Electronic Colloquium on Computational Complexity, Revision 2 of Report No. 165 (2020)

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
[2]Inria

February 15, 2021

Abstract

In this work, we initiate the study of proximity testing to Algebraic Geometry (AG) codes. An AG code $C = C(\mathcal{C}, \mathcal{P}, D)$ is a vector space associated to evaluations on $\mathcal{P}$ of functions in the Riemann-Roch space $L_{\mathcal{C}}(D)$. 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. AG codes are good candidates to construct *short* proof systems, but there exists no efficient proximity tests for them. We aim to fill this gap.

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 [BBHR18]. 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 [Kan86]. Under some hypotheses, a proximity test to C can thus be 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 this type of curves, we manage to extend our generic construction to reach a strictly linear proving time and a strictly logarithmic verification time.

*sarah.bordage@lix.polytechnique.fr
†jade.nardi@inria.fr

ISSN 1433-8092

1 Introduction

Under the generic term of *arithmetization* ([LFKN90]), algebraic techniques for constructing proof systems using properties of low-degree polynomials have emerged from the study of interactive proofs (IPs, [GMR85]). Arithmetization techniques have been enhanced and fruitfully applied to other broad families of proof systems since then, including probabilistically checkable proofs (PCPs, [BFLS91, AS92, ALM+98]). To construct a proof system for a non-deterministic relation $\mathcal{R}$, arithmetization transforms any instance-witness pair (x, w) into a word that belongs to a certain error-correcting code C if $(x, w) \in \mathcal{R}$, and is very far from C otherwise.

Since the seminal works of Kilian [Kil92] and Micali [Mic95], 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, IPs and IPCPs ([KR08]).

Considering for the first time univariate polynomials instead of multivariate ones, [BS08, Din07] 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, including [AHIV17], [BBHR19], [BCR+19], [BCG+19], [KPV19], [COS20] – to mention only the most recent ones. At some point, aforementioned sublinear arguments require a proximity test to RS codes. As a solution, one can use a prover-efficient Reed-Solomon IOP of Proximity, which is an interactive variant of PCP of Proximity introduced by [BCG+17]. The state-of-the-art IOPP for RS codes is known as the FRI protocol ([BBHR18], further improved in [BKS18], [BGKS20], [BCI+20]).

In 2013, [BKK+13] construct a PCP with linear proof length and sublinear query complexity for boolean circuit satisfiability by relying on AG codes. More precisely, 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 construction. By using AG codes, the authors of [BKK+13] reduce the field size to a constant, which avoids a logarithmic blowup in proof bit-length (occuring e.g. in [BS08] when using univariate polynomials of degree m to encode binary strings of length m). In [BKK+13], the authors point out that they are not able to apply proof composition ([AS92]) 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 [BKK+13], [BCG+17] construct an interactive oracle proof (IOP, [BCS16, RRR16]) for boolean circuit satisfiability with linear proof length and constant query complexity. However, prover and verifier complexities are still super-linear. The IOP of [BCG+17] invokes the sumcheck protocol [LFKN90] 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 [Mie09] to test proximity to the tensored code. Both constructions benefit from the use of AG codes to get constant size alphabet and linear proof bit-lengths. However, prover and verifier running times prevent them to be implemented for verifying meaningful computations.

A recent work of [RR20] constructs an IOPP for any deterministic language which can be decided in time poly(n) and space $n^{o(1)}$, with constant round, constant query complexity and linear proof length. However, prover's running time is poly(n). We exhibit families of AG codes for which one can construct a proximity test with linear proving time and logarithmic verification.

The FRI protocol for RS proximity testing admits linear prover time, logarithmic verifier time and logarithmic query complexity. A natural question is whether one can construct an IOPP targeting AG codes with similar efficiency parameters. Indeed, AG codes [Gop77], 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 [Mei13], namely the component-wise multiplication of two codewords results in codewords in a code whose minimum distance is still good. This multiplication property actually emulates multiplication of low-degree polynomials. AG codes not only feature this multiplication property but may also have arbitrary large length given a fixed finite field $\mathbb{F}$, unlike RS codes. For concrete efficiency, complexity measures such as proof length, query complexity, prover time and verifier time are closely examined and reducing the size of the alphabet has a direct impact on the binary complexities.

Keeping applications to proof systems in mind, it can be noticed that the running time of the prover is bounded from below by the time needed to encode codewords during arithmetization. Prover complexity is actually the main bottleneck in deploying zero-knowledge proof systems for large computations. In this direction, one-point AG codes on some family of curves, including Kummer type curves, are especially appealing. For instance, they have recently been shown to have subquadratic encoding [BRS20].

We dedicate a part of our study to the particular case of AG codes on Kummer type curves. To encourage the search for suitable families of AG codes, we study generic conditions that are conducive to 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. A first step in this direction could be to use the IOP of [BCG$^+$17] as a starting point, or to find an analogue of the univariate sumcheck protocol introduced by [BCR$^+$19].

1.1 Definition of an IOPP for a code C

Let C be an evaluation code with evaluation domain S of size n and alphabet Σ (i.e., $C \subseteq \Sigma^S$). Throughout this paper, 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 from C, denoted $\Delta(u, C)$, is the minimal distance between u and a codeword of C. For $u \in \Sigma^S$, 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 $f : S \to \Sigma$, distinguish between the case where $f \in C$ and f is δ-far from C. In this paper, we focus on the case where C is an AG code. We recall that 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 has demonstrated to be particularly promising for the design of proof systems in the past few years.

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 the purported codeword 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. Verifier's queries to the prover's messages are generated by public randomness and performed after the end of the interaction with the prover. Thus, such an IOPP is in particular a *public-coin* protocol (or Arthur-Merlin [Bab85]).

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 $\delta \coloneqq \Delta(f, C) > 0$ 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 define 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

Let us review the main contributions of this paper.

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 $C_i \coloneqq C(\mathcal{C}_i, \mathcal{P}_i, D_i)$ 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. The above sequence of curve is designed so that C_{i+1} arises as the quotient of the curve C_i by a cyclic group $\mathbb{Z}/p_i\mathbb{Z}$ under the quotient map π_i. We show that such a procedure is possible if a large enough solvable group $\mathcal{G}$ acts on the curve $\mathcal{C}_0$ and under some hypotheses on the divisor D_0 overviewed in Section 1.3.2 and detailed in Section 3.1. A code fulfilling all the conditions we require will be called *foldable*.

Next, we construct an IOPP for testing proximity to any foldable AG code $C(\mathcal{C}_0, \mathcal{P}_0, D_0)$ of blocklength n with linear proof length, sublinear query complexity and constant soundness. Efficiency parameters of this protocol, called AG-IOPP, are captured by the following theorem, which is proved in Theorem 4.6.

Theorem 1.1 (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

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

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 of the protocol.

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 $(C_i)_{0\leq i\leq s}$ such that the code C_s is a *Reed-Solomon 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). Theorem 1.2 is thus an improvement over Theorem 1.1 for the special case of Kummer curve.

Theorem 1.2 (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*

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

Prover complexity is $\mathsf{t_p}(n) = O(n)$ and verifier decision complexity is $\mathsf{t_v}(n) = O(\log n)$.

It is worth noting that the Hermitian curve defined over $\mathbb{F}_{q^2}$ by $y^{q+1} = x^q + x$ satisfies the hypotheses of the previous theorem. It is well known to be *maximal*, i.e. it has the maximum number of rational points with respect to its geometry. We thus provide family of codes much longer than Reed-Solomon codes that are endowed with a proximity test as efficient as the FRI protocol.

Remark 1.3 (On the concrete size of non-interactive arguments). Our public-coin IOPP can be compiled into a non-interactive argument via [BCS16]'s transformation, which consists in first applying Kilian's compiler [Kil92] at each round of interaction to commit to oracle $f^{(i)}$ using a Merkle hash tree represented by its root $\mathsf{rt}^{(i)}$. As in FRI, each set of p_i points in $\mathcal{P}_i$ that have the same image by π_i is represented by a single leaf of the Merkle tree $\mathsf{rt}^{(i)}$. Then, following

Micali's scheme [Mic95], such an interactive argument is turned into a non-interactive one by asking the prover to simulate the verifier's random messages [BCS16]. Each simulated-query answer to oracle $f^{(i)}$ is accompanied with a Merkle proof of size $\log\left(\frac{1}{p_i}|\mathcal{P}_i|\right)$. The resulting communication complexity is linear in $q(n)$ and logarithmic in both $l(n)$ and the field size $|\mathbb{F}|$ (see [BCS16] for further details).

1.3 Technical overview

Our IOPP construction relies on the generalization of the FRI protocol to AG codes. Let us first recall some ideas behind the construction of FRI protocol (see e.g. [BKS18] for a detailed presentation). Then we shall describe how we tailor these ideas and which difficulties arise to construct our IOPP.

1.3.1 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 $\mathsf{RS}[\mathbb{F},\mathcal{P},\rho] \coloneqq \left\{f \in \mathbb{F}^{\mathcal{P}} \mid \deg f < \rho|\mathcal{P}|\right\}$ 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 $\mathbb{F}^{\mathcal{P}'}$. We present in a simplified way three key ingredients that enable the FRI protocol to work.

1. *Splitting of polynomials.* The FRI protocol is based on the following observation: 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
$$f(x) = g\left(x^2\right) + x \cdot h\left(x^2\right). \tag{1}$$
One may view such a decomposition as the result of the splitting of the space of polynomials of degree less than ρn into two copies of the space of polynomials of degree less than $\rho n/2$.

2. *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 $\pi(\mathcal{P}) = \mathcal{P}'$. Moreover, $|\mathcal{P}'| = |\mathcal{P}|/2$. 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), define a *folding operator* $\mathsf{Fold}[\cdot, z] : \mathbb{F}^{\mathcal{P}} \to \mathbb{F}^{\mathcal{P}'}$ for any $z \in \mathbb{F}$ as follows:
$$\mathsf{Fold}[f, z] \coloneqq g + zh.$$
   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 $\mathsf{Fold}[\cdot, z]$ maps $\mathsf{RS}[\mathbb{F},\mathcal{P},\rho]$ into $\mathsf{RS}[\mathbb{F},\mathcal{P}',\rho]$.

3. *Distance preservation after folding.* Except with small probability over z, we have that if $\Delta(f, \mathsf{RS}[\mathbb{F},\mathcal{P},\rho]) \geq \delta$, then
$$\Delta\left(\mathsf{Fold}[f, z], \mathsf{RS}\left[\mathbb{F},\mathcal{P}',\rho\right]\right) \geq (1 - o(1))\delta.$$

The protocol then 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 equal to $\mathsf{Fold}[f, z] : \mathcal{P}' \to \mathbb{F}$. 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 evaluation domain. This induces a sequence of Reed-Solomon codes of strictly decreasing length. The code rate remains unchanged, and so does the relative minimum distance. The final test consists in testing that $f^{(r)}$ belongs to the last RS code.

Perfect completeness follows from Item 2. Prover and verifier efficiencies of the FRI protocol come from the possibility of determining any value of $\textsf{\textbf{Fold}}\,[f, z]$ 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 notably on Item 3. 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 [BBHR18, BKS18, BGKS20, BCI+20]). Based on that, one can deduce that if $\textsf{\textbf{Fold}}\,[f, z] = g + zh$ is δ-close to V for enough values of z, then both g and h are δ'-close from V. The idea of the proof of Item 3 is to exhibit a codeword which is δ-close from f, based on the decomposition of Item 1.

Remark 1.4. We point out that Item 3 holds because the functions g and h appearing in the decomposition (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. This means that the RS code is defined by polynomials of degree at most an *odd* bound.

Let us give glimpse of what happens when f is expected to have degree at most an even integer, say $2d$. The degrees of the functions g and h appearing in the decomposition of f (Item 1) are respectively $\deg g \leq d$ and $\deg h \leq d - 1$. Therefore, if $\deg f \leq 2d$, then $g + zh$ corresponds to a polynomial of degree $\leq d$. However, knowing that $g + zh$ is a polynomial of degree $\leq d$ with high probability on z only tells us that both g and h are of degree $\leq d$, which is not enough to deduce that f has degree $\leq 2d$ and not $2d+1$. It is worth noting that words corresponding to a polynomial of degree $2d + 1$ are among the *farthest* words from the RS code of degree $\leq 2d$. In the univariate case, 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 $\deg f \leq 2d$.

1.3.2 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 1 comes from the action of a multiplicative group of order 2 on the evaluation set $\mathcal{P}$. This observation is also true with the actual FRI protocol, which sets π to be an affine subspace polynomial. This phenomenon occurs in a more general framework. As soon as 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 [Kan86] 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

$$f = \sum_{j=0}^{p-1} \mu^j f_j \circ \pi \text{ with } f_j \in L_{\mathcal{C}/\Gamma}(E_j), \tag{2}$$

where the divisors E_j on the quotient curve are explicitly expressed in terms of the divisor D, the projection π and the function μ (see Theorem 2.2 for details).

Assume that no point of $\mathcal{P}$ is fixed by Γ, i.e. for every $P \in \mathcal{P}$ and $j \in \{1, \dots, p-1\}$, $\gamma^j \cdot P \neq P$. 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) above, we want to define a family of folding operators $(\textsf{Fold}[\cdot, z])_{z\in\mathbb{F}}$ from $\mathbb{F}^{\mathcal{P}}$ to $\mathbb{F}^{\mathcal{P}'}$ and a code $C' = C(\mathcal{C}/\Gamma, \mathcal{P}', D')$ such that $\textsf{Fold}[\cdot, z](C) \subseteq C'$.

In a first approach, one could choose to define the folding operators similarly to the FRI protocol by setting for $z \in \mathbb{F}$, $\textsf{Fold}[f, z] = \sum_{j=0}^{p-1} z^j f_j$ where the functions f_j come from the decomposition (2) of $f \in \mathbb{F}^{\mathcal{P}}$. With this definition, 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. In other words, we need $L_{\mathcal{C}/\Gamma}(D')$ to be not too large with respect to the components $L_{\mathcal{C}/\Gamma}(E_j)$. The best scenario is when the divisor D yields a decomposition of $L_{\mathcal{C}}(D)$ as p "copies" of the same Riemann-Roch space, as it is the case with Reed-Solomon codes of dimension a power of 2. Unfortunately, to the best of our knowledge, it is unlikely that all divisors E_j involved in the decomposition of f (Equation (2)) are the same (or even equivalent) *if $\mathcal{C}$ is not the projective line.* We are then facing an issue analogous to the one described in Remark 1.4 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 1.4 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 [HKT13, 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. Without loss of generality, we assume that $D' = E_0$. If E_0 is D-compatible, we shall embed additional terms in the folding operators to take account of the balancing functions. We shall use more randomness so as not to double the degree in z to avoid a loss in soundness. For $(z_1, z_2) \in \mathbb{F}^2$, we set

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

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

Sequence of "foldable" AG codes. With the goal of iterating the folding process in mind, 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* (Definition 3.4) 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 $\mathsf{Fold}[\cdot, \boldsymbol{z}] : \mathbb{F}^{\mathcal{P}_i} \to \mathbb{F}^{\mathcal{P}_{i+1}}$ with the desirable properties for our IOPP (i.e. $\mathsf{Fold}[\cdot, \boldsymbol{z}](C_i) = (C_{i+1})$, 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 $e \in (0, 1)$. Details are provided in Section 3.

1.4 Future works: other families of foldable AG codes

To take maximal advantage from working on AG codes rather than Reed-Solomon codes, we are inclined to apply the AG-IOPP to very long codes from *maximal curves*. Many works has been carried out on codes from the Hermitian curve [YB92, SG93, LSH97, Ren04, Mat05], and other maximal curves [CRL90, FG10, BMZ18a, BMZ18b]. One direction towards new efficient foldable codes is investigating these codes to study their foldability. This requires a large enough solvable subgroup $\mathcal{G}$ of the automorphism groups of maximal curves, which are very rich and comprehensively described in the literature [GMP12, Mon20]. Once the group $\mathcal{G}$ is chosen, the most intricate point to determine some conditions on the divisor D to construct an interesting sequence of divisors that fulfils the compatibility constraints. This is closely related to Weierstrass semigroups, as illustrated in Example 3.11 for the Kummer case, which have been carefully investigated on maximal curves in aforementioned works to display codes with excellent parameters.

Another promising research direction is exploiting the similarity of settings between foldable codes and (asymptotically good) *towers of curves*. We recall that a tower of curves consists of an infinite sequence of curves

$$\mathcal{C}_0 \leftarrow \mathcal{C}_1 \leftarrow \ \ldots \leftarrow \mathcal{C}_n \leftarrow \ldots$$

such that the number of rational points of the n^{th} curve tends to infinity as n tends to infinity. They play a prominent role in the history of AG codes as they define codes with outstanding length and correction capacity [TVZ82, BBGS14]. In the AG-IOPP, we start from a proximity problem on a code C on a curve $\mathcal{C} = \mathcal{C}_0$ and we create a sequence of curves to simplify this problem. By examining the Galois groups of the extensions in a given tower [BB09, BB11], we could process backwards: if one wants to test proximity to an AG-code from one of the curves $\mathcal{C}_n$, we could fold all the way down the tower to the curve $\mathcal{C}_0$.

Finally, we chose here to ask each intermediary quotient to be cyclic essentially to split the action of $\mathcal{G}$ as much as possible and to easily ensure an efficient local computability at each step. However, Kani's result [Kan86], from which we designed the folding operators, does not only hold for cyclic groups. If local computability can be preserved in a more general setting - e.g. by multivariate polynomial interpolation -, the hypotheses of the AG-IOPP could be relaxed to broaden the variety of foldable codes.

1.5 Organization of the paper

In Section 2, we gather some basic notions and definitions around AG codes. Section 3 establishes a valid framework for constructing AG-IOPP. We define *foldable* AG codes and study the example of Kummer type curves which provide a setting checking all the aforementioned requirements. Our IOPP construction for foldable AG codes is presented in Section 4 and a specialized variant for

Kummer curves is discussed in Section 5. Properties of those IOPPs are respectively stipulated in Theorem 4.6 and Theorem 5.2. Section 6 is quite technical and is dedicated to soundness analysis.

2 Preliminaries

2.1 Definitions and notations

We start with some reminders on important terms and notations related to the theory of AG codes. We refer readers to [TVN07, Sti93] for further details on these notions. We will always use $\mathbb{F}$ to denote a finite field.

Functions and divisors on algebraic curves. Let $\mathcal{C}$ be an algebraic curve defined over a field $\mathbb{F}$. We denote by $\mathcal{C}(\mathbb{F})$ the set of its $\mathbb{F}$-rational points and $\mathrm{Aut}(\mathcal{C})$ its automorphism group.

A *divisor* D on $\mathcal{C}$ is a formal sum of points $D = \sum n_P P$. We say that the divisor D is *effective* if $n_P \geq 0$ for every point P. The *degree* of D equals $\deg D := \sum n_P$. The *support* of D $\mathrm{Supp}(D)$ is the set of points P for which the coefficient n_P is non zero.

The set of divisors on the curve $\mathcal{C}$ forms an additive group, denoted by $\mathrm{Div}(\mathcal{C})$. It is endowed with a partial order relation $\leq$ such that $D \leq D'$ if $D' - D$ is effective. An element f of the function field $\mathbb{F}(\mathcal{C})$ of the curve $\mathcal{C}$ defines a divisor

$$(f) = \sum_P v_P(f)P$$

where $v_P(f)$ is the valuation of the function f at the point P. We denote by $(f)_0$ and $(f)_\infty$ the effective divisors such that $(f) = (f)_0 - (f)_\infty$. They correspond to the loci of zeroes and poles respectively.

Let $\phi : \mathcal{C} \to \mathcal{C}'$ be a map between two algebraic curves. It induces a *pull-back* map $\phi^* : \mathbb{F}(\mathcal{C}') \to \mathbb{F}(\mathcal{C})$ defined by $\phi^* f = f \circ \phi$ for $f \in \mathbb{F}(\mathcal{C}')$. For $D = \sum_P n_p P \in \mathrm{Div}(\mathcal{C})$, the *push-forward* of D is the divisor on $\mathcal{C}'$ defined by $\pi_*(D) = \sum_P n_P \phi(P)$.

The *Riemann-Roch space* of a divisor $D \in \mathrm{Div}(\mathcal{C})$ is the vector space defined by

$$L_{\mathcal{C}}(D) = \{f \in \mathbb{F}(\mathcal{C}) \mid (f) + D \geq 0\} \cup \{0\}.$$

The subscript specifying the curve in $L_{\mathcal{C}}(D)$ is omitted when it is clear from the context. If $D' \leq D'$, then $L_{\mathcal{C}}(D) \subseteq L_{\mathcal{C}}(D')$.

As usual, given a real number x, $\lfloor x \rfloor$ denotes the biggest integer less than or equal to x and $\lceil x \rceil$ the smallest integer bigger than or equal to x.

Definition 2.1. *Let $D = \sum n_P P \in \mathrm{Div}(\mathcal{C})$. For any positive integer n, we denote by $\left\lfloor \frac{1}{n} D \right\rfloor \in \mathrm{Div}(\mathcal{C})$ the divisor defined by*

$$\left\lfloor \frac{1}{n} D \right\rfloor := \sum \left\lfloor \frac{n_P}{n} \right\rfloor P.$$

Algebraic geometry codes. Take $D \in \mathrm{Div}(\mathcal{C})$ and $\mathcal{P} \subset \mathcal{C}(\mathbb{F})$ of size $n := |\mathcal{P}|$ such that $\mathrm{Supp}(D) \cap \mathcal{P} = \emptyset$. The *Algebraic Geometry (AG) code* $C = C(\mathcal{C}, \mathcal{P}, D)$ is defined as the image under the evaluation map

$$\mathrm{ev} : L(D) \to \mathbb{F}^n.$$

The integer n is called the *length* of C. The *dimension* of C is defined as its dimension as $\mathbb{F}$-vector space. We denote by $\Delta(C)$ the relative minimum distance of C, i.e.

$$\Delta(C) = \min \left\{ \Delta(c, c') \mid c, c' \in C \text{ and } c \neq c' \right\}.$$

In particular, AG codes on $\mathcal{C} = \mathbb{P}^1$ correspond to Reed-Solomon codes.

Throughout this paper, the term *code* will refer to a *linear code*, i.e. a linear subspace of $\mathbb{F}^n$, where n is the length of the code.

The AG code C is said to be *one-point* if the support of D consists in a single point. By the Riemann-Roch theorem, if $\deg D \geq 2g - 1$ where g is the genus of the curve $\mathcal{C}$, then $\dim L_{\mathcal{C}}(D) = \deg D - g + 1$. Moreover, if $\deg D < n$, the evaluation map is injective and the Riemann-Roch theorem gives the dimension of the associated AG code. In this case, the minimum distance is bounded from below by $n - \deg D$.

The divisor D will always be chosen so that the map ev is injective. Therefore, the elements of $\mathbb{F}^n$ will be regarded as functions in $\mathbb{F}^{\mathcal{P}}$ and elements of C simply as functions in the Riemann-Roch space $L(D)$.

Group and action. A finite group $\mathcal{G}$ is said to be *solvable* if there exists a sequence of subgroups of $\mathcal{G}$

$$\mathcal{G} = \mathcal{G}_0 \rhd \mathcal{G}_1 \rhd \cdots \rhd \mathcal{G}_r = 1,$$

such that $\mathcal{G}_{i+1}$ is a normal subgroup of $\mathcal{G}_i$ and each factor group $\mathcal{G}_i/\mathcal{G}_{i+1}$ is a cyclic group of prime order. Such a sequence is called a *composition series.* If $\mathcal{G}$ is solvable, its cardinality equals the product of the sizes of the factor groups.

Let $\mathcal{C}$ be an algebraic curve. A group Γ is said to *act on the curve* $\mathcal{C}$ if Γ is a subgroup of the automorphism group $\mathrm{Aut}(\mathcal{C})$. The *stabilizer* of a point $P \in \mathcal{C}$ is the subgroup

$$\Gamma_P = \{\gamma \in \Gamma \mid \gamma \cdot P = P\} \subset \Gamma.$$

A divisor $D = \sum_P n_P P \in \mathrm{Div}(\mathcal{C})$ is said to be Γ-*invariant* is $n_P = n_{\gamma \cdot P}$ for all $P \in \mathcal{C}$ and $\gamma \in \Gamma$.

The action of Γ on $\mathcal{C}$ gives a projection $\pi : \mathcal{C} \to \mathcal{C}/\Gamma$ onto the quotient curve $\mathcal{C}/\Gamma$. A point $Q \in \mathcal{C}/\Gamma$ is called a *ramification point* is the number of preimages of Q by π is not equal to $|\Gamma|$. Equivalently, Q is a ramification point if one of its preimages has a non trivial stabilizer.

2.2 Splitting Riemann-Roch spaces according to a cyclic group of automorphisms

Let $\mathcal{X}$ be a smooth irreducible curve over a field $\mathbb{F}$ and let Γ be a cyclic group of order m generated by an element γ. Assume that m and the characteristic of $\mathbb{F}$ are coprime and consider ζ a primitive m^{th} root of unity.

Set $\mathcal{Y} := \mathcal{X}/\Gamma$ and $\pi : \mathcal{X} \to \mathcal{Y}$ be the canonical projection morphism.

Fix a Γ-invariant divisor $D \in \mathrm{Div}(\mathcal{X})$. We want to exhibit a relation between the Riemann-Roch space $L_{\mathcal{X}}(D)$ and some Riemann-Roch spaces on $\mathcal{Y}$. The group Γ acts on the vector space $L_{\mathcal{X}}(D)$ via $\gamma \cdot f = f \circ \gamma$. By the representation theory,

$$L_{\mathcal{X}}(D) = \bigoplus_{j=0}^{m-1} L_{\mathcal{X}}(D)_j,$$

where $L_{\mathcal{X}}(D)_j \coloneqq \{g \in L_{\mathcal{X}}(D) \mid \gamma \cdot g = \zeta^j g\}$.

One of the key ingredients of this section is a theorem due to Kani [Kan86], which we reformulate here in the case where Γ is cyclic.

Theorem 2.2 ([Kan86]). *Assume that $\Gamma = \langle\gamma\rangle$ is cyclic of order m, coprime with $|\mathbb{F}|$.*

- *There exists a function $\mu \in \mathbb{F}(\mathcal{X})$ such that $\gamma \cdot \mu = \zeta\mu$,*
- $L_{\mathcal{X}}(D)_j \simeq \mu^j \pi^* \left(L_{\mathcal{Y}} \left(\left\lfloor \frac{1}{m} \pi_* \left(D + j(\mu) \right) \right\rfloor \right) \right)$,

where the floor function of a divisor is given in Definition 2.1.

One can reformulate the second item of Theorem 2.2 as follows: for every $f \in L(D)$, there exist m functions $f_j \in L(E_j)$ such that $f = \sum_{j=1}^m \mu^j f_j \circ \pi$, where $E_j = \left\lfloor \frac{1}{m} \pi_* \left(D + j(\mu) \right) \right\rfloor$.

Remark 2.3. When dealing with univariate polynomials, the theorem of Kani [Kan86] is equivalent to the splitting of a polynomial into an even part and an odd part, which plays a crucial role in the FRI protocol. It also specifies the degree of each part.

The set of polynomials of degree (less than or equal to) d is isomorphic to the Riemann-Roch space $L_{\mathbb{P}^1}(dP_\infty)$ on $\mathbb{P}^1$ where the point P_∞ can be chosen as $P_\infty = [0:1]$.

Now, let us consider the involution γ defined by $\gamma : [X_0 : X_1] \mapsto [-X_0 : X_1]$. It generates a group isomorphic to $\mathbb{Z}/2\mathbb{Z}$ and the quotient of $\mathbb{P}^1$ by this group is obtained as the image by $\pi : [X_0 : X_1] \mapsto [X_0^2 : X_1^2]$.

The divisor $D \coloneqq dP_\infty$ is invariant under γ and the function $x = \frac{X_0}{X_1}$ satisfies the first item of Theorem 2.2 and $(x) = P_0 - P_\infty$ where $P_0 = [1:0]$. Noticing that $\pi_*(P_\infty) = P_\infty$ and $\pi_*(P_0) = P_0$, we get

$$\left\lfloor \frac{1}{2} \pi_*(D + (x)) \right\rfloor = \left\lfloor \frac{1}{2}((d-1)P_\infty + P_0) \right\rfloor = \left\lfloor \frac{d-1}{2} \right\rfloor P_\infty$$

and the Riemman-Roch space $L_{\mathbb{P}^1}(dP_\infty)$ is split into two parts:

$$L_{\mathbb{P}^1}(dP_\infty) = \pi^* L_{\mathbb{P}^1}\left(\left\lfloor \frac{d}{2} \right\rfloor P_\infty \right) + x\pi^* L_{\mathbb{P}^1}\left(\left\lfloor \frac{d-1}{2} \right\rfloor P_\infty \right).$$

We recover the decomposition of a polynomial of degree d into an even part of and an odd one of respective degrees $\left\lfloor \frac{d}{2} \right\rfloor$ and $\left\lfloor \frac{d-1}{2} \right\rfloor$.

Broadly speaking, Theorem 2.2 expresses a Riemann-Roch space on a curve as sum of some Riemann-Roch spaces on the quotient curve that depend on the zeroes and poles of the function μ, including the ramification points of π according to the following lemma.

Lemma 2.4. *Assume that $\Gamma = \langle\gamma\rangle$ is a cyclic group of order m. Let P be a point of X whose stabilizer Γ_P is non trivial. Then $P \in \operatorname{Supp}(\mu)$.*

Proof. By hypothesis, there exists $j \in \{1, \ldots, m-1\}$ such that $\gamma^j \in \Gamma_P$. Then

$$\begin{aligned} (\gamma^j \cdot \mu)(P) &= \zeta^j \mu(P) && \text{by definition of } \mu \text{ in Th. 2.2,} \\ &= \mu(P) && \text{because } \gamma^j \in \Gamma_P. \end{aligned}$$

Since $\zeta^j \neq 1$, the point P is either a pole or a zero of μ. ◻

Remark 2.5. In Remark 2.3 the ramification points of π are precisely P_0 and P_∞, which are invariant under γ. Both points are zero or poles of x. Moreover, one can easily see that any suitable choice for μ would be an odd polynomial of x.

3 Foldable AG codes

In this section, we display a workable setting for the construction of an IOPP system (P, V) to test whether a given function $f : \mathcal{P} \to \mathbb{F}$ is close to the evaluation of a function in a given Riemann-Roch space. As the idea is to iteratively reduce the problem of testing proximity to $C(\mathcal{C}, \mathcal{P}, D)$ to testing proximity to a smaller AG code, we introduce a sequence of suitable AG codes of decreasing length.

3.1 Valid setting of AG codes for IOPP

3.1.1 Sequence of curves

Fix a curve $\mathcal{C}$ defined over $\mathbb{F}$ and a finite solvable group $\mathcal{G} \subseteq \mathrm{Aut}(\mathcal{C})$ whose order is coprime with the characteristic of $\mathbb{F}$. By solvability of $\mathcal{G}$, there exists a *composition series*, i.e. a sequence of subgroups of $\mathcal{G}$

$$\mathcal{G} = \mathcal{G}_0 \rhd \mathcal{G}_1 \rhd \cdots \rhd \mathcal{G}_r = 1, \tag{3}$$

such that $\mathcal{G}_{i+1}$ is a normal subgroup of $\mathcal{G}_i$ and the *factor group* $\Gamma_i := \mathcal{G}_i/\mathcal{G}_{i+1} \simeq \mathbb{Z}/p_i\mathbb{Z}$ is a cyclic group of prime order p_i. Moreover the cardinality of $\mathcal{G}$ equals $|\mathcal{G}| = \prod_{i=0}^{r-1} p_i$. We say that r is the *length* of the composition series of $\mathcal{G}$.

A sequence like (3) may not be unique but another sequence of this type would have the same length and the same set of factor groups, up to permutation, by the Jordan–Hölder theorem.

The group Γ_0 acts on $\mathcal{C}_0 := \mathcal{C}$, as a quotient of $\mathcal{G}$. We thus define the quotient curve $\mathcal{C}_1 := \mathcal{C}_0/\Gamma_0$. The group Γ_1 is constant on the orbits under Γ_0. Repeating the process for every $i \in \{0, \ldots, r-1\}$ defines a sequence of curves recursively as follows:

$$\mathcal{C}_0 := \mathcal{C} \text{ and } \mathcal{C}_{i+1} = \mathcal{C}_i/\Gamma_i.$$

We denote by $\pi_i : \mathcal{C}_i \to \mathcal{C}_{i+1}$ the canonical projection modulo the action of Γ_i.

$$\begin{tikzcd}
\mathcal{C}_0 \arrow[loop above, "\Gamma_0"] \arrow[r, two heads, "\pi_0"] & \mathcal{C}_1 \arrow[loop above, "\Gamma_1"] \arrow[r, two heads, "\pi_1"] & \cdots \arrow[r, two heads, "\pi_i"] & \mathcal{C}_i \arrow[loop above, "\Gamma_i"] \arrow[r, two heads, "\pi_{i+1}"] & \mathcal{C}_{i+1} \arrow[loop above, "\Gamma_{i+1}"] \arrow[r, two heads] & \cdots \arrow[r, two heads, "\pi_{r-1}"] & \mathcal{C}_r
\end{tikzcd} \tag{4}$$

Even if the sequence of curves (4) depends on the composition series (3) of $\mathcal{G}$, the last curve $\mathcal{C}_r$ is always isomorphic to the quotient $\mathcal{C}/\mathcal{G}$.

Definition 3.1. *A sequence of curves constructed as above will be called a* $(\mathcal{C}, \mathcal{G})$-sequence.

3.1.2 Sequence of codes

Let $(\mathcal{C}_i)$ be a $(\mathcal{C}, \mathcal{G})$-sequence. For any $i \in \{0, \ldots, r-1\}$, the factor group Γ_i which acts on the curve $\mathcal{C}_i$ is cyclic of order p_i. Write ζ_i for a primitive p_i^{th} root of unity and $\Gamma_i = \langle \gamma_i \rangle$.

For $i \in \{0, \ldots, r\}$, we aim to define an AG code $C_i \subset \mathbb{F}^{\mathcal{P}_i}$ associated to a divisor $D_i \in \mathrm{Div}(\mathcal{C}_i)$ on an evaluation set $\mathcal{P}_i$. The rest of this subsection is dedicated to the choice of the divisors D_i and the sets $\mathcal{P}_i$.

Evaluation points. From a set $\mathcal{P}_0 \subset \mathcal{C}(\mathbb{F})$, we want to define a sequence of set of points $(\mathcal{P}_i) \subset \mathcal{C}_i(\mathbb{F})$ recursively by $\mathcal{P}_{i+1} = \pi_i(\mathcal{P}_i)$.

For the consistency of our protocol, we need for each $i \in \{0, \ldots, r-1\}$ that every point in $\mathcal{P}_{i+1}$ admits exactly p_i preimages under π_i. Since the last curve $\mathcal{C}_r$ is isomorphic to the quotient $\mathcal{C}/\mathcal{G}$, it is necessary and sufficient that the first set $\mathcal{P}_0 \subset \mathcal{C}_0$ is a union of $\mathcal{G}$-orbits of size $|\mathcal{G}|$, i.e. that $\mathcal{G}$ acts freely on $\mathcal{P}_0$.

Divisors. Fix a divisor $D_0 \in \mathrm{Div}(\mathcal{C}_0)$, not only globally Γ_0-invariant but also supported by Γ_0-fixed points. This way, the support of D_0 does not meet the set $\mathcal{P}_0$.

To make our protocol complete and sound, we need to choose at each step a divisor which is compatible with the previous one in the sense of the following definition.

Definition 3.2. *Let $D_i \in \mathrm{Div}(\mathcal{C}_i)$ and $\mu_i \in \mathbb{F}(\mathcal{C}_i)$ such that*

$$\gamma_i \cdot \mu_i = \zeta_i \mu_i. \tag{5}$$

For any $j \in \{0, \dots, p_i - 1\}$, we define the divisor

$$E_{i,j} := \left\lfloor \frac{1}{p_i} \pi_{i*}(D_i + j(\mu_i)) \right\rfloor \in \mathrm{Div}(\mathcal{C}_{i+1}). \tag{6}$$

A divisor $D_{i+1} \in \mathrm{Div}(\mathcal{C}_{i+1})$ is said to be compatible with (D_i, μ_i) *if all the following assertions hold.*

1. *D_{i+1} is supported by Γ_{i+1}-fixed points,*
2. *for every $j \in \{0, \dots, p_i - 1\}$, $E_{i,j} \leq D_{i+1}$,*
3. *for every $j \in \{0, \dots, p_i - 1\}$, there exists a function $\nu_{i+i,j} \in \mathbb{F}(\mathcal{C}_{i+1})$ such that*

$$(\nu_{i+i,j})_\infty = D_{i+1} - E_{i,j}. \tag{7}$$

The divisors $E_{i,j}$ in (6) coincide with those in Theorem 2.2 and thus satisfy

$$L_{\mathcal{C}_i}(D_i) = \bigoplus_{j=0}^{p_i-1} \mu_i^j \pi_i^* L_{\mathcal{C}_{i+1}}(E_{i,j}). \tag{8}$$

The first requirement ensures that the support of D_{i+1} does not intersect with the set of evaluation points $\mathcal{P}_{i+1}$. The second one implies that $L(E_{i,j}) \subseteq L(D_{i+1})$. The last condition means that for every $f_j \in L(E_{i,j})$, the function $\nu_{i+1,j} f_j$ lies in $L(D_{i+1})$.

Among those three requirements, the third is definitely the most compelling and requires some geometric knowledge about the curves C_i. Indeed, on a general curve, not every effective divisor is the poles locus of a function and characterizing which effective divisors arise this way is at the heart of the Weierstrass gaps theory. Nonetheless, the existence of the balancing functions $\nu_{i+1,j}$ happens to be the main ingredient in Lemma 6.4, which takes a prominent role in the construction of the folding operators.

Definition 3.3 ((μ_i)-compatibility)**.** *Let $(\mathcal{C}_i)$ be a $(\mathcal{C}, \mathcal{G})$-sequence. For every $i \in \{0, \dots, r-1\}$, take $\mu_i \in \mathbb{F}(\mathcal{C}_i)$ satisfying* (5)*. A sequence of divisor $(D_i) \in \mathrm{Div}(\mathcal{C}_i)$ is said to be* (μ_i)-compatible *if for every $i \in \{0, \dots, r-1\}$, the divisor D_{i+1} is (D_i, μ_i)-compatible.*

We have now described all the key components to formally define the notion of foldable codes. However, to ensure a good soundness of the protocol, we add a constraint on each divisor D_{i+1} regarding D_i. Indeed, as illustrated by Example 3.11, even though there exists a (D_i, μ_i)-compatible divisor D_{i+1}, its degree may be unexpectedly substantial, which would likely deteriorate the relative minimum distance of C_{i+1}. We thus demand D_{i+1} to be equal to one of the divisors $E_{i,j}$ (6) that appear in the decomposition (8) of $L_{\mathcal{C}_i}(D)$.

Definition 3.4 (Foldable AG codes)**.** *Let $C = C(\mathcal{C}, \mathcal{P}, D)$ be an AG-code. This code is said to be* foldable *if the following conditions are satisfied.*

- *There exists a finite solvable group $\mathcal{G} \in \mathrm{Aut}(\mathcal{C})$ that acts freely on $\mathcal{P}$: a composition series of $\mathcal{G}$* (3) *provides a $(\mathcal{C}, \mathcal{G})$-sequence of curves $(\mathcal{C}_i)$;*
- *There exists $e \in (0, 1)$ such that $|\mathcal{G}| > |\mathcal{P}|^e$;*
- *There exist a sequence $(\mu_i) \in \mathbb{F}(\mathcal{C}_i)$ satisfying* (5) *and a sequence $(D_i) \in \mathrm{Div}(\mathcal{C}_i)$ that is (μ_i)-compatible such that for every $i \in \{0, \dots, r-1\}$,*

$$\exists\, j \in \{0, \dots, p_i - 1\} \text{ such that } D_{i+1} = E_{i,j}, \tag{9}$$

where the divisors $E_{i,j}$ are defined as per Definition 3.2.

Example 3.5 (RS codes are foldable AG codes.)**.** Assume the characteristic of $\mathbb{F}$ is larger than 2. Let $\mathcal{P} \subset \mathbb{F}$ such that $|\mathcal{P}| = 2^r$ for a certain integer r. We observe that for any degree bound d, the RS code

$$V \coloneqq \left\{ f \in \mathbb{F}^{\mathcal{P}}; \deg f \leq d \right\} = C(\mathbb{P}^1, \mathcal{P}, dP_\infty)$$

is a foldable AG code. By iterating the observation made in Example 2.3, we recover the construction of the RS proximity test of [BBHR18]. Firstly, the finite solvable $\mathbb{Z}/2^r\mathbb{Z}$ of size $|\mathcal{P}|$ acts on $\mathbb{P}^1$ via $[X_0 : X_1] \mapsto [X_0, \xi X_1]$, where ξ is a primitive 2^r-th root unity. It clearly fulfils the two first items of Definition 3.4. When considering its composition series

$$\mathbb{Z}/2^r\mathbb{Z} \rhd \mathbb{Z}/2^{r-1}\mathbb{Z} \rhd \cdots \rhd 1 \tag{10}$$

and the action of the corresponding factor group $\Gamma = \langle \gamma \rangle \simeq \mathbb{Z}/2\mathbb{Z}$, we obtain a trivial sequence of curves $(\mathcal{C}_i)$ with $\mathcal{C}_i = \mathbb{P}^1$. Next, consider the sequence (μ_i) with $\mu_i = \mu = x \coloneqq \frac{X_1}{X_0}$, then $\gamma\mu = -\mu$. Set $d_0 \coloneqq d$, and for any $i \in \{0, \dots, r-1\}$, $d_{i+1} \coloneqq \left\lfloor \frac{d_i}{2} \right\rfloor$. Note that there exists $r' < r$ such that $d_{r'}, \dots, d_r$ are all equal to 0. The sequence (D_i) with $D_i = \left\lfloor \frac{d_i}{2} \right\rfloor P_\infty$ is (μ_i)-compatible (Definition 3.2), by letting $\nu_{i+1,j}$ to be the constant function equal to 1 if $\left\lfloor \frac{d_i}{2} \right\rfloor = \left\lfloor \frac{d_i - 1}{2} \right\rfloor$, and $\nu_{i+1,j} : x \mapsto x$ otherwise.

3.2 Foldable AG codes on Kummer curves

Let us consider a Kummer curve over a finite field $\mathbb{F}$ defined by an equation of the form

$$\mathcal{C} : y^N = f(x) = \prod_{\ell=1}^{m} (x - \alpha_\ell) \tag{11}$$

where f is a degree m separable polynomial of $\mathbb{F}[X]$ and $\gcd(N, m) = 1$. Let us denote by P_ℓ the point $(\alpha_\ell, 0)$ and P_∞ the unique point of $\mathcal{C}$ lying on the line at infinity.

Sequence of curves. Assume that $\gcd(N, |\mathbb{F}|) = 1$. The group $\mathbb{Z}/N\mathbb{Z}$ acts on $\mathcal{C}$ via the morphism $(x, y) \mapsto (x, \zeta y)$ where ζ is a primitive N^{th} root of unity. The cyclic group $\mathbb{Z}/N\mathbb{Z}$ is solvable: writing the prime decomposition of $N = \prod_{i=0}^{r-1} p_i$ gives the following sequence of subgroups

$$\mathbb{Z}/N\mathbb{Z} \rhd \mathbb{Z}/N_1\mathbb{Z} \rhd \mathbb{Z}/N_2\mathbb{Z} \rhd \cdots \rhd \mathbb{Z}/N_{r-1}\mathbb{Z} \rhd 1, \tag{12}$$

where

$$N_i = \prod_{j=i}^{r-1} p_j. \tag{13}$$

The i-th factor group Γ_i is isomorphic to $\mathbb{Z}/p_i\mathbb{Z}$. It is spanned by $\gamma_i : (x, y) \mapsto (x, \zeta_i y)$ where ζ_i is a primitive p_i^{th} root of unity.

Set $\mathcal{C}_0 := \mathcal{C}$. By Section 3.1, the composition series (12) gives a sequence of curves $(\mathcal{C}_i)$ in which the i^{th} curve is defined by

$$\mathcal{C}_i : y^{N_i} = f(x). \tag{14}$$

and has genus

$$g_i = \frac{(N_i - 1)(m-1)}{2}.$$

The last curve $\mathcal{C}_r$ has genus 0 and is isomorphic to the projective line $\mathbb{P}^1$. These successive quotients provide a sequence of projections $\pi_i : \mathcal{C}_i \to \mathcal{C}_{i+1}$ defined by $\pi_i(x, y) = (x, y^{p_i})$:

$$\begin{tikzcd} \mathcal{C}_0 \arrow[loop above, "\gamma_0"] \arrow[r, two heads, "\pi_0"] & \dots \arrow[r, two heads, "\pi_i"] & \mathcal{C}_i \arrow[loop above, "\gamma_i"] \arrow[r, two heads, "\pi_{i+1}"] & \mathcal{C}_{i+1} \arrow[loop above, "\gamma_{i+1}"] \arrow[r, two heads] & \dots \arrow[r, two heads, "\pi_{r-1}"] & \mathcal{C}_r \simeq \mathbb{P}^1. \end{tikzcd}$$

Example 3.6. The Hermitian curve defined over $\mathbb{F}_{q^2}$ by

$$\mathcal{C}_0 : y^{q+1} = x^q + x. \tag{15}$$

is a well-studied particular case of Kummer type curve. In this case, every curve in a $(\mathcal{C}, \mathcal{G})$-sequence is maximal over $\mathbb{F}_{q^2}$ [Lac87, Proposition 6], i.e. $|\mathcal{C}_i(\mathbb{F}_{q^2})| = q^2 + 1 + 2g_i q$.

Stabilized points. Let us denote P_∞^i the unique point at infinity on the curve $\mathcal{C}_i$. One can easily check that $P_\infty^i := \begin{cases} (1:0:0) & \text{if } N > m \\ (0:1:0) & \text{otherwise.} \end{cases}$ Note that N and m are assumed coprime and thus are never equal.

The points of $\mathcal{C}_0$ whose stabilizer under $\mathbb{Z}/N\mathbb{Z}$ is non trivial are in fact fixed by $\mathbb{Z}/N\mathbb{Z}$ and consist precisely in $P_1, \dots, P_\ell$ and P_∞^i.

Determination of the functions μ_i. To construct a valid sequence of divisors, we have to exhibit for each step $i \in \{0, \dots, r-1\}$ a function $\mu_i \in \mathbb{F}(\mathcal{C}_i)$ satisfying $\gamma_i \cdot \mu_i = \zeta_i \mu_i$. If its existence is given by Theorem 2.2, one can easily check that

$$\mu_i = y \tag{16}$$

fits. Maharaj [Mah04] proved Theorem 2.2 on Kummer curve for this particular choice.

An example of a sequence of (y)-compatible divisors. In order to investigate (y)-compatible sequence, we need to handle the divisor associated to y and some other elementary functions on each curve $\mathcal{C}_i$, described for instance in [MQS15].

Lemma 3.7 ([MQS15])**.** *On $\mathcal{C}_i$ for every $i \in \{0, \dots, r-1\}$, we have*

1. $(x - \alpha_\ell) = N_i(P_\ell - P_\infty^i)$,
2. $(y) = P_1 + \dots + P_m - mP_\infty^i$.

We now give sufficient conditions on the curve $\mathcal{C}_0$ and the first divisor D_0 to get a sequence of (y)-compatible divisors.

Lemma 3.8. *Set* $D_0 = \sum_{\ell=1}^{m} a_{0,\ell} P_\ell + b_0 P_\infty^0 \in \mathrm{Div}(\mathcal{C}_0)$.

Assume that $m \equiv -1 \mod N$ and that the integers $a_{0,1}, \ldots, a_{0,m}, b_0$ are all divisible by N. For every $i \in \{0, \ldots, r-1\}$, set $D_{i+1} = \frac{D_i}{p_i}$. Then, the divisor D_{i+1} is (D_i, y)-compatible.

Proof. For $i \in \{1, \ldots, r\}$, let us set $a_{i,\ell} = \frac{a_{i-1,\ell}}{p_{i-1}}$ and $b_i = \frac{b_{i-1}}{p_{i-1}}$ such that $D_i = \sum_{\ell=1}^{m} a_{i,\ell} P_\ell + b_i P_\infty^i$.

Fix $i \in \{0, \ldots, r-1\}$. The divisor D_i is supported only by Γ_i-fixed points.

For any $j \in \{0, \ldots, p_i - 1\}$, we have

$$E_{i,j} = \left\lfloor \frac{1}{p_i} \pi_{i*}(D_i + j(y)) \right\rfloor = \sum_{\ell=1}^{m} \left\lfloor \frac{a_{i,\ell} + j}{p_i} \right\rfloor P_\ell + \left\lfloor \frac{b_i - jm}{p_i} \right\rfloor P_\infty^{i+1}.$$

Since N_i divides N, we have $m \equiv -1 \mod N_i$. Write $m = \kappa_i N_i - 1$ with $\kappa_i \geq 1$.

The hypothesis on the integers $a_{0,1}, \ldots, a_{0,m}, b_0$ entails

$$\left\lfloor \frac{a_{i,\ell} + j}{p_i} \right\rfloor = a_{i+1,\ell} + \left\lfloor \frac{j}{p_i} \right\rfloor = a_{i+1,\ell}$$
$$\left\lfloor \frac{b_i - jm}{p_i} \right\rfloor = b_{i+1} - \frac{j\kappa_i N_i}{p_i} + \left\lfloor \frac{j}{p_i} \right\rfloor = b_{i+1} - j\kappa_i N_{i+1}.$$

Then $E_{i,j} = D_{i+1} - j\kappa_i N_{i+1} P_\infty^{i+1}$. In particular, $D_{i+1} = E_{i,0}$ and $E_{i,j} \leq D_{i+1}$. Any $\nu_{i+1,j} \coloneqq (x - \alpha)^{\kappa_i j}$ with $\alpha \in \{\alpha_1, \ldots, \alpha_m\}$ gives the last condition on D_{i+1} for it to be (D_i, μ_i)-compatible by Definition 3.2, i.e. $D_{i+1} - E_{i,j} = (\nu_{i+1,j})_\infty$. □

We have gathered all the components to exhibit a foldable code on a family of Kummer curves.

Proposition 3.9. *Let $\mathcal{C}_0$ be a Kummer curve defined by* (11) *with $m \equiv -1 \mod N$. Take an evaluation set $\mathcal{P}_0 \subseteq \mathcal{C}_0(\mathbb{F}) \setminus \{P_1, \ldots, P_m, P_\infty^0\}$ formed by $\mathbb{Z}/N\mathbb{Z}$-orbits. Take $D_0 \in \mathrm{Div}(\mathcal{C}_0)$ satisfying hypothesis of Lemma 3.8. If $N > n^e$ for some $e \in (0, 1)$, then the AG code $C = C(\mathcal{C}_0, \mathcal{P}_0, D_0)$ is foldable.*

Remark 3.10. The condition on the coefficients of D_0 can be loosen while the previous statement still holds. If $a_{0,1}, \ldots, a_{0,m}, b_0$ are divisible by $\prod_{i=0}^{r-2} p_i$ and not necessarily by p_{r-1}, we choose $a_{r,\ell} = \left\lceil \frac{a_{r-1,\ell}}{p_{r-1}} \right\rceil$ and $b_r = \left\lfloor \frac{b_{r-1}}{p_{r-1}} \right\rfloor$ for the coefficients of D_r. The first two conditions of Definition 3.2 are satisfied. The last curve $\mathcal{C}_r$ being isomorphic to $\mathbb{P}^1$, the last requirement of Definition 3.2 is directly implied by the second one.

Lemma 3.8 provides sufficient conditions to make C_{i+1} as small as possible compared to C_i by choosing D_{i+1} among the divisors $E_{i,j}$, as required for a sequence of foldable codes by Definition 3.4. Ignoring the additional condition (9) can make the code C_{i+1} grow drastically, as illustrated by the next example.

Example 3.11. Over $\mathbb{F}_8$, consider $y^N = x^m + x$ where $N = 9$ and $m = 5$. Then $m \not\equiv -1 \mod N$ and $N = p_0 p_1$ with $p_0 = p_1 = 3$. For $D_0 = 18 P_\infty^0$, we have

$$E_{0,0} = \left\lfloor \frac{18}{3} \right\rfloor P_\infty^1 = 6P_\infty^1, \qquad E_{0,1} = \left\lfloor \frac{18 - 5}{3} \right\rfloor P_\infty^1 = 4P_\infty^1, \qquad E_{0,2} = \left\lfloor \frac{18 - 2 \times 5}{3} \right\rfloor P_\infty^1 = 2P_\infty^1.$$

Choosing $D_1 = E_{0,0}$ would satisfy the first and the second conditions of Definition 3.2 to be (D_0, y)-compatible but not the third one. One can reasonably ask the support of D_1 to consist only in $\pi_0(P^0_\infty) = P^1_\infty$, as one-point codes are generally better understood. The Weierstrass gap theory on Kummer curves (e.g. [MQS15, Theorem 3.2]) entails that if a function on $\mathcal{C}_1 : y^3 = x^5 + x$ has a pole locus of the form αP^1_∞, then $\alpha \in 3\mathbb{Z}_+ + 5\mathbb{Z}_+$. Therefore the smallest divisor of the form $D_1 = d_1 P^1_\infty$ that is (D_0, y)-compatible is $D_1 = 12P^1_\infty$. With such a choice of divisors, the code C_0 of dimension 15 is folded into the code C_1 of dimension 12 whereas the length of C_1 is the third of the length of C_0.

To estimate the parameters of the code by using the Riemnann-Roch theorem, we shall rely on the following result.

Lemma 3.12. *Assume that* $2(g_0 - 1) < \deg(D_0)$ *(resp.* $\deg(D_0) < n_0$*). Then for every* $i \in \{0, \dots, r\}$*,* $2(g_i - 1) < \deg(D_i)$ *(resp.* $\deg(D_i) < n_i$*).*

Proof. It is enough to notice that for every $i \in \{0, \dots, r-1\}$,

$$\deg D_{i+1} = \frac{\deg D_i}{p_i}, \qquad n_{i+1} = \frac{\deg n_i}{p_i}, \qquad \text{and} \quad g_{i+1} \le \frac{g_i}{p_i}.$$

□

In other words, if the degree of the first divisor is such that we can estimate the parameters of C_0 thanks to Riemann-Roch Theorem, then we handle the parameters of all the sequence of codes.

Proposition 3.13. *If* $\deg(D_0) < n_0$*, then for every* $i \in \{0, \dots, r\}$*, the code* C_i *has length* n_i *and minimum relative distance* $\Delta(C_i) = 1 - \frac{\deg D_0}{n_0}$*. In particular, the RS code* C_r *has length* $\frac{n_0}{N}$*, dimension* $\frac{\deg D_0}{N} + 1$ *and relative minimum distance* $1 - \frac{\deg D_0}{n_0}$*.*

Moreover, if $2(g_0 - 1) < \deg(D_0)$*, for every* $i \in \{0, \dots, r\}$*, the code* C_i *has dimension* $\deg D_i - g_i + 1$*.*

Proof. The length of C_i is n_i by construction and its dimension is given by the Riemann-Roch theorem. So let us prove the statement concerning the relative minimum distance.

First notice that $n_i = p_i n_{i+1}$ and $\deg(D_i) = p_i \deg(D_{i+1})$ so $1 - \frac{\deg D_i}{n_i} = 1 - \frac{\deg D_0}{n_0}$.

For $i = r$, the code C_r is a Reed-Solomon code of degree $0 \le \deg(D_r) < n_r$ by Lemma 3.12 and has the expected relative minimum distance.

Now assume that $\Delta(C_{i+1})$ equals $1 - \frac{\deg D_0}{n_0}$ and let us prove that so does $\Delta(C_i)$.

On the one hand, the divisor D_{i+1} corresponds to $E_{i,0}$ then for every $f \in C_{i+1}$, $f \circ \pi_i \in C_i$. In addition, the weight of $f \circ \pi_i$ in C_i is p_i times the weight of f in C_{i+1}. Since $n_i = p_i n_{i+1}$, we have $\Delta(C_i) \le \Delta(C_{i+1})$. On the other hand, as $\deg(C_i) < n_i$, we have $\Delta(C_i) \ge 1 - \frac{\deg D_i}{n_i}$, which concludes the proof. □

4 IOPP for foldable AG codes

Now that we have determined the needed properties of an AG-code to be foldable, we construct the fundamental building block of our IOPP by generalizing the so-called algebraic hash function of [BKS18] to the AG codes setting, and we refer to it as the *folding operator*. Next, we provide a formal description of the IOPP system (P, V) and state the theorem capturing its efficiency properties.

4.1 Folding operators

Let $C_0 = C(\mathcal{C}_0, \mathcal{P}_0, D_0)$ be a code satisfying Definition 3.4. We consider its associated $(\mathcal{C}, \mathcal{G})$-sequence of curves $(\mathcal{C}_i)$ and its sequence of divisors (D_i). By Definition 3.4, the divisor D_{i+1} in the general case is equal to one of the divisors $E_{i,j}$. From now on, we assume without loss of generality (see Remark 4.3) that for every $i \in \{0, \ldots, r-1\}$,

$$D_{i+1} = E_{i,0}. \tag{17}$$

To test proximity of a function $f^{(0)} : \mathcal{P}_0 \to \mathbb{F}$ to C_0, we aim to inductively reduce the problem to a smaller one, consisting of testing proximity to the code $C_i = C(\mathcal{C}_i, \mathcal{P}_i, D_i)$. Broadly speaking, our goal is to define from any function $f^{(i)} : \mathcal{P}_i \to \mathbb{F}$ a function $f^{(i+1)} : \mathcal{P}_{i+1} \to \mathbb{F}$ such that the relative distance $\Delta(f^{(i+1)}, C_{i+1})$ is roughly equal to $\Delta(f^{(i)}, C_i)$.

Fix $i \in \{0, \ldots, r-1\}$ and let $f : \mathcal{P}_i \to \mathbb{F}$ be an arbitrary function.

Notation 4.1 (Interpolation polynomial)**.** For each $P \in \mathcal{P}_{i+1}$, let us denote $S_P \coloneqq \pi_i^{-1}(\{P\})$ the set of p_i distinct preimages of P and consider

$$I_{f,P}(X) \coloneqq \sum_{j=0}^{p_i-1} X^j a_{j,P} \tag{18}$$

the univariate polynomial over $\mathbb{F}$ of degree less than p_i which interpolates the set of points $\left\{(\mu_i(\widehat{P}), f(\widehat{P})); \widehat{P} \in S_P\right\}$. Then for every $j \in \{0, \ldots, p_i - 1\}$, we define the function

$$f_j : \left\{ \begin{array}{ccc} \mathcal{P}_{i+1} & \to & \mathbb{F}, \\ P & \mapsto & a_{j,P}. \end{array} \right. \tag{19}$$

Given $f : \mathcal{P}_i \to \mathbb{F}$, the idea is to define p_i functions $f_j : \mathcal{P}_{i+1} \to \mathbb{F}$, where $|\mathcal{P}_{i+1}| = \frac{|\mathcal{P}_i|}{p_i}$ such that f corresponds to the evaluation of a function in $L(D_i)$ if and only if each f_j coincides with a function in $L(E_{i,j}) \subset L(D_{i+1})$. Instead of testing for each $j \in \{0, \ldots, p_i - 1\}$ whether $f_j \in C_{i+1}$, we reduce those p_i claims to a single one, by taking a random linear combination of the f_j's, which we referred to as a *folding of* f. By linearity of the codes, such a combination of the f_j's belongs to C_{i+1} whenever $f \in C_i$ (see Proposition 4.5 below). However, for soundness analysis, one needs to ensure that no f_j corresponds to a function lying in $L(D_{i+1}) \setminus L(E_{i,j})$. Some safeguards are embedded into the folding operation by introducing the balancing functions $\nu_{i+1,j}$ from Definition 3.2 in the second term of the sum in Equation (20).

Definition 4.2 (Folding operator)**.** *For any* $\boldsymbol{z} = (z_1, z_2) \in \mathbb{F}^2$*, we define the* folding of f *to be the function* $\textsf{Fold}[f, \boldsymbol{z}] : \mathcal{P}_{i+1} \to \mathbb{F}$ *such that*

$$\textsf{Fold}[f, \boldsymbol{z}] \coloneqq \sum_{j=0}^{p_i-1} z_1^j f_j + \sum_{j=1}^{p_i-1} z_2^j \nu_{i+1,j} f_j \tag{20}$$

where the functions f_j *are defined in Equation* (19) *and the functions* $\nu_{i+j,j}$ *in Definition 3.2.*

Remark 4.3. As said earlier and for the sake of clarity, we present our construction assuming that $D_{i+1} = E_{i,0}$. When $D_{i+1} = E_{i,j_i}$ for a certain $j_i \neq 0$, the second term of the folding operator can be adjusted as follows, without affecting any of our subsequent statements:

$$\textsf{Fold}[f, \boldsymbol{z}] \coloneqq \sum_{j=0}^{p_i-1} z_1^j f_j + \sum_{j=0}^{j_i-1} z_2^{j+1} \nu_{i+1,j} f_j + \sum_{j=j_i+1}^{p_i-1} z_2^j \nu_{i+1,j} f_j.$$

For foldable AG codes on Kummer curves (Section 3.2), we underline that Lemma 3.8 actually ensures that $D_{i+1} = E_{i,0}$ for every $i \in \{0, \ldots, r-1\}$.

Given the p_i points $((\mu_i(\widehat{P}), f(\widehat{P})))_{\widehat{P} \in S_P}$, one can determine the coefficients $(a_{j,P})_{0 \leq j < p}$ of $I_{f,P}$ defined in (18) by polynomial interpolation. Recalling that for each $P \in \mathcal{P}_{i+1}$, we have $f_j(P) = a_{j,P}$, we get the following lemma. This lemma will allow to obtain efficient prover time and fast verifier decision complexity.

Lemma 4.4 (Locality)**.** *Let $\boldsymbol{z} \in \mathbb{F}^2$. For each $P \in \mathcal{P}_{i+1}$, the value of $\textsf{Fold}\,[f, \boldsymbol{z}]\,(P)$ can be computed with exactly p_i queries to f, namely at the points $\pi_i^{-1}(\{P\})$.*

We now show a key property of the folding operator for the completeness of our IOPP.

Proposition 4.5 (Completeness)**.** *Let $\boldsymbol{z} \in \mathbb{F}^2$. If $f \in C_i$, then $\textsf{Fold}\,[f, \boldsymbol{z}] \in C_{i+1}$.*

Proof. Write $\boldsymbol{z} = (z_1, z_2)$. If $f \in C_i$, it coincides with a function of $L(D_i)$. By definition of the divisors $E_{i,j}$ and Theorem 2.2, there exist some functions $\widetilde{f}_j \in L(E_{i,j})$ such that

$$f = \sum_{j=0}^{p_i - 1} \mu_i^j \widetilde{f}_j \circ \pi_i.$$

Let $P \in \mathcal{P}_{i+1}$. For any $\widehat{P} \in S_P$,

$$\textsf{Fold}\left[f, (\mu_i(\widehat{P}), 0)\right](P) = I_{f,P}(\mu_i(\widehat{P})) = f(\widehat{P}) = \sum_{j=0}^{p_i - 1} \mu_i(\widehat{P})^j \widetilde{f}_j(P).$$

Moreover, for all $P \in \mathcal{P}_{i+1}$, polynomials $I_{f,P}(X), \textsf{Fold}\,[f, (X, 0)]\,(P) \in \mathbb{F}[X]$ are of degree less than p_i and agree on $\left\{\mu_i(\widehat{P}); \widehat{P} \in S_P\right\}$ of size p_i, therefore they are equal. In particular,

$$\textsf{Fold}\left[f, (\mu_i(\widehat{P}), 0)\right](P) = \sum_{j=0}^{p_i - 1} \mu_i(\widehat{P})^j f_j(P).$$

Thus, for all $P \in \mathcal{P}_{i+1}$,

$$\sum_{j=0}^{p_i - 1} \mu_i(\widehat{P})^j (\widetilde{f}_j(P) - f_j(P)) = 0$$

and the polynomial

$$\sum_{j=0}^{p_i - 1} X^j (\widetilde{f}_j(P) - f_j(P))$$

of degree less than p_i is zero on at least $\left|\left\{\mu_i(\widehat{P}); P \in \mathcal{P}_{i+1}\right\}\right| = p_i$ points. Hence, for every $j \in \{0, \ldots, p_i - 1\}$, the function f_j defined in Equation (19) coincides with $\widetilde{f}_j$ and

$$\textsf{Fold}\,[f, \boldsymbol{z}] := \sum_{j=0}^{p_i - 1} z_1^j \widetilde{f}_j + \sum_{j=1}^{p_i - 1} z_2^j \nu_{i+1,j} \widetilde{f}_j$$

where $\widetilde{f}_j \in L(E_{i,j}) \subseteq L(D_{i+1})$ and $\nu_{i+1,j} f_j \in L(D_{i+1})$, by definition of the divisors $E_{i,j}$, D_{i+1} and the functions $\nu_{i+1,j}$ (see Definition 3.2). Thus each term of $\textsf{Fold}\,[f, \boldsymbol{z}]$ lies in the vector space C_{i+1}, which concludes the proof. $\square$

4.2 Description of the AG-IOPP for foldable AG codes

Let $C_0 = C(\mathcal{C}_0, \mathcal{P}_0, D_0)$ be a foldable AG code over an alphabet $\mathbb{F}$. We formally describe our IOPP system $(\mathsf{P}_{\mathsf{AG}}, \mathsf{V})$ for testing proximity of a function $f^{(0)} : \mathcal{P}_0 \to \mathbb{F}$ to C_0. As in the FRI protocol, our AG-IOPP is divided in two phases, referred to as COMMIT and QUERY and respectively outlined in Figure 1 and Figure 2. Before any interaction, P and V agree on:

– a $(\mathcal{C}, \mathcal{G})$-sequence of curves $(\mathcal{C}_i)$, for which we denote the length of the composition serie of $\mathcal{G}$ by r.
– a sequence of functions $(\mu_i) \in \mathbb{F}(\mathcal{C}_i)$ satisfying (5),
– a sequence of codes (C_i) where for each $i \in \{0, \ldots, r\}$, $C_i = (\mathcal{C}_i, \mathcal{P}_i, D_i)$ and $\mathcal{C}_i, \mathcal{P}_i$ and D_i are defined as per Section 3.1,
– a sequence of balancing functions $(\nu_{i+1})_{0 \le i < r}$ of p_i-tuples of functions in $\mathbb{F}(\mathcal{C}_{i+1})$ such that $\nu_{i+1} = (\nu_{i+1,j})_{0<j<p_i}$ and $\nu_{i+1,j}$ satisfies (7).

We recall that the choice of a sequence $(\mathcal{C}_i)$ induces a sequence of projections $\pi_i : \mathcal{C}_i \to \mathcal{C}_{i+1}$.

COMMIT phase. The COMMIT phase (Figure 1) is an interaction over r rounds between P and V. For each round $i \in \{0, \ldots, r-1\}$, the verifier samples a random challenge $\boldsymbol{z}^{(i)} \in \mathbb{F}^2$. As an answer, the prover gives oracle access to function $f^{(i+1)} : \mathcal{P}_{i+1} \to \mathbb{F}$, which is expected to be equal to $\textsf{Fold}\left[f^{(i)}, \boldsymbol{z}^{(i)}\right]$. To compute the values of $f^{(i+1)}$ on $\mathcal{P}_{i+1}$, an honest prover P exploits the fact that the folding of $f^{(i)}$ is locally computable (Lemma 4.4). Namely, for each $P \in \mathcal{P}_{i+1}$, P computes the coefficients $(a_{j,P})_{0 \le j < p}$ of $I_{f^{(i)},P} \in \mathbb{F}[X]$ from $f^{(i)}|_{S_P}$, evaluates $\nu_{i+1,j}$ at P, and set

$$\textsf{Fold}\left[f^{(i)}, \boldsymbol{z}^{(i)}\right](P) \coloneqq \sum_{j=0}^{p_i-1} \left(z_1^{(i)}\right)^j a_{j,P} + \sum_{j=1}^{p_i-1} \left(z_2^{(i)}\right)^j \nu_{i+1,j}(P) a_{j,P}.$$

COMMIT Phase
(interactive)

Common input: C_0 a foldable AG code defined by $(\mathbb{F}, \mathcal{C}_0, \mathcal{P}_0, D_0)$, r a number of rounds, (C_i) a sequence of codes, $(\nu_{i+1})_{0 \le i < r}$ and (μ_i) some sequences of functions.
Prover's input: $f^{(0)} : \mathcal{P}_0 \to \mathbb{F}$.
Output: a sequence of oracle functions $(f^{(0)}, \ldots, f^{(r)}) \in \mathbb{F}_q^{\mathcal{P}_1} \times \ldots \times \mathbb{F}_q^{\mathcal{P}_r}$.

1. For each round i from 0 to $r-1$:
   (a) V picks uniformly at random $\boldsymbol{z}^{(i)}$ in $\mathbb{F}^2$ and sends it to P,
   (b) P computes $f^{(i+1)} = \textsf{Fold}\left[f^{(i)}, \boldsymbol{z}^{(i)}\right]$,
   (c) P gives oracle access to $f^{(i+1)} : \mathcal{P}_{i+1} \to \mathbb{F}$.

Figure 1: AG Codes IOPP – COMMIT Phase

QUERY phase. (Figure 2) During the QUERY phase, one of the two tasks of the verifier V is to check that each pair of successive oracle functions $(f^{(i)}, f^{(i+1)})$ is consistent. A standard idea is to check that the equality

$$f^{(i+1)} = \textsf{\textbf{Fold}}\left[f^{(i)}, \boldsymbol{z}^{(i)}\right] \tag{21}$$

holds at a random point in $\mathcal{P}_{i+1}$. By leveraging the local property of the folding operator, such a test requires only p_i queries to $f^{(i)}$ and 1 query to $f^{(i+1)}$. As in [BBHR18], we call this step of verification a *round consistency test*. This test corresponds to the block inside Step 1.(b) of the QUERY phase in Figure 2. The verifier begins by sampling at random $Q_0 \in \mathcal{P}_0$ and once this is done, all the locations of the round consistency tests run inside the current **query test** are determined. More specifically, for each round i, V defines $Q_{i+1} \coloneqq \pi_i(Q_i)$ to be the random point where Equation (21) is checked. Through this process, the round consistency tests are correlated to improve soundness. Such a **query test** can be seen as a *global* consistency test, similar to the one of the FRI protocol. For the final test, V reads $f^{(r)} : \mathcal{P}_r \to \mathbb{F}$ in its entirety to test if $f^{(r)} \in C_r$.

QUERY Phase:
(run by V only)

Input: (the first four items must correspond to the COMMIT phase)
- C_0 an AG code defined by $(\mathbb{F}, \mathcal{C}_0, \mathcal{P}_0, D_0)$, r a number of rounds,
- sequence of codes (C_i), sequences of functions (ν_{i+1}) and (μ_i),
- transcript including $\boldsymbol{z}^{(0)}, \dots, \boldsymbol{z}^{(r-1)} \in \mathbb{F}^2$,
- oracle functions $f^{(0)}, f^{(1)}, \dots, f^{(r-1)}, f^{(r)}$,
- α repetition parameter.

Output: **accept** or **reject**.

1. Repeat α times the following **query test**:
   (a) Pick $Q_0 \in \mathcal{P}_0$ uniformly at random.
   (b) For $i = 0$ to $r - 1$, run the following *round consistency test*:
      i. Define $Q_{i+1} \in \mathcal{P}_{i+1}$ by $Q_{i+1} = \pi_i(Q_i)$,
      ii. Query $f^{(i+1)}$ to get $f^{(i+1)}(Q_{i+1})$ and query $f^{(i)}$ at points $\widehat{Q} \in S_{Q_{i+1}}$,
      iii. Compute the value $\textsf{\textbf{Fold}}\left[f^{(i)}, \boldsymbol{z}^{(i)}\right](Q_{i+1})$,
      iv. If $f^{(i+1)}(Q_{i+1}) \neq \textsf{\textbf{Fold}}\left[f^{(i)}, \boldsymbol{z}^{(i)}\right](Q_{i+1})$, then return **reject**.
2. *Final test:* return **acccept** if and only if $f^{(r)} \in C_r$.

Figure 2: AG Codes IOPP – QUERY Phase

4.3 Properties of the AG-IOPP

For any $\varepsilon \in (0, 1]$, let $J_\varepsilon : [0, 1] \to [0, 1]$ be the function such that $J_\varepsilon(\lambda) = 1 - \sqrt{1 - (1 - \varepsilon)\lambda}$ and denote $J_\varepsilon^l = \underbrace{J_\varepsilon \circ \cdots \circ J_\varepsilon}_{l \text{ times}}$.

Theorem 4.6. *Let $C_0 = C(\mathcal{C}_0, \mathcal{P}_0, D_0)$ be a foldable AG code of length $n \coloneqq |\mathcal{P}_0|$. By definition, C_0 admits a solvable group $\mathcal{G} \in \mathrm{Aut}(\mathcal{C}_0)$ such that $|\mathcal{G}| > n^e$ for a certain $e \in (0, 1)$ and in-*

duces a sequence of codes (C_i)*. Denote* p_{max} *the largest integer of the prime decomposition of* $|\mathcal{G}|$, $\lambda := \min_i \Delta(C_i)$ *and* $\gamma := \min\left(J_\varepsilon^{p_{max}}(\lambda), \frac{1}{2}(\lambda + \frac{\varepsilon}{2})\right)$*.*

The protocol described in Figures 1-2 is an IOPP system (P, V) *for* C_0 *satisfying:*

Perfect completeness: *If* $f^{(0)} \in C_0$ *and* $f^{(1)}, \ldots, f^{(r)}$ *are honestly generated by the prover, the verifier outputs* accept *with probability 1.*

Soundness: *Assume* $f^{(0)}$ *is* δ*-far from* C_0 *and let* $\varepsilon \in (0, 1)$*. With probability at least* $1 - \mathit{err}_{commit}$ *over the randomness of the verifier during the* COMMIT *phase, where*

$$\mathit{err}_{commit} \leq \log n \frac{p_{\max} - 1}{|\mathbb{F}|} \left(\frac{4}{\varepsilon}\right)^{p_{\max}+1},$$

and for any oracles $f^{(1)}, \ldots, f^{(r)}$ *adaptively chosen by a possibly dishonest prover* P^**, the probability that the verifier* V *outputs* accept *after a single* query test *is at most*

$$\mathit{err}_{query}(\delta) \leq (1 - \min(\delta, \gamma) + \varepsilon \log n).$$

Overall, for any prover P^**, the soundness error* $\mathit{err}(\delta)$ *after* α *repetitions of the* QUERY *phase satisfies*

$$\begin{aligned} \mathit{err}(\delta) &\leq \mathit{err}_{commit} + (\mathit{err}_{query}(\delta))^\alpha \\ &< \log n \frac{p_{\max} - 1}{|\mathbb{F}|} \left(\frac{4}{\varepsilon}\right)^{p_{\max}+1} + (1 - \min(\delta, \gamma) + \varepsilon \log n)^\alpha. \end{aligned}$$

Moreover, the IOPP system is public-coin, has round complexity $r(n) < \log n$ *and proof length* $l(n) < n$*. The verifier sends* $k(n) < 2 \log n$ *random field elements and makes* $q(n) < \alpha p_{max} \log n + n^{1-e}$ *queries.*

Proposition 4.7. *If* $p_{max} = 2$ *and* $\varepsilon < 1/3$*, soundness error of the IOPP provided by Theorem 4.6 satisfies*

$$\mathit{err}(\delta) \leq \frac{8 \log n}{\varepsilon^2} + (1 - \min(\delta, 1 - (1 - \lambda + \varepsilon)^{\frac{1}{3}}) + \varepsilon \log n)^\alpha.$$

The proof of Proposition 4.7 directly follows the analysis of Section 6 and is sketched in Appendix B.

Remark 4.8. When $\delta < \gamma$, error probability during the QUERY phase is roughly $(1 - \delta)^\alpha$. Thus, when targeting a fixed soundness error $2^{-\kappa}$, the ability to take a large proximity parameter δ yields to a smaller number of repetitions α. Hence, larger threshold γ is desirable to get better soundness error for a single query test. The value of this constant appears in soundness analysis from [BKS18, Theorem 4.5] for $p_{max} \geq 2$ and [BGKS20, Lemma 3.2] for $p_{max} = 2$ (see Section 6 and Appendix B). Improving such results for AG codes would lead to greater threshold γ, which would allow to take a smaller repetition parameter α when targeting soundness error $2^{-\kappa}$. As a result, this would also reduce the total number of queries $q(n)$ stated in Theorem 4.6 and leads to shorter non-interactive arguments (cf. Remark 1.3).

We break down the proof of Theorem 4.6 into two parts, the first one is given below. The second part, dedicated to soundness error, is covered by Section 6.

4.4 Proof of Theorem 4.6 - Part 1 (all but soundness)

(Perfect completeness) Let us assume that $f^{(0)} \in C_0$. For $i < r-1$, by letting $f^{(i+1)} = \textsf{Fold}\left[f^{(i)}, \boldsymbol{z}^{(i)}\right]$, the testing relation of the step 1.(b).iv. of the $\textsf{QUERY}$ phase is satisfied by definition of $\textsf{Fold}\left[f^{(i)}, \boldsymbol{z}^{(i)}\right]$. Furthermore, recalling Proposition 4.5, we have that for all i, if $f \in C_i$ then $\textsf{Fold}\left[f^{(i)}, \boldsymbol{z}^{(i)}\right] \in C_{i+1}$ for any $\boldsymbol{z}^{(i)} \in \mathbb{F}_q^2$. Thus the final test also passes, since $f^{(r)} \in C_r$. Therefore, the verifier accepts at the end of the $\textsf{QUERY}$ phase.

(Round complexity) We have that $\displaystyle\prod_{i=0}^{r-1} p_i = \frac{n}{n_r}$, where $n_r = |\mathcal{P}_r| = \frac{n}{|\mathcal{G}|} < n^{1-e}$. For every $i \in \{0, \ldots, r-1\}$, $2 \leq p_i \leq p_{max}$. Therefore $r(n) \leq \log_2 n - \log_2 n_r < \log_2 n$.

(Randomness complexity) The randomness complexity is $k(n) = 2r(n) < 2\log_2 n$.

(Query complexity) Notice that for $i \in \{0, \ldots, r-2\}$, $f^{(i+1)}(Q_{i+1})$ is reused for the next round consistency test. Hence, $q(n) = \alpha\left(\sum_{i=0}^{r-1} p_i\right) + n^{1-e} \leq \alpha r p_{max} + n^{1-e}$.

(Proof length) The total proof length $l(n)$ is the sum of the lengths of all the oracles provided by $\textsf{P}$ during the $\textsf{COMMIT}$ phase, counted in field elements. Denoting $t_{i+1} \coloneqq \prod_{j=0}^{i} p_j$, we notice that $|\mathcal{P}_{i+1}| = \frac{|\mathcal{P}_i|}{p_i} = \frac{|\mathcal{P}_0|}{t_{i+1}}$. Thus, we have

$$l(n) = \sum_{i=1}^{r} |\mathcal{P}_i| = \sum_{i=1}^{r} \frac{|\mathcal{P}_0|}{t_i} \leq n \sum_{i=1}^{r} \frac{1}{2^i} = n\left(1 - \frac{1}{2^r}\right) < n.$$

5 IOPP for AG codes on Kummer curves

In this section, we extend the AG-IOPP defined in Section 4.2 for the valid setting of Kummer curves (described in Section 3.2).

5.1 Description of the AG-IOPP for AG codes on Kummer curves

Assume $C_0 = C(\mathcal{C}_0, \mathcal{P}_0, D_0)$ is a foldable AG code of blocklength $n_0 = |\mathcal{P}_0|$ on a Kummer curve $\mathcal{C}_0$ (cf. Proposition 3.9). This means that $\mathcal{C}_0$ is defined by an equation $y^N = f(x)$, where $f \in \mathbb{F}[X]$ is a separable degree-m polynomial, $m \equiv -1 \mod N$, N is coprime with $|\mathbb{F}|$, $|\mathcal{P}_0| = \alpha N$ for some integer α, and $\deg D_0 < \alpha N$. Assume α is a power of 2 and N is a η-smooth integer for a small fixed parameter $\eta \in \mathbb{N}$.

We consider a sequence of codes (C_i) as provided by Section 3.2. Proposition 3.13 states that the relative minimum distances of the codes C_i are all equal to $\Delta(C_0) = 1 - \frac{\deg D_0}{\alpha N}$. Therefore, the ordering on the integers involved in the prime decomposition $\prod_{i=0}^{s-1} p_i$ of N does not impact the parameters of the protocol. Moreover, the code $C_s = C(\mathcal{C}_s, \mathcal{P}_s, D_s)$ corresponds to a RS code

$$C_s = \textsf{RS}\left[\mathbb{F}, \mathcal{P}_s, \frac{\deg D_0}{N}\right] = \left\{f : \mathcal{P}_s \to \mathbb{F}; \deg f \leq \frac{\deg D_0}{N}\right\}$$

of blocklength $|\mathcal{P}_s| = \alpha$, which is itself a foldable AG code (see Example 3.5). Taking this into consideration, we want to iterate the folding operation until we get a RS code of dimension 1, as it is done in the $\textsf{FRI}$ protocol [BBHR18]. As in Example 3.5, we set $d_0 = \frac{\deg D_0}{N}$ and define $d_{i+1} = \left\lfloor \frac{d_i}{2} \right\rfloor$ for any integer i. Set s' the smallest integer such that $d_{s'} = 0$. Then, we consider the sequence of codes $(C_{s+i})_{1\leq i\leq s'}$ when applying the construction described in Section 3.1 to the initial code C_s. Letting $r = s + s'$, we iteratively reduce the proximity test to the code C_0 to a membership test to the code C_r, which is a Reed-Solomon code of dimension 1. If $f^{(0)} \in C_0$, then $f^{(r)}$ is expected to be a constant function, and this can be tested in a trivial way. We can leverage the fact that C_r is a Reed-Solomon code to extend the protocol described in Section 4.2. We obtain a r-rounds IOPP system (P, V) for C_0, which is described in Figures 3 (COMMIT phase) and 4 (QUERY phase).

COMMIT Phase

Common input: C_0 a foldable AG code on a Kummer curve defined by $(\mathbb{F}, \mathcal{C}_0, \mathcal{P}_0, D_0)$, r a number of rounds, (C_i) a sequence of codes, $(\nu_{i+1})_{0\leq i<r}$ some balancing functions.
Prover's input: $f^{(0)} : \mathcal{P}_0 \to \mathbb{F}$.
Output: a sequence of oracle functions $(f^{(0)}, \ldots, f^{(r-1)}) \in \mathbb{F}^{\mathcal{P}_1} \times \ldots \times \mathbb{F}^{\mathcal{P}_{r-1}}$ and $\beta \in \mathbb{F}$.

1. For each round i from 0 to $r - 1$:
   (a) V picks uniformly at random $\boldsymbol{z}^{(i)}$ in $\mathbb{F}^2$ and sends it to P,
   (b) P computes $f^{(i+1)} = \textbf{Fold}\left[f^{(i)}, \boldsymbol{z}^{(i)}\right]$,
   (c) If $i < r - 1$: P gives oracle access to $f^{(i+1)} : \mathcal{P}_{i+1} \to \mathbb{F}$.
   (d) If $i = r - 1$: P commits to $\beta \in \mathbb{F}$ (if $f^{(0)} \in C_0$, then $f^{(r)}$ is supposed to be constant equal to β).

Figure 3: IOPP for AG codes on Kummer curves - COMMIT Phase

QUERY Phase:

Input: (the first four items must correspond to the COMMIT phase)
- C_0 an AG code defined by $(\mathbb{F}, \mathcal{C}_0, \mathcal{P}_0, D_0)$, r a number of rounds,
- sequence of codes (C_i) and balancing functions (ν_{i+1}),
- transcript including $\boldsymbol{z}^{(0)}, \dots, \boldsymbol{z}^{(r-1)} \in \mathbb{F}^2$,
- oracle functions $f^{(0)}, f^{(1)}, \dots, f^{(r-1)}$ and a constant $\beta \in \mathbb{F}$,
- α repetition parameter.

Output: **accept** or **reject**.

1. Repeat α times the following **query test**:
   (a) Pick $Q_0 \in \mathcal{P}_0$ uniformly at random.
   (b) For $i = 0$ to $r-1$, run the following *round consistency test*:
      i. Define $Q_{i+1} \in \mathcal{P}_{i+1}$ by $Q_{i+1} = \pi_i(Q_i)$,
      ii. Query $f^{(i+1)}$ to get $f^{(i+1)}(Q_{i+1})$ and query $f^{(i)}$ at points $\widehat{Q} \in S_{Q_{i+1}}$, (if $i = r-1$, set $f^{(r)}(Q_r) = \beta$)
      iii. Compute the value $\textsf{Fold}\left[f^{(i)}, \boldsymbol{z}^{(i)}\right](Q_{i+1})$,
      iv. If $i < r-1$: return **reject** if and only if $f^{(i+1)}(Q_{i+1}) \neq \textsf{Fold}\left[f^{(i)}, \boldsymbol{z}^{(i)}\right](Q_{i+1})$
      v. If $i = r-1$: return **reject** if and only if $\beta \neq \textsf{Fold}\left[f^{(i)}, \boldsymbol{z}^{(i)}\right](Q_{i+1})$
2. Return **acccept**.

Figure 4: IOPP for AG codes on Kummer curves - QUERY Phase

Example 5.1. On $\mathbb{F}_{q^2}$ with $q = 2^{61} - 1$ (9^{th} Mersenne prime), we consider the curve

$$\mathcal{C}_0 : y^N = x^3 + x$$

where $N = 2^r$ with $r = 16$. It is maximal [TT14] of genus $g = N - 1$. We consider the code C_0 associated to $D_0 = 2^{17} P_\infty^0$ on an evaluation set $\mathcal{P}_0 \subset \mathcal{C}_0(\mathbb{F}_{q^2})$ of size $n = 2^{20}$. Its dimension equals $\dim C_0 = 2^{16} + 2$ and its relative minimum distance λ is bounded from below by $1 - 2^{-3}$. Take $\varepsilon = 2^{-6.5}$. By Proposition 4.7,

$$\mathsf{err}_{commit} \leq \frac{8r}{\left|\mathbb{F}_{q^2}\right| \varepsilon^2} \leq 2^{3+4+13-121} = 2^{-101}$$

$$\mathsf{err}_{query}(\delta) \leq (1 - \delta + r\varepsilon)$$

where $1 - \delta = (1 - \lambda + \varepsilon)^{\frac{1}{3}} \leq 0.51432$. Hence

$$\mathsf{err}_{query}(\delta) \leq 0.51432 + \frac{16}{2^{6.5}} \approx 0.6910.$$

By running the QUERY phase with repetition parameter $\alpha \geq 190$, we get $(\mathsf{err}_{query})^\alpha \leq 2^{-101}$ and $\mathsf{err}(\delta) \leq 2^{-100}$. The last code C_r is a small Reed-Solomon code of length $n_r = 2^4$ and dimension 2. The total number of rounds of the IOPP is thus $R = r + 1$.

5.2 Properties of the AG-IOPP with Kummer curves

Theorem 5.2 (Kummer case)**.** *Let $C = (\mathcal{C}_0, \mathcal{P}_0, D_0)$ be a foldable AG code on a Kummer curve satisfying the hypotheses of Proposition 3.9 with N a η-smooth integer. Denote $n = |\mathcal{P}_0|$. The IOPP (P, V) described in Section 5.1 satisfies Theorem 4.6. Moreover, each oracle $f^{(i)}$ with $1 \le i < r-1$ can be honestly computed using $O(|\mathcal{P}_i|)$ arithmetic operations. Overall, prover arithmetic complexity is $\mathsf{t_p}(n) = O(n)$ and verifier arithmetic complexity is $\mathsf{t_v}(n) = O(\log n)$.*

Proof. (Round complexity) We have that $\prod_{i=0}^{s-1} p_i = \dfrac{n}{n_s}$, where $n_s = |\mathcal{P}_s|$. For every $i \in \{0, \dots, s-1\}$, $2 \le p_i \le \eta$. Therefore $r \le \log_2 n - \log_2 n_s$. Moreover, $2^{s'} \le n_s$, thus the round complexity is $r(n) = r = s + s' \le \log_2 n$.

(Randomness complexity) The randomness complexity is $k(n) \le 2r(n) \le 2 \log_2 n$.

(Query complexity) Notice that for $i \in \{0, \dots, R-2\}$, $f^{(i+1)}(Q_{i+1})$ is reused for the next round consistency test. Hence, $q(n) \le \alpha r \eta + 1 \le \alpha \eta \log_2 n + 1$.

(Proof length) The total proof length $l(n)$ is the sum of the lengths of all the oracles provided by P during the COMMIT phase, counted in field elements. Recall that $|\mathcal{P}_0| = 2^l \prod_{i=0}^{s-1} p_i$ for a certain integer $l > s'$. For $i \in \{r, \dots, r-1\}$, set $p_i = 2$. Denoting $t_{i+1} \coloneqq \prod_{j=0}^{i} p_j$, we notice that $|\mathcal{P}_{i+1}| = \frac{|\mathcal{P}_i|}{p_i} = \frac{|\mathcal{P}_0|}{t_{i+1}}$. Thus, we have

$$l(n) = \sum_{i=1}^{r} |\mathcal{P}_i| = \sum_{i=1}^{r} \frac{|\mathcal{P}_0|}{t_i} \le n \sum_{i=1}^{r} \frac{1}{2^i} = n\left(1 - \frac{1}{2^r}\right) < n.$$

(Prover complexity) By assumption, we have $\max_i p_i \le \eta$ for a given parameter $\eta \in \mathbb{N}$. Fix a round index $i < r-1$, we start by bounding the number of operations of the i^{th} step of the COMMIT phase. To simplify notation, denote $f = f^{(i)}$. For any $\boldsymbol{z} = (z_1, z_2) \in \mathbb{F}^2$, computing the successive powers $(z_1^j, z_2^j)_{0 \le j < p_i}$ takes $2(p_i - 2)$ multiplications. For each $P \in \mathcal{P}_{i+1}$, an honest prover must compute the coefficients $(a_{j,P})_{0 \le j, < P}$ of the polynomial $I_{f,P}(X)$ of degree $\deg I_{f,P} < p_i$ from the interpolation set $\left\{(\mu_i(\widehat{P}), f(\widehat{P})) \mid \widehat{P} \in S_P\right\}$ of size p_i. Since $\mu_i = y$, computing $\mu_i(\widehat{P})$ for $\widehat{P} \in S_P$ is done for free. Moreover, the values of μ_i on S_P form a geometric progression of common ratio ζ_i. Monomial interpolation at p_i points in a geometric progression sequence can be done using $L_i \coloneqq 2\mathsf{M}(p_i) + O(p_i)$ operations (see [BS05, Proposition 5]), where $\mathsf{M}(p_i)$ denotes the cost of multiplying univariate polynomials of degree less than p_i and is known to be $\widetilde{O}(p_i)$, hence $L_i = \widetilde{O}(p_i)$. Thus, evaluating $f_0, \dots, f_{p_i-1}$ on $\mathcal{P}_{i+1}$ can be done in $|\mathcal{P}_{i+1}|\,\widetilde{O}(p_i)$ operations.

Letting $\nu_{i+1,j}$ be as defined in proof of Lemma 3.8, the sequence of functions $(\nu_{i+1,j})_{0<j<p_i}$ can be evaluated at the same point $P \in \mathcal{P}_{i+1}$ in time $O(\log m + p_i)$ using exponentiation by squaring. Remark that the multi-evaluation of $\nu_{i+1,j}$ does not depend on the interaction and can be precomputed. Thus, the evaluations of $\nu_{i+1,1}, \dots \nu_{i+1,p_i-1}$ on $\mathcal{P}_{i+1}$ are obtained with $O((\log m + p_i)\,|\mathcal{P}_{i+1}|)$ operations.

Overall, one can honestly evaluate $\textsf{\textbf{Fold}}\,[f, \boldsymbol{z}] : \mathcal{P}_{i+1} \to \mathbb{F}$ with $O_{\eta,m}(|\mathcal{P}_{i+1}|)$ operations in $\mathbb{F}$. We showed previously that $\sum_{i=1}^{R-1} |\mathcal{P}_i| < n$, thus when summing over $R-1$ rounds, we get that the cost of (honestly) generating the oracles $f^{(1)}, \dots, f^{(R-1)}$ is $O_{\eta,m}(n)$.

Finally, prover complexity is $\mathsf{t_p}(n) = O_{\eta,m}(n)$.

(Verifier decision complexity) Verifier complexity is inferred from the previous discussion about prover complexity. For each round, the verifier computes the successive powers of z_1 and z_2, interpolates $I_{f,P}$ for a point $P \in \mathcal{P}_{i+1}$ in $\widetilde{O}(p_i)$ operations, evaluates $(\nu_{i+1,j})_{0<j<p_i}$ at point P in $O(\log m + p_i)$ operations, then computes $\textsf{Fold}\,[f, \boldsymbol{z}]\,(P)$ in a number of operations which is independent of n. Hence, verifier complexity for repetition parameter α is $\mathsf{t_v}(n) = O_{\eta,m}(\alpha \log(n))$.

(Soundness) Soundness analysis is carried out in Section 6. In particular, in Section 6.2, we set $f^{(r)}$ to be the constant function equal to β. Thus $f^{(r)} \in C_r$, and the verifier accepts if and only if all round consistency tests passes.

□

6 Soundness analysis

We move to the analysis of the soundness error stated in Theorem 4.6. We conduct our analysis using techniques similar to [BGKS20, Section 5.5]. In the first subsection, we establish a result about distance preservation of the folding operation (Corollary 6.3), which will be used in the second subsection to bound the probability of error of the verifier.

6.1 Preliminaries

Roughly speaking, we want to show that, if f is δ-far from C_i, then the folding $\textsf{Fold}\,[f, \boldsymbol{z}]$ of f is almost δ-far from C_{i+1} with high probability over $\boldsymbol{z} \in \mathbb{F}^2$. For soundness analysis, it will be easier to show a weighted version of such statement.

Definition 6.1 (Weighted agreement)**.** *For any function $\eta \in [0,1]^D$, we define the η-agreement of two functions $u, v \in \mathbb{F}^{\mathcal{P}}$ by*

$$\omega_\eta(u, v) \coloneqq \frac{1}{|\mathcal{P}|} \sum_{\substack{P \in \mathcal{P} \\ u(P)=v(P)}} \eta(P).$$

Given $V \subset \mathbb{F}^{\mathcal{P}}$ and $u \in \mathbb{F}^{\mathcal{P}}$, we set

$$\omega_\eta(u, V) \coloneqq \max_{v \in V} \omega_\eta(u, v).$$

Notice that since $\eta \in [0,1]^{\mathcal{P}}$, we have for any $V \subset \mathbb{F}^{\mathcal{P}}$ and any $u \in \mathbb{F}^{\mathcal{P}}$,

$$\omega_\eta(u, V) \leq 1 - \Delta(u, V). \tag{22}$$

We now state a preliminary result concerning the weighted agreement on a low-degree parametrized curve. Proof builds upon the one of [BKS18, Theorem 4.5] and is given in Appendix A.

Proposition 6.2. *Let $\eta \in [0,1]^{\mathcal{P}}$ and $\varepsilon, \delta > 0$ such that and $\delta < J_\varepsilon^l(\lambda)$. Let $u_0, \ldots, u_{l-1} \in \mathbb{F}^{\mathcal{P}}$ such that*

$$\Pr_{z \in \mathbb{F}}\left[\omega_\eta\left(\sum_{i=0}^{l-1} z^i u_i, V\right) > 1 - \delta\right] \geq \frac{l-1}{|\mathbb{F}|}\left(\frac{2}{\varepsilon}\right)^{l+1}, \tag{23}$$

then there exists $T \subset \mathcal{P}$, and $v_0, \ldots, v_{l-1} \in V$ such that:

- $\sum_{P \in T} \eta(P) \geq (1 - \delta - \varepsilon)|\mathcal{P}|$
- *for each i, $u_{i\,|T} = v_{i\,|T}$.*

Here, for a function $u \in \mathbb{F}^{\mathcal{P}}$, $u_{|T} \in \mathbb{F}^{T}$ corresponds to the function obtained by restriction on $T \subset \mathcal{P}$.

As mentioned earlier, soundness analysis relies on the relation between the weighted agreement of f to C_i and the weighted agreement of the folding of f to C_{i+1}, constrained by the next corollary.

Corollary 6.3. *Fix $i \in \{0, \dots, r-1\}$. For a function $\eta : \mathcal{P}_i \to [0,1]$, define $\theta : \mathcal{P}_{i+1} \to [0,1]$ by*

$$\forall P \in \mathcal{P}_{i+1},\ \theta(P) := \frac{1}{p_i} \sum_{\widehat{P} \in S_P} \eta(\widehat{P}).$$

Let λ_i be the minimal relative distance of C_i. Fix $\varepsilon \in (0,1[$ and $\delta < \min\left(J_{\varepsilon}^{p_i}(\lambda_i), \frac{1}{2}\left(\lambda_i + \frac{\varepsilon}{2}\right)\right)$. For any function $f : \mathcal{P}_i \to \mathbb{F}$ such that $\omega_\eta(f, C_i) < 1 - \delta$, we have

$$\Pr_{\boldsymbol{z} \in \mathbb{F}^2}\left[\omega_\theta(\textsf{Fold}\left[f, \boldsymbol{z}\right], C_{i+1}) > 1 - \delta + \varepsilon\right] \le \frac{p_i - 1}{|\mathbb{F}|} \left(\frac{4}{\varepsilon}\right)^{p_i+1}.$$

Proving Corollary 6.3 requires the lemma stated next. We prove Corollary 6.3, then prove Lemma 6.4.

Lemma 6.4. *Let $i \in \{0, \dots, r-1\}$, $D_i \in \mathrm{Div}(\mathcal{C}_i)$ and $\mu_i \in \mathbb{F}(\mathcal{C}_i)$ satisfying Equation* (5)*. Consider a divisor $D_{i+1} \in \mathrm{Div}(\mathcal{C}_{i+1})$ that is (D_i, μ_i)-compatible in the sense of Definition 3.2.*

Fix $j \in \{0, \dots, p_i - 1\}$. Then a function $g \in \mathbb{F}(\mathcal{C}_{i+1})$ belongs to $L(E_{i,j})$ if and only if both functions g and $g\nu_{i+1,j}$ belong to $L(D_{i+1})$.

Proof of Corollary 6.3. Let $f : \mathcal{P}_i \to \mathbb{F}$ be an arbitrary function. According to Equation (19), there exist p_i function $f_j : \mathcal{P}_{i+1} \to \mathbb{F}$ such that for any $\boldsymbol{z} = (z_1, z_2) \in \mathbb{F}^2$,

$$\textsf{Fold}\left[f, \boldsymbol{z}\right] = \sum_{j=0}^{p_i-1} z_1^j f_j + \sum_{j=1}^{p_i-1} z_2^j \nu_{i+1,j} f_j.$$

Rewrite $\textsf{Fold}\left[f, \boldsymbol{z}\right]$ as a polynomial in z_2, i.e. $\textsf{Fold}\left[f, \boldsymbol{z}\right] = f_{z_1} + z_2 f'_1 + \dots + z_2^{p_i-1} f'_{p_i-1}$ where we set $f_{z_1} := \sum_{j=0}^{p_i-1} z_1^j f_j$ and $f'_j := \nu_{i+1,j} f_j$. Finally, set

$$K := \frac{p_i - 1}{2|\mathbb{F}|} \left(\frac{4}{\varepsilon}\right)^{p_i+1}.$$

Let us prove the corollary by contrapositive: assume that

$$\Pr_{\boldsymbol{z} \in \mathbb{F}^2}\left[\omega_\theta(\textsf{Fold}\left[f, \boldsymbol{z}\right], C_{i+1}) > 1 - \delta + \varepsilon\right] > 2K$$

or in other words that $\Pr_{z_1 \in \mathbb{F}}\left[\Pr_{z_2 \in \mathbb{F}}\left[\omega_\theta(\textsf{Fold}\left[f, \boldsymbol{z}\right], C_{i+1}) > 1 - \delta + \varepsilon\right] > K\right] > K$.

Fix $z_1 \in \mathbb{F}$ such that $\Pr_{z_2 \in \mathbb{F}}\left[\omega_\theta(\textsf{Fold}\left[f, \boldsymbol{z}\right], C_{i+1}) > 1 - \delta + \varepsilon\right] > K$. By Proposition 6.2, there exist $v_{z_1}, v'_1, \dots, v'_{p_i-1} \in C_{i+1}$ and $\mathcal{T}' \subset \mathcal{P}$ such that

- $\sum_{P \in \mathcal{T}'} \theta(P) \ge \left(1 - \delta + \frac{\varepsilon}{2}\right) |\mathcal{P}_{i+1}|$,
- $v_{z_1 |\mathcal{T}'} = f_{z_1}|_{\mathcal{T}'}$,
- for each $j \in \{1, \dots, p_i - 1\}$, $v'_j|_{\mathcal{T}'} = f'_j|_{\mathcal{T}'}$.

In particular, $\omega_\theta(f_{z_1}, C_{i+1}) \geq \omega_\theta(f_{z_1}, v_{z_1}) = \dfrac{1}{|\mathcal{P}_{i+1}|} \sum_{P\in\mathcal{T}'} \theta(P) \geq 1-\delta+\dfrac{\varepsilon}{2}$.

It means that

$$\Pr_{z_1\in\mathbb{F}}\left[\omega_\theta(f_{z_1}, C_{i+1}) \geq 1-\delta+\frac{\varepsilon}{2}\right] \geq \Pr_{z_1\in\mathbb{F}}\left[\Pr_{z_2\in\mathbb{F}}[\omega_\theta(\textsf{\textbf{Fold}}\,[f,\boldsymbol{z}], C_{i+1}) > 1-\delta+\varepsilon] > K\right] > K.$$

The polynomial form of f_{z_1} in z_1 enables us to reapply Proposition 6.2: there exist $v_0, v_1, \ldots, v_{p_i-1} \in C_{i+1}$ and $\mathcal{T} \subset \mathcal{P}$ such that

- $\sum_{P\in\mathcal{T}} \theta(P) \geq (1-\delta)\,|\mathcal{P}_{i+1}|$,
- for each $j \in \{0, \ldots, p_i - 1\}$, $v_j\,{}_{|\mathcal{T}} = f_j\,{}_{|\mathcal{T}}$.

On $\mathcal{T}' \cap \mathcal{T}$, we thus have $v_j'|_{\mathcal{T}'\cap\mathcal{T}} = f_j'|_{\mathcal{T}'\cap\mathcal{T}} = (\nu_{i+1,j} f_j)|_{\mathcal{T}'\cap\mathcal{T}} = (\nu_{i+1,j} v_j)|_{\mathcal{T}'\cap\mathcal{T}}$. The cardinality of $\mathcal{T}' \cap \mathcal{T}$ satisfies

$$|\mathcal{T}' \cap \mathcal{T}| = |\mathcal{T}'| + |\mathcal{T}| - |\mathcal{T}' \cup \mathcal{T}| \geq \sum_{P\in\mathcal{T}'} \theta(P) + \sum_{P\in\mathcal{T}} \theta(P) - |\mathcal{P}_{i+1}| \geq (1-2\delta+\frac{\varepsilon}{2})|\mathcal{P}_{i+1}|.$$

The assumption on δ ensures that $2\delta - \frac{\varepsilon}{2} < \lambda_{i+1}$ where λ_{i+1} is the minimal distance of C_{i+1}, hence the codewords of C_{i+1} associated to v_j' and $\nu_{i+1,j} v_j$ are equals for every $j \in \{0, \ldots, p_i - 1\}$. This implies that both functions v_j and $\nu_{i+1,j} v_j$ belong to $L(D_{i+1})$. By Lemma 6.4, we get that the function v_j lies in $L(E_{i,j})$.

Now let us define $v : \mathcal{P}_i \to \mathbb{F}$ by

$$\forall Q \in \mathcal{P}_i,\ v(Q) := \sum_{j=0}^{p_i-1} \mu_i^j(Q) v_j \circ \pi_i(Q).$$

By definition of the divisors $E_{i,j}$ (6), the function v belong to $L(D_i)$. Now let us prove that it agrees with f on $S_{\mathcal{T}} := \bigsqcup_{P\in\mathcal{T}} S_P$.

Let $P \in \mathcal{T}$ and $\widehat{P} \in S_P$.

$$\begin{aligned}
f(\widehat{P}) = I_{f,P}(\mu_i(\widehat{P})) &= \sum_{j=0}^{p_i-1} \mu_i(\widehat{P})^j f_j(P) && \text{by definition of } I_{f,P},\\
&= \sum_{j=0}^{p_i-1} \mu_i(\widehat{P})^j v_j \circ \pi_i(\widehat{P}) && \text{since } f_j\,{}_{|\mathcal{T}} = v_j\,{}_{|\mathcal{T}} \text{ and } P = \pi_i(\widehat{P}),\\
&= v(\widehat{P}).
\end{aligned}$$

As a result, since $v \in C_i$, we can conclude that

$$\omega_\eta(f, C_i) \geq \omega_\eta(f, v) \geq \frac{1}{|\mathcal{P}_i|} \sum_{P\in\mathcal{T}} \sum_{\widehat{P}\in S_P} \eta(\widehat{P}) = \frac{1}{|\mathcal{P}_{i+1}|} \sum_{P\in\mathcal{T}} \theta(P) \geq 1-\delta.$$

□

Proof of Lemma 6.4. Assume that $g \in L(E_{i,j})$. Then the second and third items of Definition 3.2 ensure that g and $g\nu_{i+1,j}$ lie in $L(D_{i+1})$.

Conversely, assume that g and $g\nu_{i+1,j}$ belong to $L(D_{i+1})$ and write $D_{i+1} = \sum n_P P$. The hypotheses on g imply that $g \in L(D_{i+1}) \cap L(D_{i+1} - (\nu_{i+1,j}))$. By [MP93, Lemma 2.6], the function g belongs to $L(D'_{i+1})$, where the divisor D'_{i+1} is defined by

$$D'_{i+1} \coloneqq \sum_P n'_P P \text{ where } n'_P \coloneqq \min(n_P, n_P + v_P(\nu_{i+1,j})).$$

Then $D'_{i+1} = D_{i+1} - (\nu_{i+1,j})_\infty = E_{i,j}$ by the third item of Definition 3.2. □

6.2 Proof of Theorem 4.6 (part 2: soundness)

Let $(f^{(i)})_{1 \le i \le r}$ be the output of the COMMIT phase. For simplicity, assume the repetition parameter is set to $\alpha = 1$. The soundness error for $\alpha > 1$ directly follows from this case.

Let $Q_0 \in \mathcal{P}_0$ be the point selected at random by the verifier at the beginning of the QUERY phase. We recall that Q_0 defines a sequence $(Q_i)_{1 \le i \le r}$ satisfying $Q_{i+1} = \pi_i(Q_i)$. In particular, $Q_i \in S_{Q_{i+1}}$, where $S_{Q_{i+1}} = \pi_i^{-1}\{(Q_{i+1})\}$. The verifier accepts if both

1. for all $i \in \{0, \dots, r-1\}$, $f^{(i+1)}(Q_{i+1}) = \textsf{Fold}\left[f^{(i)}, \boldsymbol{z}^{(i)}\right](Q_{i+1})$,
2. $f^{(r)} \in C_r$.

Notice that if $f^{(r)} \notin C_r$, the verifier rejects with probability 1. So from now on, we assume $f^{(r)} \in C_r$.

Coloring the graph induced by prover's oracles. Consider the $(r+1)$-layered graph G with vertex set $\mathcal{P}_0 \sqcup \mathcal{P}_1 \sqcup \cdots \sqcup \mathcal{P}_r$ and edges from $P_{i+1} \in \mathcal{P}_{i+1}$ to $P_i \in \mathcal{P}_i$ if and only if $\pi_i(P_i) = P_{i+1}$. For any edge of G, we say that P_{i+1} is a *parent* of P_i. Any pair of points sharing the same parent are said to be *siblings*. For any point $P_r \in \mathcal{P}_r$, denote $G_{|P_r}$ the subgraph of G corresponding to the complete tree with root P_r. Notice that the trees $G_{|P_r}$ are disjoint.

A query test starts by selecting a leaf $Q_0 \in \mathcal{P}_0$. This leaf belongs to a tree $G_{|P_r}$ for a certain $P_r \in \mathcal{P}_r$, and the verifier queries one set of siblings at each layer $i \in \{0, \dots, r-1\}$ of $G_{|P_r}$. We referred to such a subset of vertices of G as the *path from* Q_0 *to* P_r (a path to P_r does not include P_r).

We now color each vertex of G according to its success in passing the round consistency test. For $i \in \{0, \dots, r-1\}$, a vertex $P_i \in \mathcal{P}_i$ is colored green if

$$f^{(i+1)}(\pi_i(P_i)) = \textsf{Fold}\left[f^{(i)}, \boldsymbol{z}^{(i)}\right](\pi_i(P_i))$$

and colored red otherwise. Notice that P_i gets the same color than its siblings. The verifier outputs accept if and only if every vertex along the queried path from Q_0 to P_r is green.

Tracking agreement between $f^{(i)}$ and the folding of $f^{(i-1)}$. Define $\eta^{(0)} : \mathcal{P}_0 \to \{0,1\}$ by setting $\eta^{(0)}(P) = 1$ if and only if $P \in \mathcal{P}_0$ is green. For all $i \in \{1, \dots, r\}$, define a function

$$\eta^{(i)} : \mathcal{P}_i \to (0,1)$$

such that $\eta^{(i)}(P)$ is equal to the fraction of leaves $P_0 \in \mathcal{P}_0$ for which the path from P_0 to P contains only green vertices. By construction the probability err_{query} that the verifier accepts during the QUERY phase is given by

$$\mathsf{err}_{query} = \frac{1}{n_r} \sum_{P \in \mathcal{P}_r} \eta^{(r)}(P),$$

where n_i denotes the size of $\mathcal{P}_i$. For $i \in \{0, \ldots, r\}$, let us set

$$\omega_{f^{(i)}} \coloneqq \omega_{\eta^{(i)}}(f^{(i)}, C_i),$$

where the η-agreement function ω_η is defined in Definition 6.1. Since $f^{(r)} \in C_r$, observe that

$$\mathsf{err}_{query} = \omega_{f^{(r)}}. \tag{24}$$

For $i \in \{0, \ldots, r-1\}$, let $E^{(i+1)} \subseteq \mathcal{P}_{i+1}$ be the set of coordinates where $f^{(i+1)}$ differs from $\mathsf{Fold}\left[f^{(i)}, \boldsymbol{z}^{(i)}\right]$, i.e. $E^{(i+1)} \coloneqq \left\{P \in \mathcal{P}_{i+1} \mid \forall \widehat{P} \in S_P, \widehat{P} \text{ is } \mathsf{red}\right\}$. Define $\theta^{(i+1)} : \mathcal{P}_{i+1} \to (0;1)$ such that

$$\theta^{(i+1)}(P) = \frac{1}{p_i} \sum_{\widehat{P} \in S_P} \eta^{(i)}(\widehat{P}).$$

Denoting $\mathbb{1}_{E^{(i+1)}}$ the indicator function of $E^{(i+1)} \subseteq \mathcal{P}_{i+1}$, we observe that

$$\eta^{(i+1)} = (1 - \mathbb{1}_{E^{(i+1)}})\, \theta^{(i+1)}. \tag{25}$$

Define $\beta^{(i+1)} \coloneqq \omega_{\theta^{(i+1)}}\left(\mathsf{Fold}\left[f^{(i)}, \boldsymbol{z}^{(i)}\right], C_{i+1}\right)$ which, by Equation (25), satisfies

$$\beta^{(i+1)} \geq \omega_{f^{(i+1)}}. \tag{26}$$

Let $\varepsilon' \in (0, \varepsilon)$. Set $\delta^{(i)} = \min(1 - \omega_{f^{(i)}}, J_\varepsilon^{p_i}(\lambda_i)) - \varepsilon'$. Then $\delta^{(i)}$ fulfills all the hypotheses of Corollary 6.3 and

$$\Pr_{\boldsymbol{z}^{(i)}}\left[\beta^{(i+1)} > \max\left(\omega_{f^{(i)}}, 1 - J_\varepsilon^{p_i}(\lambda_i)\right) + \varepsilon\right] \leq \frac{p_i - 1}{|\mathbb{F}|}\left(\frac{4}{\varepsilon - \varepsilon'}\right)^{p_i+1}.$$

Thus, for all $i \in \{0, \ldots, r-1\}$, we get that

$$\Pr_{\boldsymbol{z}^{(i)}}\left[\beta^{(i+1)} > \max\left(\omega_{f^{(i)}}, 1 - J_\varepsilon^{p_i}(\lambda_i)\right) + \varepsilon\right] \leq \frac{p_i - 1}{|\mathbb{F}|}\left(\frac{4}{\varepsilon}\right)^{p_i+1}.$$

by making ε' going to 0, by continuity of the right hand-side at $\varepsilon \neq 0$.

Let $\lambda \coloneqq \min(\Delta(C_i))$ and $p_{\max} = \max(p_i)$. Since the function $J_\varepsilon^{p_i}$ is strictly increasing and the sequence of functions $(J_\varepsilon^l)_l$ is decreasing, we have for all $i \in \{0, \ldots, r-1\}$,

$$\Pr_{\boldsymbol{z}^{(i)}}[\beta^{(i+1)} > \max\left(\omega_{f^{(i)}}, 1 - J_\varepsilon^{p_{max}}(\lambda)\right) + \varepsilon] \leq \frac{p_{\max} - 1}{|\mathbb{F}|}\left(\frac{4}{\varepsilon}\right)^{p_{\max}+1}.$$

From Equation (26), we deduce that for all $i \in \{0, \ldots, r-1\}$,

$$\Pr_{\boldsymbol{z}^{(i)}}\left[\omega_{f^{(i+1)}} > \max\left(\omega_{f^{(i)}}, 1 - J_\varepsilon^{p_{max}}(\lambda)\right) + \varepsilon\right] \leq \frac{p_{\max} - 1}{|\mathbb{F}|}\left(\frac{4}{\varepsilon}\right)^{p_{\max}+1}. \tag{27}$$

Set $\mathsf{err}_{commit} \coloneqq r\frac{p_{\max}-1}{|\mathbb{F}|}\left(\frac{4}{\varepsilon}\right)^{p_{\max}+1}$. Thus, from Equation (27) and by union bound, we get that

$$\Pr_{\boldsymbol{z}^{(0)}, \ldots, \boldsymbol{z}^{(r-1)}}\left[\omega_{f^{(r)}} \leq \max\left(\omega_{f^{(0)}}, 1 - J_\varepsilon^{p_{max}}(\lambda)\right) + r\varepsilon\right] \geq 1 - \mathsf{err}_{commit}. \tag{28}$$

Recall that $\omega_{f^{(0)}} \leq 1 - \Delta(f^{(0)}, C_0) < 1 - \delta$ and $\mathsf{err}_{query} = \omega_{f^{(r)}}$ (from Equation (24)). We deduce that with probability at least $1 - \mathsf{err}_{commit}$ over the verifier random choices during the COMMIT phase, the probability that the verifier accepts during the QUERY phase is at most

$$\begin{aligned}\mathsf{err}_{query} = \omega_{f^{(r)}} &\leq \max(\omega_{f^{(0)}}, 1 - J_\varepsilon^{p_{max}}(\lambda)) + r\varepsilon \\ &< 1 - \min\left(\delta, J_\varepsilon^{p_{max}}(\lambda)\right) + r\varepsilon.\end{aligned}$$

This concludes the proof of soundness of Theorem 4.6.

Acknowledgments

The first author benefits from the support of the Chair "Blockchain & B2B Platforms", led by l'*X – École Polytechnique* and the *Fondation de l'École Polytechnique*, sponsored by *Capgemini*. The second author thanks Marc Perret for his precious advices in the early days of this project.

References

[AHIV17] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight Sublinear Arguments Without a Trusted Setup. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 2087–2104. ACM, 2017.

[ALM+98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof Verification and the Hardness of Approximation Problems. 45(3):501–555, 1998. extended version of FOCS'92.

[AS92] Sanjeev Arora and Shmuel Safra. 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*, pages 2–13. IEEE Computer Society, 1992.

[Bab85] László Babai. Trading Group Theory for Randomness. In Robert Sedgewick, editor, *Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA*, pages 421–429. ACM, 1985.

[BB09] Alp Bassa and Peter Beelen. On the Construction of Galois Towers. *Contemporary mathematics*, 487:9–20, 2009.

[BB11] Alp Bassa and Peter Beelen. The Galois closure of Drinfeld modular towers. *Journal of Number Theory*, 131(3):561–577, 2011.

[BBGS14] Alp Bassa, Peter Beelen, Arnaldo Garcia, and Henning Stichtenoth. An Improvement of the Gilbert–Varshamov Bound Over Nonprime Fields. *IEEE Transactions on Information Theory*, 60(7):3859–3861, 2014.

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. 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*, pages 14:1–14:17, 2018.

[BBHR19] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable Zero Knowledge with No Trusted Setup. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part III*, volume 11694 of *Lecture Notes in Computer Science*, pages 701–732. Springer, 2019.

[BCG+17] Eli Ben-Sasson, Alessandro Chiesa, Ariel Gabizon, Michael Riabzev, and Nicholas Spooner. 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*, pages 40:1–40:15, 2017.

[BCG+19] Eli Ben-Sasson, Alessandro Chiesa, Lior Goldberg, Tom Gur, Michael Riabzev, and Nicholas Spooner. Linear-Size Constant-Query IOPs for Delegating Computation. In Dennis Hofheinz and Alon Rosen, editors, *Theory of Cryptography - 17th International Conference, TCC 2019, Nuremberg, Germany, December 1-5, 2019, Proceedings, Part II*, volume 11892 of *Lecture Notes in Computer Science*, pages 494–521. Springer, 2019.

[BCI+20] Eli Ben-Sasson, Dan Carmon, Yuval Ishai, Swastik Kopparty, and Shubhangi Saraf. Proximity Gaps for Reed-Solomon Codes. *IACR Cryptol. ePrint Arch.*, 2020:654, 2020.

[BCR+19] Eli Ben-Sasson, Alessandro Chiesa, Michael Riabzev, Nicholas Spooner, Madars Virza, and Nicholas P. Ward. Aurora: Transparent Succinct Arguments for R1CS. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part I*, volume 11476 of *Lecture Notes in Computer Science*, pages 103–128. Springer, 2019.

[BCS16] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive Oracle Proofs. In *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part II*, pages 31–60, 2016.

[BFLS91] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. 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*, pages 21–31, 1991.

[BGKS20] Eli Ben-Sasson, Lior Goldberg, Swastik Kopparty, and Shubhangi Saraf. DEEP-FRI: Sampling Outside the Box Improves Soundness. In *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*, pages 5:1–5:32, 2020.

[BKK+13] Eli Ben-Sasson, Yohay Kaplan, Swastik Kopparty, Or Meir, and Henning Stichtenoth. 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*, pages 320–329. IEEE Computer Society, 2013.

[BKS18] Eli Ben-Sasson, Swastik Kopparty, and Shubhangi Saraf. 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*, pages 24:1–24:23, 2018.

[BMZ18a] Daniele Bartoli, Maria Montanucci, and Giovanni Zini. AG codes and AG quantum codes from the GGS curve. *Designs, Codes and Cryptography*, 86(10):2315–2344, October 2018.

[BMZ18b] Daniele Bartoli, Maria Montanucci, and Giovanni Zini. Multi point AG codes on the GK maximal curve. *Designs, Codes and Cryptography*, 86(1):161–177, January 2018.

[BRS20] Peter Beelen, Johan Rosenkilde, and Grigory Solomatov. Fast Encoding of AG Codes over C_{ab} Curves, 2020.

[BS05] Alin Bostan and Eric Schost. Polynomial evaluation and interpolation on special sets of points. *Journal of Complexity*, 21(4):420–446, 2005.

[BS08] Eli Ben-Sasson and Madhu Sudan. Short PCPs with Polylog Query Complexity. *SIAM J. Comput.*, 38(2):551–607, 2008.

[COS20] Alessandro Chiesa, Dev Ojha, and Nicholas Spooner. Fractal: Post-quantum and Transparent Recursive Proofs from Holography. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I*, volume 12105 of *Lecture Notes in Computer Science*, pages 769–793. Springer, 2020.

[CRL90] M. Carral, D. Rotillon, and A. Thiong Ly. Codes defined from some maximal curves. *Journal of Pure and Applied Algebra*, 67(3):247–257, 1990.

[Din07] Irit Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3):12, 2007.

[FG10] Stefania Fanali and Massimo Giulietti. One-Point AG Codes on the GK Maximal Curves. *IEEE Transactions on Information Theory*, 56(1):202–210, January 2010.

[GMP12] Robert Guralnick, Beth Malmskog, and Rachel Pries. The Automorphism Groups of a Family of Maximal Curves. *Journal of Algebra*, 361:92–106, 2012.

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The Knowledge Complexity of Interactive Proof-Systems (Extended Abstract). In Robert Sedgewick, editor, *Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA*, pages 291–304. ACM, 1985.

[Gop77] Valerii Denisovich Goppa. Codes associated with divisors. *Problemy Peredachi Informatsii*, 13(1):33–39, 1977.

[HKT13] J. W. P. Hirschfeld, G. Korchmáros, and F. Torres. *Algebraic Curves over a Finite Field*. Princeton University Press, Princeton, 25 Mar. 2013.

[Kan86] Ernst Kani. The Galois-module structure of the space of holomorphic differentials of a curve. *Journal für die reine und angewandte Mathematik*, 367:187–206, 1986.

[Kil92] Joe Kilian. A Note on Efficient Zero-Knowledge Proofs and Arguments (Extended Abstract). In S. Rao Kosaraju, Mike Fellows, Avi Wigderson, and John A. Ellis, editors, *Proceedings of the 24th Annual ACM Symposium on Theory of Computing, May 4-6, 1992, Victoria, British Columbia, Canada*, pages 723–732. ACM, 1992.

[KPV19] Assimakis Kattis, Konstantin Panarin, and Alexander Vlasov. RedShift: Transparent SNARKs from List Polynomial Commitment IOPs. *IACR Cryptol. ePrint Arch.*, 2019:1400, 2019.

[KR08] Yael Tauman Kalai and Ran Raz. Interactive PCP. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations*, volume 5126 of *Lecture Notes in Computer Science*, pages 536–547. Springer, 2008.

[Lac87] Gilles Lachaud. Sommes d'Eisenstein et nombre de points de certaines courbes algébriques sur les corps finis. *C. R. Acad. Sci. Paris*, 305, 01 1987.

[LFKN90] Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. 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*, pages 2–10. IEEE Computer Society, 1990.

[LSH97] John Little, Keith Saints, and Chris Heegard. On the structure of Hermitian codes. *Journal of Pure and Applied Algebra*, 121(3):293–314, 1997.

[Mah04] Hiren Maharaj. Code Construction on Fiber Products of Kummer Covers. *Information Theory, IEEE Transactions on*, 50:2169 – 2173, 10 2004.

[Mat05] Gretchen L. Matthews. Weierstrass Semigroups and Codes from a Quotient of the Hermitian Curve. *Designs, Codes and Cryptography*, 37(3):473–492, 2005.

[Mei13] Or Meir. IP = PSPACE Using Error-Correcting Codes. *SIAM J. Comput.*, 42(1):380–403, 2013.

[Mic95] Silvio Micali. Computationally-Sound Proofs. In Johann A. Makowsky and Elena V. Ravve, editors, *Proceedings of the Annual European Summer Meeting of the Association of Symbolic Logic, Logic Colloquium 1995, Haifa, Israel, August 9-18, 1995*, volume 11 of *Lecture Notes in Logic*, pages 214–268. Springer, 1995.

[Mie09] Thilo Mie. Short PCPPs Verifiable in Polylogarithmic Time with O(1) Queries. *Annals of Mathematics and Artificial Intelligence*, 56(3–4):313–338, August 2009.

[Mon20] Maria Montanucci. On algebraic curves with many automorphisms in characteristic p. *arXiv preprint arXiv:2001.07514*, 2020.

[MP93] Carlos Munuera and Ruud Pellikaan. Equality of geometric Goppa codes and equivalence of divisors. *Journal of Pure and Applied Algebra*, 90(3):229 – 252, 1993.

[MQS15] Ariane M. Masuda, Luciane Quoos, and Alonso Sepúlveda. One- and Two-Point Codes over Kummer Extensions. *arXiv e-prints*, page arXiv:1510.06425, October 2015.

[Ren04] Jian Ren. On the structure of Hermitian codes and decoding for burst errors. *IEEE Transactions on Information Theory*, 50(11):2850–2854, 2004.

[RR20] Noga Ron-Zewi and Ron D. Rothblum. 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*, pages 846–857. IEEE, 2020.

[RRR16] Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. Constant-round interactive proofs for delegating computation. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 49–62. ACM, 2016.

[SG93] B.Z. Shen and M.J. Ganley. On encoding and decoding of the codes from Hermitian curves. 1993.

[Sti93] Henning Stichtenoth. *Algebraic function fields and codes.* Universitext. Springer, 1993.

[TT14] Saeed Tafazolian and Fernando Torres. On the curve $y^n = x^m + x$ over finite fields. *Journal of Number Theory*, 145:51–66, 2014.

[TVN07] Michael Tsfasman, Serge Vladut, and Dmitry Nogin. *Algebraic Geometric Codes: Basic Notions.* American Mathematical Society, USA, 2007.

[TVZ82] M. A. Tsfasman, S. G. Vlăduţ, and Th. Zink. Modular curves, Shimura curves, and Goppa codes, better than Varshamov-Gilbert bound. *Math. Nachr.*, 109:21–28, 1982.

[YB92] Tomik Yaghoobian and Ian F. Blake. Hermitian codes as generalized Reed-Solomon codes. *Designs, Codes and Cryptography*, 2(1):5–17, 1992.

A Proof of Proposition 6.2

Proposition 6.2 is a weighted version of [BKS18, Theorem 4.5]. We only highlight the changes to be made in the proof of [BKS18, Theorem 4.5].

For $z \in \mathbb{F}$ and $(v_0, \ldots, v_{l-1}) \in V^l$, let us set $v_z \coloneqq \sum_{i=0}^{l-1} z^i v_i$. Rewriting the proof of Theorem 4.5 [BKS18] with setting

$$A = \{z \in \mathbb{F} \mid \omega_\eta (u_z, V) > 1 - \delta\}$$

provides $v_0, \ldots, v_{l-1} \in V$ and a set

$$C \coloneqq \{z \in \mathbb{F} \mid \omega_\eta (u_z, v_z) > 1 - \delta\} \subset A$$

with cardinality $|C| > \frac{l-1}{\varepsilon}$. Let us set $T \coloneqq \{P \in \mathcal{P} \mid u_{i\,|T} = v_{i\,|T} \text{ for all } i\}$. Therefore

$$\begin{aligned}
1 - \delta &< \frac{1}{|C|} \sum_{z \in C} \omega_\eta (u_z, v_z) \\
&= \frac{1}{|C| \times |\mathcal{P}|} \sum_{z \in C} \sum_{P \in \mathcal{P}} \eta(P) \mathbb{1}_{u_z(P) = v_z(P)} \\
&= \frac{1}{|\mathcal{P}|} \sum_{P \in \mathcal{P}} \eta(P) \frac{1}{|C|} \sum_{z \in C} \mathbb{1}_{u_z(P) = v_z(P)}
\end{aligned}$$

Notice that if there exists $i \in \{0, \ldots, l-1\}$ such that u_i which does not coincide with v_i, the number of $z \in \mathbb{F}$ such that $u_z(P) = v_z(P)$ is at most $l - 1$. Then

$$\begin{aligned}
1 - \delta &\le \frac{1}{|\mathcal{P}|} \sum_{P \in T} \eta(P) + \frac{1}{|\mathcal{P}|} \sum_{P \in C \setminus T} \eta(P) \frac{l-1}{|C|} \\
&\le \frac{1}{|\mathcal{P}|} \sum_{P \in T} \eta(P) + \varepsilon,
\end{aligned}$$

which gives the first item of the proposition.

B Improved soundness for $p_{\max} = 2$

In the case where $p_i = 2$ for every i, a stronger bound on soundness can be obtained, as stated in Proposition 4.7. We give a sketch of proof, starting from the result of [BGKS20, Lemma 3.2] and applying exactly the same analysis as in Section 6. The expression of $\textsf{Fold}[f, \boldsymbol{z}] : \mathcal{P}_{i+1} \to \mathbb{F}$ in when $p_{max} = 2$ is

$$\textsf{Fold}[f, (z_1, z_2)] = f_0 + z_1 f_1 + z_2 \nu_{i+1,1} f_1. \tag{29}$$

By applying to [BGKS20, Lemma 3.2] the same reasoning than the one applied for proof of Proposition 6.2, we get the following proposition.

Proposition B.1. *Let $\eta \in [0,1]^{\mathcal{P}}$ and $\varepsilon, \delta > 0$ with $\varepsilon < 1/3$ and $\delta < 1 - (1 - \lambda + \varepsilon)^{1/3}$, where $\lambda = \Delta(V)$. Let $u_0, u_1 \in \mathbb{F}^{\mathcal{P}}$ such that*

$$\Pr_{z \in \mathbb{F}} [\omega_\eta (u_0 + z u_1, V) > 1 - \delta] \geq \frac{2}{\varepsilon^2 |\mathbb{F}|}, \tag{30}$$

then there exists $T \subset \mathcal{P}$, and $v_0, v_1 \in V$ such that:

- $\sum_{P \in T} \eta(P) \geq (1 - \delta - \varepsilon)|\mathcal{P}|$
- *for each i, $u_{i \mid T} = v_{i \mid T}$.*

This yields an analogous of Corollary 6.3, stated next.

Corollary B.2. *Fix $i \in \{0, \ldots, r-1\}$. For a function $\eta : \mathcal{P}_i \to [0,1]$, define $\theta : \mathcal{P}_{i+1} \to [0,1]$ by*

$$\forall P \in \mathcal{P}_{i+1},\ \theta(P) \coloneqq \frac{1}{p_i} \sum_{\widehat{P} \in S_P} \eta(\widehat{P}).$$

Let λ_i be the minimal relative distance of C_i. Fix $\varepsilon \in \left(0, \frac{1}{3}\right)$ and $\delta < \min\left(1 - (1 - \lambda_i + \varepsilon)^{1/3}, \frac{1}{2}\left(\lambda_i + \frac{\varepsilon}{2}\right)\right)$. For any function $f : \mathcal{P}_i \to \mathbb{F}$ such that $\omega_\eta(f, C_i) < 1 - \delta$, we have

$$\Pr_{\boldsymbol{z} \in \mathbb{F}^2} [\omega_\theta(\textsf{Fold}[f, \boldsymbol{z}], C_{i+1}) > 1 - \delta + \varepsilon] \leq \frac{8}{\varepsilon^2 |\mathbb{F}|}.$$

After making the substitutions related to the above statements in Section 6.2, we set

$$\textsf{err}_{commit} = r \frac{8}{\varepsilon^2 |\mathbb{F}|},$$

and with probability at least $1 - \textsf{err}_{commit}$, the verifier accepts on input f such that $\Delta(f, C_0) > \delta$ with probability at most

$$\textsf{err}_{query} < 1 - \min\left(\delta, 1 - (1 - \lambda_{min} + \varepsilon)^{1/3}\right) + r\varepsilon.$$

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il