor D', we say that D' is *compatible* with D. Finding a convenient divisor D' compatible with a given divisor D is definitely the trickiest part in defining the folding operators properly.

To preserve soundness, we ask for D' to coincide with the divisor E_j with the largest Riemann-Roch space, say $D' = E_0$. If E_0 is D-compatible, we shall add additional terms in the folding operators to take account of the balancing functions. We use more randomness not to double the degree in z, thus avoiding degrading soundness. For $(z_1, z_2) \in \mathbb{F}^2$, we set

$$\mathbf{Fold}[f, (z_1, z_2)] = \sum_{j=0}^{p-1} z_1^j f_j + \sum_{j=1}^{p-1} z_2^j \nu_j f_j.$$

We prove that $\mathbf{Fold}[\cdot, (z_1, z_2)](C) \subseteq C'$, the function $\mathbf{Fold}[f, (z_1, z_2)] \in \mathbb{F}^{\mathcal{P}'}$ can be locally computed from p values of f, and $\mathbf{Fold}[\cdot, (z_1, z_2)]$ preserves the distance to the code.

Sequence of "foldable" AG codes. To iterate the folding process, we assume that the base curve $\mathcal{C}$ is endowed with a *suitable* acting group $\mathcal{G}$ that we decompose into smaller groups to fragment its action and create intermediary quotients

$$\mathcal{C}_0 \xrightarrow{\pi_0} \mathcal{C}_1 \xrightarrow{\pi_1} \mathcal{C}_2 \xrightarrow{\pi_2} \cdots \xrightarrow{\pi_{r-1}} \mathcal{C}_r,$$

where the morphism $\pi_i : \mathcal{C}_i \to \mathcal{C}_{i+1}$ is the quotient map by a cyclic group $\Gamma_i \simeq \mathbb{Z}/p_i\mathbb{Z}$. A condition on the group $\mathcal{G}$ to have such a sequence is the *solvability*.

A code $C = C(\mathcal{C}, \mathcal{P}, D)$ is said to be a *foldable AG code* if we are able to construct a sequence of AG codes $C_i \coloneqq C(\mathcal{C}_i, \mathcal{P}_i, D_i)$ that support a family of randomized folding operators $\mathbf{Fold}[\cdot, \boldsymbol{z}] : \mathbb{F}^{\mathcal{P}_i} \to \mathbb{F}^{\mathcal{P}_{i+1}}$ with the desirable properties for our IOPP (i.e. $\mathbf{Fold}[\cdot, \boldsymbol{z}](C_i) = (C_{i+1})$, local computability, distance preservation to the code). Moreover, to ensure that the last code C_r has sufficiently small length and to obtain an IOPP with sublinear query complexity, we require the size of $\mathcal{G}$ to be greater than $|\mathcal{P}|^e$ for a certain $e \in (0, 1)$.

References

1. Arora, S., Lund, C., Motwani, R., Sudan, M., Szegedy, M.: Proof Verification and the Hardness of Approximation Problems **45**(3), 501–555 (1998). https://doi.org/10.1145/278298.278306, extended version of FOCS'92
2. Arora, S., Safra, S.: Probabilistic Checking of Proofs; A New Characterization of NP. In: 33rd Annual Symposium on Foundations of Computer Science, Pittsburgh, Pennsylvania, USA, 24-27 October 1992. pp. 2–13. IEEE Computer Society (1992)
3. Babai, L.: Trading Group Theory for Randomness. In: Sedgewick, R. (ed.) Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA. pp. 421–429. ACM (1985)
4. Babai, L., Fortnow, L., Levin, L.A., Szegedy, M.: Checking Computations in Polylogarithmic Time. In: Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA. pp. 21–31 (1991). https://doi.org/10.1145/103418.103428
5. Beelen, P., Rosenkilde, J., Solomatov, G.: Fast Encoding of AG Codes over C_{ab} Curves (2020)
6. Ben-Sasson, E., Bentov, I., Horesh, Y., Riabzev, M.: Fast Reed-Solomon Interactive Oracle Proofs of Proximity. In: 45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic. pp. 14:1–14:17 (2018)
7. Ben-Sasson, E., Bentov, I., Horesh, Y., Riabzev, M.: Scalable Zero Knowledge with No Trusted Setup. In: Boldyreva, A., Micciancio, D. (eds.) Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part III. Lecture Notes in Computer Science, vol. 11694, pp. 701–732. Springer (2019)
8. Ben-Sasson, E., Chiesa, A., Gabizon, A., Riabzev, M., Spooner, N.: Interactive Oracle Proofs with Constant Rate and Query Complexity. In: 44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland. pp. 40:1–40:15 (2017)
9. Ben-Sasson, E., Chiesa, A., Goldberg, L., Gur, T., Riabzev, M., Spooner, N.: Linear-Size Constant-Query IOPs for Delegating Computation. In: Hofheinz, D., Rosen, A. (eds.) Theory of Cryptography - 17th International Conference, TCC 2019, Nuremberg, Germany, December 1-5, 2019, Proceedings, Part II. Lecture Notes in Computer Science, vol. 11892, pp. 494–521. Springer (2019)

10. Ben-Sasson, E., Chiesa, A., Spooner, N.: Interactive Oracle Proofs. In: Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part II. pp. 31–60 (2016)
11. Ben-Sasson, E., Kaplan, Y., Kopparty, S., Meir, O., Stichtenoth, H.: Constant Rate PCPs for Circuit-SAT with Sublinear Query Complexity. In: 54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA. pp. 320–329. IEEE Computer Society (2013)
12. Ben-Sasson, E., Kopparty, S., Saraf, S.: Worst-Case to Average Case Reductions for the Distance to a Code. In: 33rd Computational Complexity Conference, CCC 2018, June 22-24, 2018, San Diego, CA, USA. pp. 24:1–24:23 (2018)
13. Ben-Sasson, E., Sudan, M.: Short PCPs with Polylog Query Complexity. SIAM J. Comput. **38**(2), 551–607 (2008)
14. Goldwasser, S., Micali, S., Rackoff, C.: The Knowledge Complexity of Interactive Proof-Systems (Extended Abstract). In: Sedgewick, R. (ed.) Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA. pp. 291–304. ACM (1985)
15. Goppa, V.D.: Codes associated with divisors. Problemy Peredachi Informatsii **13**(1), 33–39 (1977)
16. Hirschfeld, J.W.P., Korchmáros, G., Torres, F.: Algebraic Curves over a Finite Field. Princeton University Press, Princeton (25 Mar 2013). https://doi.org/https://doi.org/10.1515/9781400847419
17. Kani, E.: The Galois-module structure of the space of holomorphic differentials of a curve. Journal für die reine und angewandte Mathematik **367**, 187–206 (1986)
18. Kattis, A., Panarin, K., Vlasov, A.: RedShift: Transparent SNARKs from List Polynomial Commitment IOPs. IACR Cryptol. ePrint Arch. **2019**, 1400 (2019), `https://eprint.iacr.org/2019/1400`
19. Kilian, J.: A Note on Efficient Zero-Knowledge Proofs and Arguments (Extended Abstract). In: Kosaraju, S.R., Fellows, M., Wigderson, A., Ellis, J.A. (eds.) Proceedings of the 24th Annual ACM Symposium on Theory of Computing, May 4-6, 1992, Victoria, British Columbia, Canada. pp. 723–732. ACM (1992)
20. Lund, C., Fortnow, L., Karloff, H.J., Nisan, N.: Algebraic Methods for Interactive Proof Systems. In: 31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Volume I. pp. 2–10. IEEE Computer Society (1990)
21. Meir, O.: IP = PSPACE Using Error-Correcting Codes. SIAM J. Comput. **42**(1), 380–403 (2013)
22. Micali, S.: Computationally-Sound Proofs. In: Makowsky, J.A., Ravve, E.V. (eds.) Proceedings of the Annual European Summer Meeting of the Association of Symbolic Logic, Logic Colloquium 1995, Haifa, Israel, August 9-18, 1995. Lecture Notes in Logic, vol. 11, pp. 214–268. Springer (1995)
23. Mie, T.: Short PCPPs Verifiable in Polylogarithmic Time with O(1) Queries. Annals of Mathematics and Artificial Intelligence **56**(3–4), 313–338 (Aug 2009)
24. Ron-Zewi, N., Rothblum, R.D.: Local Proofs Approaching the Witness Length [extended abstract]. In: 61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020. pp. 846–857. IEEE (2020)